PROGRAMA DE DOCTORADO EN CIENCIA Y TECNOLOGÍA

TESIS DOCTORAL

OPTIMIZACIÓN ALGORÍTMICA Y COMPUTACIONAL DE LA ESTIMACIÓN DEL ESPECTRO ANGULAR DE POTENCIAS DE TEMPERATURA Y POLARIZACIÓN DE OBSERVACIONES DEL FONDO CÓSMICO DE MICROONDAS

PhD THESIS

ALGORITHMIC AND COMPUTATIONAL OPTIMIZATION OF THE ESTIMATION OF TEMPERATURE AND POLARIZATION ANGULAR POWER SPECTRUM FOR COSMIC MICROWAVE BACKGROUND OBSERVATIONS

AUTOR
JUAN DANIEL BILBAO AHEDO
DIRECTORES
RITA BELÉN BARREIRO VILAS
DIEGO HERRANZ MUÑOZ

UNIVERSIDAD DE CANTABRIA

Escuela de Doctorado de la Universidad de Cantabria

Santander 2022

Agradecimientos

Quisiera expresar mi agradecimiento a Belén Barreiro, mi directora de Tesis. Sin su ayuda y orientación no habría podido realizarla. También a Diego Herranz, codirector, cuya colaboración ha facilitado mi empeño en esta tarea. Ha sido un camino largo para llegar hasta el final, con muchos encuentros y algún desencuentro. Por encima de todo, valoro el trabajo y todas las horas que esta Tesis les ha robado. Muchas gracias a los dos.

Tampoco podría haber recorrido este camino sin la ayuda de Patricio Vielva y Enrique Martínez. Los ratos que he pasado con todos o algunos de vosotros cuatro aprendiendo, complementando y contrastando ideas han sido un placer. Con vosotros, y con todos, he descubierto nuevas formas de pensar y proceder.

Quiero, expresar mi agradecimiento al grupo de cosmología del IFCA, en su conjunto. A quienes lo forman ahora y a quienes me encontré cuando llegué a él, y ya no están. Siempre me sentí muy bien acogido y cómodo. Siempre dispuse de ayuda cuando la necesité. Agradezco los estimulantes retos que me propusieron cuando caí por ahí; especialmente, a José María Diego. En buena parte, son la causa de que ahora esté escribiendo este texto.

Quiero agradecer a mi familia y amigos el apoyo que me dieron y la ilusión que transmitían. Una de las cosas más bonitas de haber hecho este trabajo es observar la alegría que les producía cada uno de los pequeños avances que iba realizando. Me apena no haber podido ver esa alegría en el rostro de mi querido padre.

Quiero expresar mi reconocimiento y agradecimiento a la comunidad científica. A los del presente y a los del pasado, a los grandes nombres y a los no tan conocidos. Desde muy pronto, me cautivó vuestra forma de pensar, vuestras metas y cómo lo hacéis para avanzar en el conocimiento y la comprensión del mundo en que vivimos; buscando respuestas en la razón y la experimentación, lejos de las arbitrariedades y supersticiones que, lamentablemente, siguen siendo tan comunes e invasoras. En cuanto a este trabajo, gracias a que lo muy difícil y lo simplemente difícil ya lo habíais resuelto, me ha tocado lo sencillo, y he podido contribuir con mi pequeño paso adelante. Espero humildemente que sea de utilidad. Respecto a mí, el hecho de efectuarlo me ha producido gran satisfacción.

Abstract

El Fondo Cósmico de Microondas (FCM) codifica información clave que conecta de forma directa las observaciones que realizamos aquí y en el presente con el estado del universo temprano. Además, de acuerdo con nuestros modelos teóricos, el conocimiento de las propiedades estadísticas del FCM conduce al conocimiento sobre las características de nuestro Universo en el límite de los instantes iniciales, y a predecir su futuro. La información relevante que nos transmite el FCM está codificada en su espectro angular de potencias. Este trabajo está dedicado al estudio del Estimador Cuadrático de Máxima Verosimilitud (QML), un método óptimo de estimación del espectro de potencia. Analizamos sus propiedades, las condiciones de tipo matemático que se han de cumplir para utilizarlo, soluciones en el caso en el que alguna no se cumpla y estudiamos el rendimiento del método en múltiples situaciones de interés práctico en el presente y el futuro inmediato. El método QML conlleva una alta carga computacional. Tras un análisis detallado de los entresijos matemáticos propios del método, hemos desarrollado una implementación óptima que formalmente permite aplicarlo con los medios técnicos actuales en situaciones que hasta ahora eran inviables. Para ponerlo en práctica, hemos escrito un código que implementa nuestra formulación del método, capaz de aprovechar la potencia de cálculo de los supercomputadores. El código es de acceso público y libre.

The Cosmic Microwave Background (CMB) encodes key information that directly connects the observations we make here and now with the state of the early universe. Moreover, according to our theoretical models, knowledge of the statistical properties of the CMB leads to knowledge about the characteristics of our Universe in the limit of the initial instants, and to predict its future. The relevant information provided by the CMB is encoded in its angular power spectrum. This study is devoted to the exploration of the Quadratic Maximum Likelihood Estimator (QML), an optimal method of estimating the power spectrum. We analyze its properties, the mathematical conditions that must be fulfilled to use it, solutions in the case that some of them are not fulfilled and we study the performance of the method in multiple situations of practical interest in the present and the immediate future. The QML method is computationally intensive. After a detailed analysis of the mathematical intricacies of the method, we have developed an optimal implementation that formally allows it to be applied with current technical means in situations that were unfeasible until now. To put it into practice, we have written a code that implements our formulation of the method, capable of taking advantage of the computing power of supercomputers. The code is publicly and freely available.

CONTENTS

LIST OF FIGURES

List of Tables

Chapter 1

Introduction

1.1 The cosmic microwave background

Once the expansion of the universe [1] was mostly accepted as an experimental fact, two cosmological visions radically different in their philosophical conceptions (*prejudices* could be more adequate) were formulated.

According to the steady state theory [2, 3] the universe is eternal and with properties that remain constant in time: the decrease of the density caused by the expansion is compensated by the constant creation of matter. This theory states that the cosmos has always been as we observe it in the present time.

According to the Big Bang theory [4, 5, 6, 7] the universe evolves with time. In this context, since in the present time the galaxies are moving apart, in the past all the elements that conform the universe must be closer and more concentrated. If we project this scheme up to a remote past and assume the principle that the laws of physic are the same for the whole universe, among the consequences that can be deduced, we find that there was an epoch in which matter was in the state of plasma in thermodynamic equilibrium with radiation. Under these conditions, it must follow the frequency distribution of a black body at the temperature of the thermodynamic fluid. Starting from that equilibrium state, the densities of matter and radiation and the temperature of the fluid must gradually decrease as a consequence of the expansion. At some point, the density of photons and their energy distribution fall below the threshold at which radiation keeps matter in the plasma state. As a consequence, the electrons are captured by the atomic nucleus, and matter and radiation decouple. According to the principle that there are no privileged observers in the universe, the wavelength of the decoupled radiation increases, at each place in the universe, at the same proportion as the expansion factor grows. In turn, the combination of the photon density decreasing and the growth of the wavelength drives the radiation to conserve a black body distribution. Therefore, after the decoupling, the nature of the frequency distribution of the radiation does not change with time, it just changes its temperature. According to this theory, a nearly homogeneous and isotropic radiation field with a black body spectral distribution and low temperature must be filling the universe at the present time: the Cosmic Microwave Background (CMB).

Therefore, originally the CMB was a prediction of the theoretical physics —year 1948 [4]— within the framework given by the Big Bang theory paradigm as a consequence of the observation of the expansion in the universe. It was detected for the first time in 1964 (in 1965, the detection was published in [8] and the interpretation as a Big Bang remnant in [9]). Currently, we can assert that the theoretically expected isotropy is a contrasted experimental fact (e.g. [10]) that shows a black body temperature of 2.7260 ± 0.0013 K [11]. On the other hand, since we can only observe the CMB from our planet and its immediate proximity, we assume homogeneity both as a consequence of the principle that there are no privileged observers and by consistency with other observables in cosmology.

The CMB pervades the entire universe, but we can only detect and measure the properties of radiation incident on us from a particular region, the surface of a sphere whose radius is the distance that photons have traveled since they last scattered at the recombination epoch; for obvious reasons, we call it the last scattering surface (LSS). The observed frequency distribution of the CMB over the surface shows that, indeed, it follows a black body spectrum at all points. However, the temperature of the CMB is not perfectly isotropic. The largest anisotropy observed in the CMB crosses diametrically the sky from a slightly warmer region to a slightly colder region in the opposite region of the sky. The amplitude of this *temperature dipole* is 3.3621 ± 0.0010 mK. The CMB temperature dipole is a consequence of the Doppler shift caused by the movement of the observer with respect to the CMB rest frame.[1] Removing the distortion caused by motion, we find that the temperature of the radiation is practically isotropic, although it presents deviations relative to the mean temperature of the order of 1 in 100 000. These anisotropies are directly connected with the state of the plasma at the moment of the decoupling, and this in turn to the physical laws and the initial and boundary conditions that took place at the origin of our universe. Therefore, it could be possible to obtain useful information about the physics of the early universe by observing the CMB.

---

[1] In the next section, we show how the CMB as seen by an observer at rest can be determined from the CMB measured by an observer in motion.

1.1.1 The CMB as seen from a reference frame at rest

Applying special relativity one can find the relationship between the temperature T of the radiation of a black body as seen by an observer at rest and the temperature T' as seen by another observer that moves with velocity v with respect to the first one (more details in [12] and [13])

$$T' = \frac{T}{\gamma(1-\beta\cos\theta')}, \tag{1.1}$$

where we find the usual γ factor of special relativity and the $\beta = v/c$ factor. The variable θ' is the angle between the direction of the point of the sky towards which the observer moves and that where the observer is measuring the temperature radiation T'.

If $v \ll c$, the last expression can be transformed into

$$T' = T(1+\beta\cos\theta'). \tag{1.2}$$

Assuming that the anisotropies caused by the Doppler effect are dominant, we can fit β if we assume that T is equal at all points. With this procedure, it is found that we move with respect to the CMB with a velocity close to 400 km s^{-1} (more precisely, in a detailed analysis taking into account the Doppler and aberration effects, *Planck Collaboration*[2] found a component in the dipole direction of 384 km s^{-1}± 78 km s^{-1} (stat.) ± 115 km s^{-1} (syst.) [13]). Finally, once we have the value of the velocity, we can compute T from T'. Figure 1.1 shows the map of the anisotropies of the CMB as seen by an observer at rest (details about the figure can be found in [15]).

1.2 The angular power spectrum

1.2.1 The scales of the anisotropies

Let us assume that we have removed the Doppler effect in the observed CMB caused by our movement and that we have the data of the temperature $T(\hat{r})$ of the CMB at each point $\hat{r}$ of the sky. The deviation with respect to the mean value $T_0 = \langle T(\hat{r})\rangle$ is

$$\delta T(\hat{r}) = T(\hat{r}) - T_0. \tag{1.3}$$

Because of two fundamental reasons, it would not be appropriate to try to connect directly the specific observed values with the physical laws and with the parameters that characterize our universe. First, because by the principle that states that all the

---

[2] *Planck* was a European Space Agency mission [14] launched in 2009 that observed the CMB in a frequency range from 30 to 857 GHz over the whole sky with unprecedented resolution and sensitivity.

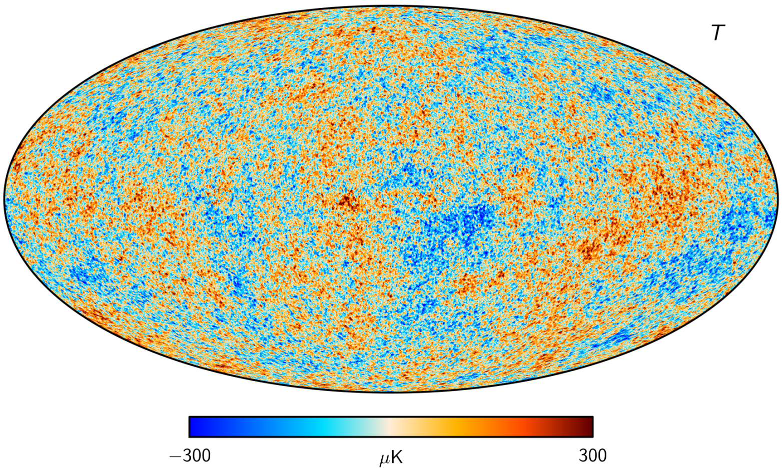


Figure 1.1: Full-mission cleaned temperature CMB map from the SEVEM pipeline, published in the 2018 *Planck release*. The signal shown in the map has been inpainted in the regions where in practice the foregrounds blur the CMB signal. The map is shown at 5 arcmin FWHM resolution (more information and figures can be found at https://wiki.cosmos.esa.int/planck-legacy-archive/index.php/CMB_maps)

points of the sky are equivalent, there is nothing in a specific pair of coordinates that gives them a special characteristic with respect to the other points. Second, because the values of the anisotropies at each point are given by random variables, and therefore to pretend to assign a specific value to the anisotropies of a collection of points would be something like trying to advance the sequence of numbers that will come out when rolling a dice several times. Therefore, the most convenient procedure is to translate the information contained in the anisotropies in pixel space into another format that could be connected directly with physics. Since the data is distributed over the surface of the sphere, it is reasonable to express it as an expansion in spherical harmonics

$$\delta T(\hat{r}) = \sum_{\ell m} a_{\ell m} Y_{\ell m}(\hat{r}). \tag{1.4}$$

Given the precise realization of the anisotropies that has emerged from our universe and the specific pattern that we measure from our position today, each coefficient $a_{\ell m}$ has a precise value. However, within the framework of the theory, ours is only one of the infinite possible realizations of the universe. Since the theory can only advance statistical predictions about the characteristics of the possible universes compatible with it and, furthermore, a fundamental element in its principles is that there are no privileged points or directions, we have

$$\langle a_{\ell m} \rangle_{\text{universes}} = 0. \tag{1.5}$$

For the moment, we have variables of zero mean, but if there are indeed anisotropies, their variances cannot be zero. Appealing again to the symmetries, if there are no privileged directions for the anisotropies, the average of the crossing of two different terms in the harmonic expansion has to cancel

$$\langle a_{\ell m} a^*_{\ell' m'} \rangle_{\text{universes}} \propto \delta_{\ell\ell'} \delta_{mm'}. \tag{1.6}$$

If the above were not correct and there were correlations between the different components of the spherical harmonic expansion, then there would be privileged zones on the surface of the sphere. On the other hand, since the index m refers to the orientation in space, fixed the index ℓ, by symmetry, the values of the auto-correlation of elements of different m must be equal

$$\langle a_{\ell m} a^*_{\ell m} \rangle_{\text{universes}} = \langle a_{\ell m'} a^*_{\ell m'} \rangle_{\text{universes}}. \tag{1.7}$$

Therefore, the only remaining non-null variances are the auto-correlation terms, and their values only depend on the index ℓ. From a theoretical point of view in which we have multiple realizations of likely universes according to the physical laws that we try to infer, we define the angular power spectrum of the anisotropies $\tilde{C}_\ell$ as the variances of the coefficients of the spherical harmonic expansion computed over the realizations

$$\tilde{C}_\ell = \langle a_{\ell m} a^*_{\ell m} \rangle_{\text{universes}}. \tag{1.8}$$

The ℓ index makes reference to the size of the oscillations of the Legendre polynomials and, in particular in our case, to the size of the contribution to the anisotropies of the spherical harmonic coefficients. Therefore, the power spectrum registers the power of the anisotropies in the different angular scales.

In practice, we only have one realization of the universe to observe and only one point and one instant to do it —our planet, its vicinity and the last decades—. In short, the averages over universes of the previous expressions are a theoretical entelechy, and we can only have access to the spherical harmonic coefficients of a single sample of CMB.

Therefore, we need an alternative definition of the power spectrum that can be used in practice. Assuming that the symmetry properties with respect to the orientation in the space of the anisotropies of a collection of universes are satisfied, we define the measurable power spectrum on the ℓ scale as the average in m of the absolute values of the coefficients of the expansion in harmonics, i.e.,

$$C_\ell = \langle a_{\ell m} a^*_{\ell m} \rangle_m. \tag{1.9}$$

The average of the last expression over universes, yields

$$\langle C_\ell \rangle_{\text{universes}} = \langle \langle a_{\ell m} a^*_{\ell m} \rangle_m \rangle_{\text{universes}} = \langle \langle a_{\ell m} a^*_{\ell m} \rangle_{\text{universes}} \rangle_m = \langle \tilde{C}_\ell \rangle_m. \tag{1.10}$$

Since $\tilde{C}_\ell$ is independent of m, we get

$$\langle C_\ell \rangle_{\text{universes}} = \tilde{C}_\ell. \tag{1.11}$$

1.2.2 The two-point correlation function

The power spectrum is directly related to the average over the surface of the sphere of the product of the temperature of pairs of points at a fixed angular distance, what is typically referred as the two-point correlation function. Given a realization and being $\hat{r}$ and $\hat{r}'$ two directions such that $\hat{r} \cdot \hat{r}' = \cos(\theta)$, the function is

$$\mathbf{C}(\theta) = \langle \delta T(\hat{r}) \delta T(\hat{r}')_{\hat{r}\cdot\hat{r}'=\cos(\theta)} \rangle_{\text{surface}} \tag{1.12}$$

For a continuous random field on the sphere, the correlation function is

$$\mathbf{C}(\theta) = \sum_\ell \frac{1}{4\pi} \left[\sum_m a_{\ell m} a^*_{\ell m} \right] P_\ell(\cos(\theta)). \tag{1.13}$$

That by eq. (1.9) yields

$$\mathbf{C}(\theta) = \sum_\ell \frac{2\ell+1}{4\pi} C_\ell P_\ell(\cos(\theta)). \tag{1.14}$$

Details about the discretized two-point correlation function can be found in [16].

1.2.3 Cosmic variance

If we could observe infinite universe realizations, in theory,[3] we could limitlessly reduce the statistical contribution to the error with which we measure the power spectrum. Since in practice we have only one realization, we must consider the theoretical statistical contribution to the error of this fact.

To determine the error, one must first know or assume the statistical nature of the fluctuations. Let us assume that the fluctuations are Gaussian. This is a common assumption for two main reasons. From a theoretical perspective, the models describing the origin of the fluctuations, which are closely related to the CMB anisotropies, are expected to be Gaussian since they are caused by quantum fluctuations stretched up to

---

[3] Leaving aside technical and experimental issues.

cosmological scales. From an experimental point of view, the data also seem to indicate that the CMB anisotropies follow a Gaussian distribution. As an added advantage, the mathematical apparatus associated with the study of the CMB is simpler when assuming Gaussianity with respect to other possibilities regarding its statistical nature.

In this context, the coefficients of the expansion in spherical harmonics are also Gaussian random variables. Our goal is to find the variance of the power spectrum measured from one realization (eq. (1.9)). In more detail

$$C_\ell = \frac{1}{2\ell+1}\sum_{m=-\ell}^{\ell} |a_{\ell m}|^2. \tag{1.15}$$

To find the statistical error with which we can compute the power spectrum of a given realization, we need to resort to the variance of the $a_{\ell m}$ coefficients assuming infinite realizations, i.e., $\tilde{C}_\ell$. By dividing both sides of the above expression by $\tilde{C}_\ell$ and rearranging some terms, we have

$$(2\ell+1)\frac{C_\ell}{\tilde{C}_\ell} = \sum_{m=-\ell}^{\ell} \frac{|a_{\ell m}|^2}{\tilde{C}_\ell}. \tag{1.16}$$

Note that at the right side of the equality we have the sum of the variances of independent Gaussian random variables of zero mean and unit variance. Therefore, on the left side of the equality, we have a variable that follows a χ^2 distribution with $2\ell+1$ degrees of freedom.[4] Thus, we have the variance

$$\mathrm{Var}\left((2\ell+1)\frac{C_\ell}{\tilde{C}_\ell}\right) = 2(2\ell+1). \tag{1.17}$$

Finally, we can solve from the last expression what we call *cosmic variance*, the variance of the power spectrum

$$\mathrm{Var}(C_\ell) = 2(2\ell+1)\frac{\tilde{C}_\ell^2}{(2\ell+1)^2} = \frac{2}{2\ell+1}\tilde{C}_\ell^2. \tag{1.18}$$

It is customary to write the expression of the cosmic variance in the form of the expected statistical error of the values of the power spectrum obtained from one realization

$$\Delta C_\ell = \frac{\tilde{C}_\ell}{\sqrt{(2\ell+1)/2}}. \tag{1.19}$$

---

[4] Actually we do not have $2\ell+1$ independent $a_{\ell m}$ variables because the $a_{\ell m}$'s of $m<0$ are not independent of those of $m>0$, but since they are complex quantities, counting real and imaginary parts it is true that we have $2\ell+1$ independent variables.

In short, as a result of having only one realization of the universe, for statistical reasons, the values of the power spectrum are affected by an error given by the expression (1.19). In practice, the error can be increased by other effects, such as not having data over the full-sky or the presence of instrumental noise.

The way the cosmic variance depends on ℓ causes the error in C_ℓ to increase as ℓ decreases –larger error in larger scales–. This is an effect due to the fact that the average by which the power spectrum is calculated counts with fewer elements for smaller ℓ.

It is interesting to mention that the value of the cosmic variance, eq. (1.19), corresponds to the case in which we have only one realization of the universe. Although this does not have practical effects, if we could have access to data from n independent samples, the value of the cosmic variance would be

$$\Delta C_\ell^{n\,\mathrm{maps}} = \frac{\tilde{C}_\ell}{\sqrt{n(2\ell+1)/2}}. \tag{1.20}$$

1.3 Polarization

When an electromagnetic wave is scattered by a free electron there is a change in the direction of wave propagation. Since the electric and magnetic fields oscillate in a plane perpendicular to the propagation direction, the oscillation amplitude in the plane given by the incident and the outgoing directions is reduced after scattering, while the amplitude in the direction perpendicular to the plane remains unchanged. Figure 1.2 illustrates this process, usually referred to as Thomson scattering when the photons are of low energy. In an extreme case, when the two directions of propagation become perpendicular, the wave becomes completely polarized.

However, polarized radiation cannot be produced in an isotropic medium with abundant scattering because, by symmetry, there are no privileged directions; thus, the combination of the polarization induced in each direction of propagation will tend to cancel each other out. That is, polarized radiation can only be produced in an anisotropic fluid. A detailed analysis shows that the quadrupole term of the Fourier expansion of the fluid density acts as the source of the CMB polarization (for more details on polarization see [17, 18, 19]). The underlying idea is that when unpolarized radiation coming from two perpendicular directions with different intensity —anisotropic radiation— incides on a point and, for simplicity, assuming that both waves are scattered in the direction perpendicular to the plane given by the incident directions, Thomson scattering will eliminate the incident oscillations of the electric and magnetic fields $\vec{E}$ and $\vec{B}$ oriented in the direction of scattering and will let pass the os-

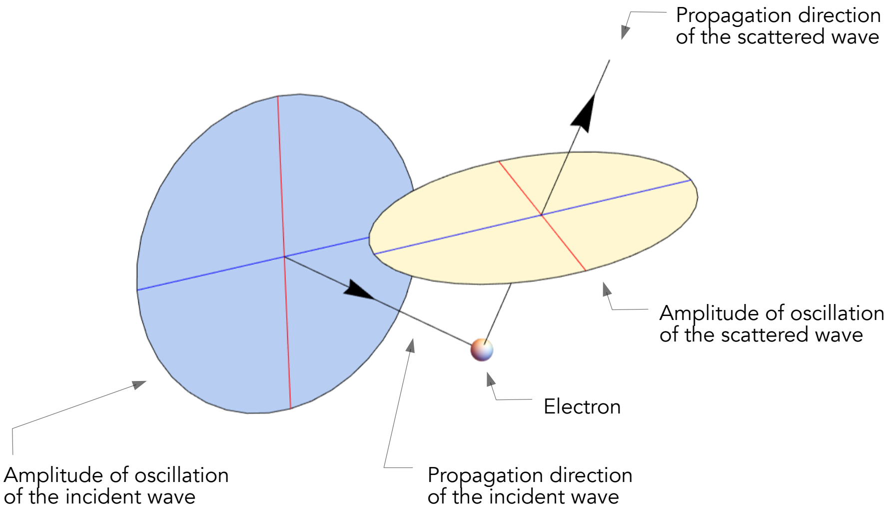


Figure 1.2: Thomson scattering. The incident wave is unpolarized (in the figure, the amplitude space is disk-shaped) and the deflected wave is partially polarized (the disk becomes an ellipse). The two blue semi-axes —perpendicular to the two propagation directions— have the same length in the disk and in the ellipse. The red semi-axis —which is in the plane given by the two directions of propagation— is shorter in the ellipse than in the disk. When the two directions of propagation are perpendicular, the minor semi-axis of the ellipse is of zero length and the wave is completely polarized.

cillations in the other two directions. Since we are assuming anisotropies, the radiation coming from each of the two directions is of different intensity, so the amplitudes in the fields of the filtered oscillations in one direction will be higher than the amplitudes in the other direction, and the scattered radiation will be polarized.

On the other hand, even having quadrupole-type anisotropies, a continuous scattering constantly creates and erases polarization. The polarization that we can observe at present can only be the trace of the anisotropies in the moments prior to —or simultaneous with— the decoupling between radiation and matter, when a large part of the electrons have already been captured by the atomic nuclei, so the amplitude of the CMB polarization is expected to be smaller by about an order of magnitude than the fluctuations of temperature [20].

The radiation temperature is a scalar and its description in terms of the spherical harmonics is as simple as shown in eq. (1.4). The description of polarization is a bit more complicated.

Polarization is usually represented by the well-known Stokes parameters, the set of observables I, Q, U and V. Suppose that we have an electromagnetic wave propagating in the z-direction. The electric field of the wave can oscillate in the $x-y$ plane. Decomposing it in these two directions, we get

$$E_x(t) = a_x(t)\cos(2\pi\nu t - \theta_x(t)) \qquad E_y(t) = a_y(t)\cos\left(2\pi\nu t - \theta_y(t)\right) \tag{1.21}$$

In the last expression, the amplitudes on each axis, the frequency and the phases can be identified. The Stokes parameters are defined (for simplicity, the time dependence is not explicitly shown) as

$$\begin{aligned} I &\equiv \langle a_x^2\rangle + \langle a_y^2\rangle & \qquad Q &\equiv \langle a_x^2\rangle - \langle a_y^2\rangle \\ U &\equiv \langle 2a_x a_y \cos(\theta_x - \theta_y)\rangle & \qquad V &\equiv \langle 2a_x a_y \sin(\theta_x - \theta_y)\rangle, \end{aligned} \tag{1.22}$$

where the averages are computed over time, extending over a time much longer than $1/\nu$.

The parameter I is always positive and describes the intensity of the radiation. When the E_x and E_y components are uncorrelated, the radiation is unpolarized and the averages of Q, U and V cancel.

The parameter V depends on the sine of the phase shift between the x and y components. Since the sine takes its maximum value at the argument $\pi/2$, V is favoured when the components are shifted ninety degrees, which means circular polarization. Due to the characteristics of the physical process, Thomson scattering does not produce this type of polarization, so, as far as the CMB is concerned, V is zero. Figure 1.3 shows the CMB Q and U maps of the 2018 version of the *Planck* collaboration [15].

If we fix the direction of propagation on the z-axis and rotate the x and y axes by an angle ψ, such that $\mathbf{e}_{x'} = \mathbf{e}_x \cos\psi + \mathbf{e}_y \sin\psi$, $\mathbf{e}_{y'} = -\mathbf{e}_x \sin\psi + \mathbf{e}_y \cos\psi$, the $a_x(t)$ and $a_y(t)$ components of the oscillation amplitude will change in the same way as the $\mathbf{e}_{x'}$ and $\mathbf{e}_{y'}$ vectors do. Due to the existence of the products of the sine and cosine functions of ψ in the averages defining Q and U, referred to the new axes, the functions sine and cosine with a 2ψ argument appear in the computation of Q and U, specifically

$$Q' = Q\cos(2\psi) + U\sin(2\psi) \qquad U' = -Q\sin(2\psi) + U\cos(2\psi). \tag{1.23}$$

From the above, we find that two quantities of spin ±2 can be built from Q and U

$$(Q \pm iU)' = e^{\mp 2i\psi}(Q \pm iU). \tag{1.24}$$

Therefore, we can write an expression related to the elements of the CMB polarization as an expansion on spherical harmonics, in this case of spin +2 and -2

$$\begin{aligned} (Q+iU)(\hat{r}) &= \textstyle\sum_{\ell m} a_{2,\ell m}\, {}_2Y_{\ell m}(\hat{r}) \\ (Q-iU)(\hat{r}) &= \textstyle\sum_{\ell m} a_{-2,\ell m}\, {}_{-2}Y_{\ell m}(\hat{r}), \end{aligned} \tag{1.25}$$

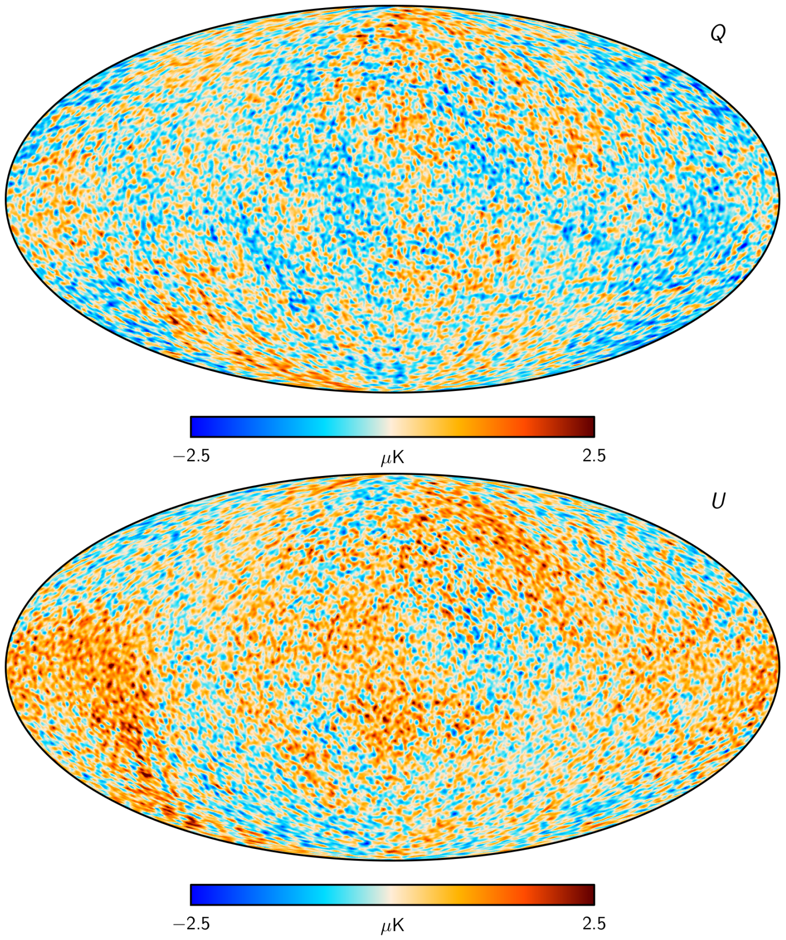


Figure 1.3: Full-mission cleaned Q and U CMB polarization maps from the SEVEM pipeline, published in the 2018 *Planck release*. The signal shown in the maps has been inpainted in the regions where in practice the foregrounds blur the CMB signal. In order to suppress instrumental noise, the maps are shown at 80 arcmin FWHM resolution (more information and figures can be found at https://wiki.cosmos.esa.int/planck-legacy-archive/index.php/CMB_maps)

where Q and U are measured with respect to the unit vectors of the specific coordinate system located at the point on the surface of the sphere in the direction of observation.

Instead of $a_{2,\ell m}$ and $a_{-2,\ell m}$ is convenient to use lineal combinations

$$\begin{aligned} a_{E,\ell m} &= -(a_{2,\ell m} + a_{-2,\ell m})/2 \\ a_{B,\ell m} &= -(a_{2,\ell m} - a_{-2,\ell m})/2 \end{aligned} \tag{1.26}$$

The Q and U quantities in eq. (1.25) can be separated by mean of linear combinations, thus getting

$$\begin{aligned} Q(\hat{r}) &= \sum_{\ell m} -a_{E,\ell m}X_{1,\ell m}(\hat{n}) - i a_{B,\ell m}X_{2,\ell m}(\hat{n}), \\ U(\hat{r}) &= \sum_{\ell m} i a_{E,\ell m}X_{2,\ell m}(\hat{n}) - a_{B,\ell m}X_{1,\ell m}(\hat{n}), \end{aligned} \tag{1.27}$$

where $X_{1,\ell m}$ and $X_{2,\ell m}$ are a combination of the $s=\pm 2$ spin-weighted harmonics

$$\begin{aligned} X_{1,\ell m}(\hat{r}) &= ({}_2Y_{\ell m} + {}_{-2}Y_{\ell m})/2, \\ X_{2,\ell m}(\hat{r}) &= ({}_2Y_{\ell m} - {}_{-2}Y_{\ell m})/2. \end{aligned} \tag{1.28}$$

Therefore, eq. (1.4) and (1.27) define the quantities $a_{T,\ell m}$, $a_{E,\ell m}$ and $a_{B,\ell m}$, that describe in harmonic space the temperature and polarization of the CMB maps.

Even if we have a model describing the physics of the universe, if it includes the principle of non-existence of privileged directions, we cannot predict the values of $a_{T,\ell m}$, $a_{E,\ell m}$ and $a_{B,\ell m}$. But we can predict the angular power spectrum in the framework of a model that includes this symmetry principle. The statistical isotropy determines for the six quadratic quantities

$$\begin{array}{ll} \langle a^*_{T,\ell m} a_{T,\ell' m'}\rangle = C_\ell^{TT}\delta_{\ell\ell'}\delta_{mm'} & \langle a^*_{E,\ell m} a_{E,\ell' m'}\rangle = C_\ell^{EE}\delta_{\ell\ell'}\delta_{mm'} \\ \langle a^*_{B,\ell m} a_{B,\ell' m'}\rangle = C_\ell^{BB}\delta_{\ell\ell'}\delta_{mm'} & \langle a^*_{T,\ell m} a_{E,\ell' m'}\rangle = C_\ell^{TE}\delta_{\ell\ell'}\delta_{mm'} \\ \langle a^*_{T,\ell m} a_{B,\ell' m'}\rangle = C_\ell^{TB}\delta_{\ell\ell'}\delta_{mm'} & \langle a^*_{E,\ell m} a_{B,\ell' m'}\rangle = C_\ell^{EB}\delta_{\ell\ell'}\delta_{mm'}, \end{array} \tag{1.29}$$

where the averages are computed over the set of universe realizations. Figure 1.4 shows a summary of current measurements of the TT, EE and BB CMB power spectra. The principal goal of this thesis is the estimation of the power spectrum of the anisotropies from maps of temperature and polarization.

Because of parity considerations, the cross-correlation between B and the other two quantities vanishes. Therefore, if the sources of the anisotropies were invariant under parity inversion, the components C_ℓ^{TB} y C_ℓ^{EB} are expected to cancel. If non zero values of some of these components were measured that could not be explained by the presence of spurious signals such as instrumental effects or foregrounds contamination, we would be finding exotic effects indicating beyond-standard-model physics (e.g. [23, 24, 25, 26]).

1.4 The sources of the CMB anisotropies

As a first approximation, the CMB blackbody radiation is isotropic. However, it shows small local deviations, which give rise to the pattern of anisotropies. The deviations

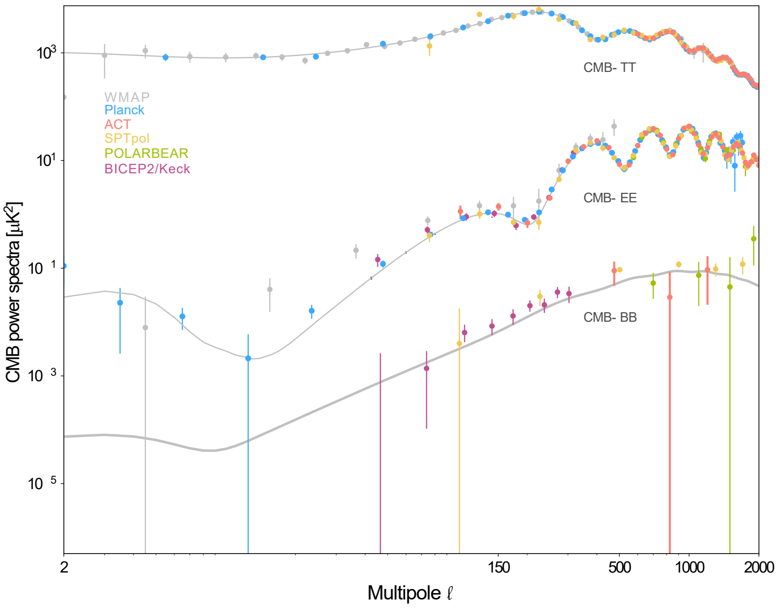


Figure 1.4: Summary of current measurements of CMB power spectra. Solid lines correspond to the *Planck* best-fit model, but assuming r=0.004 [21], while points give measurements from different experiments. Figure adapted from [22], where details and references can be found.

from the mean value are caused by local perturbations in the fluid density, which have many consequences on the spatial distribution of the CMB [27, 28, 29]. First, differences in fluid density cause the equilibrium temperature to be different at different points. Second, density differences introduce dynamics in the fluid leading to material displacements and oscillations, which introduces a Doppler shift in the radiation. Finally, density differences introduce inhomogeneities in the gravitational potential. In general, it can be said that photons loose energy when they leave the potential wells and gain energy when they fall into them; as a result, the difference in potentials at the point where the photon was emitted and at the point where the observer is located introduces shifts in the temperature of the radiation. Since all the photons we observe fall to the same potential, the one we inhabit, the anisotropies associated with this effect depend only on the potentials at the LSS at the directions from which they arrive. On the other hand, the potentials that photons pass through on the way from the LSS to the observer may vary as a function of the time from the time the photon enters the well until it leaves it, which introduces new elements in the temperature variation. The following expression shows the observed temperature fluctuation measured from the radiation coming from the direction given by the unitary vector $\vec{r}$ and accounts for all these effects

$$\frac{\Delta T}{T}(\vec{r}) \approx \frac{1}{4}\frac{\delta\rho_\gamma}{\rho_\gamma}(\vec{r}) + \vec{r}\cdot(\vec{v}_o - \vec{v}_e) + \frac{\phi_e(\vec{r}) - \phi_o}{3} + \int_e^o \frac{\partial\phi}{\partial t} dt. \tag{1.30}$$

The first term corresponds to the intrinsic fluctuations in the density of the plasma located in the direction $\vec{r}$ on the LSS and it is connected to the fact that the energy density of a black body is proportional to T^4. The second term is the Doppler effect. The third and fourth terms give account of the gravitational redshift: the contribution of the local potential at the emission point (Sachs-Wolfe effect) and the contribution of the variation with time of the potentials the photons are going through (integrated Sachs-Wolfe effect, ISW), respectively [27]. The first three terms are related to properties on the LSS, and we usually refer to them as *primary anisotropies*. The last term belongs to the category of *secondary anisotropies*, which are related to the properties of the points in the path between the emission and absorption points.[5]

There are other sources of secondary anisotropies, such as the Rees-Sciama effect [30], the Ostriker–Vishniac effect [31], the deviation of the photons caused by gravitational lensing, that redistribute the power and smooth the sharp features in the CMB power spectrum [32], the modification of the black body spectrum due to the scattering of CMB photons by electrons on regions of hot gas, the thermal Sunyaev-Zeldovich effect [33] and the kinetic Sunyaev-Zeldovich effect [34], that appears when photons pass through a plasma with bulk velocity, which introduces also Doppler shifts in the CMB. There are other dispersive processes that affect the anisotropies due to the scattering of photons by free electrons after the reionization of the universe [35].

What we observe from the CMB is consistent with anisotropies caused by local density perturbations affected by secondary anisotropies, something that can be considered as an established fact, so what remains is to explain the causes of these perturbations.

1.5 The inflationary paradigm

The inflationary models appear as a theoretical solution to some of the classical problems in Cosmology. Two issues stand out in particular. First, the fact that we can observe very distant regions in the sky that, in principle, have not been in causal contact, but that have very similar properties. A condition that can not be easily explained, unless we impose some very specific initial conditions in the early universe.

---

[5] There are actually two contributions to the ISW effect, the early one, that occurs around recombination, and the late one, after recombination. The early one is in essence a primary anisotropy.

The second critical issue is the observed flatness of the universe at present time, which leads to a very fine tuning of the values of some cosmological parameters in the past, in order to be able to observe the current values. This, again, leads to a very unsatisfactory explanation. Both issues lead us to make arrangements to make our theories conform to what we observe.

Of course, as it always happens, to explain certain facts one must assume some hypotheses, which in this case are embedded in elements of particle physics. Besides solving the mentioned issues, the inflationary models provide a natural explanation of the inhomogeneities in the gravitational potentials, that in turn are the sources of the CMB anisotropies. When inflation comes into play, they become a consequence of basic physical principles and can also be easily characterized statistically. As a bonus, inflation also predicts a background of primordial gravitational waves that would leave its imprint in the B-mode of CMB polarization, which may be detectable with current technology. That is, inflation explains known facts and predicts new ones, a very valuable circumstance.

The inflationary model introduces a quantum field, the inflaton field, whose energy in the initial moments of the Big Bang —even before nucleosynthesis— dominated over the other components of energy of the universe. Basically, it can be said that the field evolved towards a lower energy state and, while it did so, quantum oscillations were produced locally in all places of the universe. Under certain conditions, the scale factor of the universe evolves during inflation growing exponentially. Meanwhile, the quantum oscillations get stretched and its physical size is increased —in the sense of how long it takes for light to travel through them— and, simultaneously, the field causing inflation oscillates quantumly in the points of the new physical space.

One of the results is that regions of space small enough to have interacted and reached equilibrium before inflation end up being stretched to sizes that currently exceed that of the observable universe. This explains the homogeneity in the global properties that we observe at present in areas of the universe at distances that make it impossible for them to have been in causal contact without resorting to inflation.

After being stretched by a superluminal expansion up to the point of exceeding the causal contact limit, the quantum fluctuations stop evolving physically and end up frozen. Once inflation has ended and the exponential growth stops, the horizon of light starts to grow over time, such that regions start to enter again within the horizon; the physics starts to play again and the regions that have reached causal contact start evolving together.

As the field fluctuated quantumly as it was stretched, the new points of space generated by inflation in turn fluctuated quantumly; the final result of inflation, which we now observe, is a space with the same properties at first order at all places —homogeneity— but that shows local deviations. The global and homogeneous contribution to the local properties is derived from those of the small zone of the universe in equilibrium before inflation started, that grew until it became the observable universe we see; the local deviations are the effect of the sum of successive layers of quantum fluctuations stretched up to various scales while the universe was inflating. The intersection of the evolution of the local deviations in the three-dimensional space with the last scattering surface is the origin of the anisotropies we see at the present.

In the span of time since inflation ended, the physical interaction has introduced dynamics on the small scales and has altered the trace of inflation at those scales. But the large scales remain out of causal contact and the inflation trail remains frozen there. Certain inflationary models predict very flat spectrum for low ℓ, in the sense $\ell(\ell+1)C_\ell^{TT} \sim$ const. This prediction has been confirmed by CMB anisotropy measurements.

In inflationary models, the fluctuations in the field introduce fluctuations in the space-time metric. Furthermore, since the origin of the fluctuations is quantum, the perturbations are Gaussian. The perturbations of the metric can be decomposed as the contribution of three components: scalar, vector and tensor, that evolve independently in the linear regime. Vector perturbations are not created during inflation [36] (and in any case decay with the expansion of the universe) but inflation does produce the other two type of perturbations. The scalar component couples with the gravitational field and induces density fluctuations. The tensor component produces gravitational waves.

Regarding the effects on the CMB, both scalar and tensor perturbations induce E mode. In theory, the contribution to E of each of the components could be differentiated by analyzing the variation with ℓ of the power spectrum and correlating it with the forms of the contributions to the spectrum of the scalar and tensor components predicted by the models. However the foregoing procedure presents serious difficulties, because of degeneracies in the cosmological parameters with respect to the contribution of scalar and tensor components to the E-mode.

In linear theory the scalar perturbations produce E-mode of polarization only. However, tensor perturbations produce also B-mode. Since we only have a theoretical mechanism that gives rise to the primordial B-mode of CMB polarization, a detection would represent a very powerful evidence in favor of the inflationary theory. Moreover, the measurement of the ratio between the power of the tensorial and the scalar perturbations, known as the parameter r, would allow one to determine essential characteristics of the inflationary field.

In short, the situation is as follows: the inflationary theory explains, among others, the flatness problem and of the homogeneity at large scales of the universe, it also explains the origin of the anisotropies; besides, it predicts other effects that could be observed in the CMB. If the effects of gravitational waves were detected, it would be a great support for the theory. Conversely, the primordial B-mode may never be detected. If this were the case, one could not be sure whether the theory is not right or the power of the tensor modes is too low to be detected.

1.6 The cosmological parameters

According to current knowledge, from what we observe in the present and the physical theories, the universe can be described by a model that contains several parameters whose values cannot be determined by the theory and that, instead, we have to determine experimentally.

The Standard Cosmological Model is based on the cosmological principle. It assumes that the dynamics of the universe is described by general relativity and that there were initial deviations from homogeneity that gave rise later to the anisotropies in the CMB and were the germ of the large-scale structures we observe. The model includes dark matter in the form of non-relativistic particles as an element to explain the excess gravity that we cannot account for from ordinary observable matter. In addition, it includes a dark energy term as a driver of the acceleration of the expansion of the universe. We refer to a model that describes the universe on the basis of these ingredients as the Standard Cosmological Model, the Concordance Model or the ΛCMD model. In this context, inflation is considered to be the preferred mechanism for introducing the initial perturbations. In particular, the observed CMB spectra are well described by a standard spatially-flat six-parameter ΛCMD cosmology with a power-law spectrum of adiabatic scalar perturbations [21, 37].

The cosmological model parameters can be sorted into three classes: some of them describe the spacetime background (its content and current kinematics); others, the initial conditions, and there is a set of miscellaneous parameters.

Current estimations of the cosmological parameters obtained from *Planck* data are shown in table 1.1. The parameter h characterizes the current kinematic state of the universe, it specifies the Hubble constant in units of 100 km s^{-1} Mpc^{-1}. The parameter Ω_b is the ratio to the critical density of the mass density given by the existing parti-

Parameter	TT,TE,EE+lowE+lensing 68% limits
$\Omega_b h^2$	0.02237 ± 0.00015
$\Omega_c h^2$	0.1200 ± 0.0012
$100\theta_*$	1.04110 ± 0.00031
τ	0.0544 ± 0.0073
$\ln(10^{10} A_s)$	3.044 ± 0.014
n_s	0.9649 ± 0.0042

Table 1.1: Parameter constraints assuming the ΛCDM cosmology with a power-law initial spectrum, no tensors, spatial flatness, a cosmological constant as dark energy, and the sum of neutrino masses fixed to 0.06 eV. The legend of the second column makes reference to the data used to fit the parameters: *Planck* primary CMB data plus the *Planck* measurement of CMB lensing (Table 2, column 5 of [21]).

cles of the Standard Model (it is traditional in Cosmology to refer to both leptons and baryons simply as baryons). The parameter Ω_c is the ratio of the dark matter density to the critical density. The parameter θ_* is the angular scale of the sound horizon at recombination, a measure of the distance a sound wave has traveled through the plasma since inflation ended to the instant of decoupling. The parameter τ, optical depth, is the probability that a CMB photon undergoes a scattering process on its travel from the LSS to the point where it is captured by a detector. The parameters A_s and n_s are related to the initial conditions of the perturbations. Assuming purely adiabatic scalar perturbations at very early times, the curvature power spectrum follows a power law; A_s is the (dimensionless) base amplitude and n_s, the spectral index, is the exponent.

The data are consistent with a flat universe in accelerated expansion with a Hubble constant $H_0 = (67.36 \pm 0.54)$ km s^{-1} Mpc^{-1}, matter density parameter $\Omega_m = 0.3153 \pm 0.0073$ and a dark energy parameter $\Omega_\Lambda = 0.6847 \pm 0.0073$. On the other hand, they provide strong support to density perturbations caused by the simplest inflationary models: adiabatic, Gaussian and nearly-scale invariant perturbation and spatial flatness [37]. Referred to primordial gravitational wave signals, the data establish an upper limit to the tensor-to-scalar ratio of $r_{0.05}<0.032$ at 95% CL [38].

1.7 Summary and next steps

As noted in the previous sections, CMB anisotropies are related to plasma density perturbations during the decoupling process. These, in turn, are related to the distribution of perturbations of the metric resulting from inflation. Of course, once that inflation ended, and up to the time of decoupling (and further), the density fluctuations that re-entered the causality horizon evolved physically and did so following well-known physical laws. The specific details of the type of perturbations and their probability distribution after inflation ended —and those of the subsequent evolution— depend, according to our theory, on the cosmological parameters, that encapsulate the characteristics of our universe. As the perturbations are connected with the pattern of CMB anisotropies, it can be expected that a precise knowledge of the strength of the anisotropies at different angular scales can help to gain crucial insight into essential aspects of the nature of our universe. In addition, it should be mentioned that, as noted, the detection of the primordial polarization B-mode would be a special indicator of inflation.

Assuming that the fluctuations are Gaussian, all the information about the probability distribution of the anisotropies is encapsulated in the two-point correlation function or, equivalently, in the angular power spectrum. Therefore, it is clear that the determination of the power spectrum of the anisotropies of our universe is essential to characterize it. Indeed, it is one of the most valuable observables that nature can offer us.

Unfortunately, the precise knowledge of the power spectrum does not lead to a unambiguous knowledge of the cosmological parameters since, roughly speaking, different sets of values of the cosmological parameters can lead to the same power spectrum, but it is still an essential source of knowledge. On the other hand, if the anisotropies were not of Gaussian nature, the power spectrum would not contain all the information, although it would still remain as a crucial source of data.

The next sections will be dedicated to analyze our options to determine the CMB power spectrum.

The context and the goal

The pattern of anisotropies that we can observe comes from the only realization that we have available from a probability distribution that is fixed by the physical laws that govern our universe and that would govern the other infinite number of possible realizations that could occur instead of ours. That is, the situation is as follows: our universe exists and has physical laws. The laws that are derived from its observation indicate that our universe is a particular case among the infinite possibilities, whose possible existence is determined by laws of probability. From observing this available case, we try to infer the laws and the parameters of those laws, which govern all of them. As with all probability laws, information about the general law can be inferred from a sample of realizations, but the inference is accompanied by an intrinsic statistical uncertainty that depends on the size of the sample and the characteristics of the distribution.

Therefore, at this point, our goal is, from the information encoded in the anisotropies of a particular realization and of a particular region, the LSS, to find the power spectrum leading to the general statistical law, and its uncertainty.

As shown in Section 1.2.3, we know that the uncertainty will be at least as large as the cosmic variance. But, since we have assumed a complete knowledge of the $a_{\ell m}$ of the CMB map to deduce it and this is certainly not guaranteed, we can expect the uncertainty to be greater than the cosmic variance.

In the following sections, we will work on the possibilities of determining the power spectrum and the degree of accuracy with which this can be done. But before we continue, let us introduce some mathematical tools that we will need in our study of CMB.

1.8 A mathematical interlude

In this section, we will introduce certain expressions and mathematical tools that will be useful in the rest of this work. To do so, we will delve into the connection between pixel and harmonic spaces and analyze the form and structure of the covariance matrices of the maps in both spaces.

1.8.1 Connection between pixel and harmonic spaces

Equations (1.4) and (1.27) show the relationship between the data expressed in the pixel space (T, Q and U) and in the harmonic space ($a_{T,\ell m}$,$a_{E,\ell m}$ and $a_{B,\ell m}$). Arranging the observed N_{pix} pixels of a CMB signal map in a signal vector $\mathbf{s}$ in pixel space of length $3N_{\text{pix}}$, being the first N_{pix} elements of the vector the values of the map in intensity, the next N_{pix} elements the values of Q and finally the values of U, eq. (1.4) and (1.27) can be expressed in a matrix form as

$$\mathbf{s} = \mathbf{Y}\tilde{\mathbf{s}}, \tag{1.31}$$

where the vector signal $\tilde{\mathbf{s}}$ in harmonic space is a vector of elements $a_{T,\ell m}$, $a_{E,\ell m}$ and $a_{B,\ell m}$ of size $3L$ (where $L = \sum_{\ell=2,\ell_{\text{max}}}(2\ell+1)$). The matrix $\mathbf{Y}$ has dimensions $3N_{\text{pix}} \times 3L$ and has the structure given by

$$\begin{pmatrix} T \\ Q \\ U \end{pmatrix} = \begin{pmatrix} \mathbf{Y}_{TT} & 0 & 0 \\ 0 & \mathbf{Y}_{QE} & \mathbf{Y}_{QB} \\ 0 & \mathbf{Y}_{UE} & \mathbf{Y}_{UB} \end{pmatrix} \begin{pmatrix} a^T_{lm} \\ a^E_{lm} \\ a^B_{lm} \end{pmatrix}. \tag{1.32}$$

The elements of each of the columns of the block $\mathbf{Y}_{TT}$ are the values of each of the spherical harmonics $Y_{\ell m}$ evaluated at each of the pixels of the map. Similarly those of the columns of the blocks $\mathbf{Y}_{QE}$, $\mathbf{Y}_{QB}$, $\mathbf{Y}_{UE}$ and $\mathbf{Y}_{UB}$ are related to the spherical harmonics $X_{1,\ell m}$ and $X_{2,\ell m}$, as shown in eq. (1.27).

The covariance matrix $\mathbf{S}$ of the signal map in pixel space is related to the covariance matrix $\tilde{\mathbf{S}}$ in harmonic space as

$$\mathbf{S} \equiv \langle \mathbf{s}\mathbf{s}^t \rangle = \mathbf{Y}\langle \tilde{\mathbf{s}}\tilde{\mathbf{s}}^\dagger \rangle \mathbf{Y}^\dagger = \mathbf{Y}\tilde{\mathbf{S}}\mathbf{Y}^\dagger, \tag{1.33}$$

where † denotes the conjugate transpose of a matrix. The matrix $\tilde{\mathbf{S}}$ has dimension $3L \times 3L$, and it is constituted by 3×3 diagonal blocks of length L related to the auto and cross-spectra of the intensity and polarization

$$\tilde{\mathbf{S}} = \begin{pmatrix} \tilde{\mathbf{S}}_{TT} & \tilde{\mathbf{S}}_{TE} & \tilde{\mathbf{S}}_{TB} \\ \tilde{\mathbf{S}}_{TE} & \tilde{\mathbf{S}}_{EE} & \tilde{\mathbf{S}}_{EB} \\ \tilde{\mathbf{S}}_{TB} & \tilde{\mathbf{S}}_{EB} & \tilde{\mathbf{S}}_{BB} \end{pmatrix}. \tag{1.34}$$

For instance, the TT block represents the correlations between all the (ℓ, m) coefficients for intensity.

The off-diagonal elements of each block are zero since for an isotropic random field, as expected in the standard cosmological model, the values of the six different blocks of eq. (1.34) are given by eq. (1.29). In addition, in the standard cosmological model, we expect the TB and EB cross-correlations to vanish. Therefore, the corresponding four blocks of the $\tilde{\mathbf{S}}$ matrix are also zero.

Note also that since each block runs over all the values of (ℓ, m), we have $2\ell + 1$ repetitions of the corresponding C_ℓ^{XY}, up to a total of L elements in the diagonal of each block. Therefore, each block $\tilde{\mathbf{S}}_{XY}$ can be written as the sum over ℓ of C_ℓ^{XY} multiplying a diagonal matrix whose diagonal is made of zeroes except at $2\ell + 1$ positions that are occupied by ones. As an illustration, let us write $\tilde{\mathbf{S}}_{TT}$ in a very simplified case —that is of no cosmological interest— in which ℓ goes from 0 to 1

$$\tilde{\mathbf{S}}^{\mathrm{Toy}}_{TT} = \begin{pmatrix} C_0^{TT} & 0 & 0 & 0 \\ 0 & C_1^{TT} & 0 & 0 \\ 0 & 0 & C_1^{TT} & 0 \\ 0 & 0 & 0 & C_1^{TT} \end{pmatrix}, \tag{1.35}$$

then

$$\tilde{\mathbf{S}}^{\text{Toy}}_{TT} = C_0^{TT}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + C_1^{TT}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{1.36}$$

Then, we can define the easily generalizable matrices $\tilde{\mathbf{P}}_\ell$, such that

$$\tilde{\mathbf{S}}^{\text{Toy}}_{TT} = C_0^{TT}\tilde{\mathbf{P}}_0 + C_1^{TT}\tilde{\mathbf{P}}_1. \tag{1.37}$$

In a case of interest in cosmology, the matrix $\tilde{\mathbf{S}}^{TT}$ could be

$$\tilde{\mathbf{S}}_{TT} = \sum_\ell C_\ell^{TT}\tilde{\mathbf{P}}_\ell. \tag{1.38}$$

The rest of the blocks of the covariance matrix in harmonic space can be constructed in the same way

$$\tilde{\mathbf{S}}_{XY} = \sum_\ell C_\ell^{XY}\tilde{\mathbf{P}}_\ell. \tag{1.39}$$

The covariance matrix in the pixel space has the following structure

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_{TT} & \mathbf{S}_{TQ} & \mathbf{S}_{TU} \\ \mathbf{S}_{QT} & \mathbf{S}_{QQ} & \mathbf{S}_{QU} \\ \mathbf{S}_{UT} & \mathbf{S}_{UQ} & \mathbf{S}_{UU} \end{pmatrix}. \tag{1.40}$$

From eq. (1.33) and assuming $C_\ell^{TB} = C_\ell^{EB} = 0$, six of the blocks are given by

$$\begin{aligned}
\mathbf{S}_{TT} &= \mathbf{Y}_{TT}\tilde{\mathbf{S}}_{TT}\mathbf{Y}^\dagger_{TT} \\
\mathbf{S}_{TQ} &= \mathbf{Y}_{TT}\tilde{\mathbf{S}}_{TE}\mathbf{Y}^\dagger_{EQ} \\
\mathbf{S}_{TU} &= \mathbf{Y}_{TT}\tilde{\mathbf{S}}_{TE}\mathbf{Y}^\dagger_{EU} \\
\mathbf{S}_{QQ} &= \mathbf{Y}_{QE}\tilde{\mathbf{S}}_{EE}\mathbf{Y}^\dagger_{EQ} + \mathbf{Y}_{QB}\tilde{\mathbf{S}}_{BB}\mathbf{Y}^\dagger_{BQ} \\
\mathbf{S}_{QU} &= \mathbf{Y}_{QE}\tilde{\mathbf{S}}_{EE}\mathbf{Y}^\dagger_{EU} + \mathbf{Y}_{QB}\tilde{\mathbf{S}}_{BB}\mathbf{Y}^\dagger_{BU} \\
\mathbf{S}_{UU} &= \mathbf{Y}_{UE}\tilde{\mathbf{S}}_{EE}\mathbf{Y}^\dagger_{EU} + \mathbf{Y}_{UB}\tilde{\mathbf{S}}_{BB}\mathbf{Y}^\dagger_{BU},
\end{aligned} \tag{1.41}$$

where the matrices $\tilde{\mathbf{S}}_{XY}$ are the —diagonal— blocks of matrix $\tilde{\mathbf{S}}$, and the remaining blocks $\mathbf{S}_{QT}$, $\mathbf{S}_{UT}$ and $\mathbf{S}_{UQ}$ are the transpose of their symmetric partners of $\mathbf{S}$.

Let us focus on $\mathbf{S}^{TT}$. By eqs. (1.41) and (1.38) we get

$$\mathbf{S}_{TT} = \sum_\ell C_\ell^{TT}\mathbf{Y}_{TT}\tilde{\mathbf{P}}_\ell\mathbf{Y}^\dagger_{TT}. \tag{1.42}$$

Therefore, we can define the matrices in the pixel space equivalent to the matrices $\tilde{\mathbf{P}}_\ell^{TT}$

$$\mathbf{P}_\ell^{TT} = \mathbf{Y}_{TT}\tilde{\mathbf{P}}_\ell\mathbf{Y}_{TT}^\dagger. \tag{1.43}$$

Because of the structure of $\tilde{\mathbf{P}}_\ell$, the product $\tilde{\mathbf{P}}_\ell\mathbf{Y}_{TT}^\dagger$ is a matrix made of zeroes except at $2\ell+1$ rows whose values are the conjugates of the $2\ell+1$ spherical harmonics of index ℓ. Then, the product $\mathbf{Y}_{TT}\tilde{\mathbf{P}}_\ell\mathbf{Y}_{TT}^\dagger$ is a matrix of size $N_{\text{pix}} \times N_{\text{pix}}$ whose values are, for each pair of pixels

$$\mathbf{P}_\ell^{TT}(ij) = \sum_{m=-\ell}^{\ell} Y_{\ell m}(\hat{r}_i)Y_{\ell m}^*(\hat{r}_j) \tag{1.44}$$

By the addition theorem of the spherical harmonics, we have

$$\mathbf{P}_\ell^{TT}(ij) = \frac{2\ell+1}{4\pi}P_\ell(\hat{r}_i \cdot \hat{r}_j), \tag{1.45}$$

where P_ℓ are the Legendre polynomials. Therefore, the block $\mathbf{S}_{TT}$ of the covariance matrix is a sum in ℓ of the power spectrum multipole multiplied by a matrix of index ℓ

$$\mathbf{S}_{TT} = \sum_\ell C_\ell^{TT}\mathbf{P}_\ell^{TT}. \tag{1.46}$$

A similar reasoning will lead to equivalent conclusions for the blocks $\mathbf{S}_{TQ}$, $\mathbf{S}_{TU}$ (and their transposes)

$$\mathbf{S}_{TQ} = \sum_\ell C_\ell^{TE}\mathbf{P}_\ell^{TQ} \tag{1.47}$$

and

$$\mathbf{S}_{TU} = \sum_\ell C_\ell^{TE}\mathbf{P}_\ell^{TU}, \tag{1.48}$$

where the elements of the matrices $\mathbf{P}_\ell^{TQ}$ and $\mathbf{P}_\ell^{TU}$ can be computed in a similar way to those of $\mathbf{P}_\ell^{TT}$, eq. (1.44), but mixing the harmonics $Y_{\ell m}$ with $X_{1,\ell m}$ and $X_{2,\ell m}$.

The blocks that mix Q and U with Q and U can be found in the same way, for example

$$\mathbf{S}_{QQ} = \sum_\ell \left[C_\ell^{EE}\mathbf{P}_\ell^{QQ_{EE}} + C_\ell^{BB}\mathbf{P}_\ell^{QQ_{BB}}\right], \tag{1.49}$$

where two matrices of the kind $\mathbf{P}_\ell^{QQ}$ have been defined

$$\mathbf{P}_\ell^{QQ_{EE}} = \mathbf{Y}_{QE}\tilde{\mathbf{P}}_\ell\mathbf{Y}_{EQ}^\dagger \qquad \mathbf{P}_\ell^{QQ_{BB}} = \mathbf{Y}_{QB}\tilde{\mathbf{P}}_\ell\mathbf{Y}_{BQ}^\dagger \tag{1.50}$$

The two remaining blocks are

$$\mathbf{S}_{QU} = \sum_\ell \left[C_\ell^{EE}\mathbf{P}_\ell^{QU_{EE}} + C_\ell^{BB}\mathbf{P}_\ell^{QU_{BB}}\right] \tag{1.51}$$

and

$$\mathbf{S}_{UU} = \sum_{\ell} \Big[C_\ell^{EE} \mathbf{P}_\ell^{UU_{EE}} + C_\ell^{BB} \mathbf{P}_\ell^{UU_{BB}} \Big]. \tag{1.52}$$

Now we can move from the blocks of the covariance matrix expressed as a sum on ℓ to the whole matrix **S** expressed as a double sum on the components of the power spectrum and on ℓ. For example, let us focus on the component EE of the power spectrum. The multipoles values C_ℓ^{EE} appear as a factor on eq. (1.49), (1.51) and (1.52). Arranging the matrices $\mathbf{P}_\ell^{QQ_{EE}}$, $\mathbf{P}_\ell^{QU_{EE}}$, $\mathbf{P}_\ell^{UU_{EE}}$ and the transpose of $\mathbf{P}_\ell^{QU_{EE}}$ of size $N_{\mathrm{pix}} \times N_{\mathrm{pix}}$ in a matrix of the size of the covariance matrix in pixel space, $3N_{\mathrm{pix}} \times 3N_{\mathrm{pix}}$ in the position of the blocks QQ, QU, UU and UQ, we can define the new matrices $\mathbf{P}_\ell^{EE}$

$$\mathbf{P}_\ell^{EE} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_\ell^{QQ_{EE}} & \mathbf{P}_\ell^{QU_{EE}} \\ \mathbf{0} & (\mathbf{P}_\ell^{QU_{EE}})^t & \mathbf{P}_\ell^{UU_{EE}} \end{pmatrix} \tag{1.53}$$

Proceeding in a similar way with the matrices $\mathbf{P}_\ell^{XY_{BB}}$, that multiply coefficients of the class C_ℓ^{BB} in eq. (1.49), (1.51) and (1.52), the matrix $\mathbf{P}_\ell^{BB}$ can be defined. The matrices $\mathbf{P}_\ell^{TT}$ and $\mathbf{P}_\ell^{TE}$ can be defined in the same way. Finally, the matrix **S** can by written as[6]

$$\mathbf{S} = \sum_{\ell} \Big[C_\ell^{TT} \mathbf{P}_\ell^{TT} + C_\ell^{TE} \mathbf{P}_\ell^{TE} + C_\ell^{EE} \mathbf{P}_\ell^{EE} + C_\ell^{BB} \mathbf{P}_\ell^{BB} \Big]. \tag{1.54}$$

Finally, defining an index i that includes the component of the power spectrum and the index ℓ, that is, if $\ell = 2 \ldots \ell_{\mathrm{max}}$, i runs from $i = 1$ to $i = 4 \times (\ell_{\mathrm{max}} - 1)$, we get

$$\mathbf{S} = \sum_i C_i \mathbf{P}_i, \tag{1.55}$$

that is, the covariance matrices of intensity and of intensity and polarization are formally identical. Furthermore, the covariance matrix only depends on the angular power spectrum, and this dependence is linear. It is important to mention that this fact is independent of whether the fluctuations are Gaussian or not. It only depends on the absence of privileged directions.

On the other hand, the covariance matrix in harmonic space is formally identical

$$\tilde{\mathbf{S}} = \sum_i C_i \tilde{\mathbf{P}}_i, \tag{1.56}$$

but being the $\tilde{\mathbf{P}}_i$ matrices much simpler, since they consist of many zeros and a few ones in strategic positions. Obviously this form of the covariance matrix is simpler because the properties of the anisotropies in our theoretical model are simpler in this space.

---

[6] We are assuming that $C_\ell^{TB} = C_\ell^{EB} = 0$

The specific pattern of each matrix $\tilde{\mathbf{P}}_i$ depends on the component of the power spectrum and on ℓ. The matrices $\tilde{\mathbf{P}}_\ell$, of size $L \times L$, conform the blocks of the matrices $\tilde{\mathbf{P}}_i$, of size $3L \times 3L$, conveniently located. For example

$$\tilde{\mathbf{P}}_\ell^{TE} = \begin{pmatrix} \mathbf{0} & \tilde{\mathbf{P}}_\ell & \mathbf{0} \\ \tilde{\mathbf{P}}_\ell & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{1.57}$$

With the help of these matrices, we can write

$$\mathbf{S} = \sum_\ell \left[C_\ell^{TT} \mathbf{Y} \tilde{\mathbf{P}}_\ell^{TT} \mathbf{Y}^\dagger + C_\ell^{TE} \mathbf{Y} \tilde{\mathbf{P}}_\ell^{TE} \mathbf{Y}^\dagger + C_\ell^{EE} \mathbf{Y} \tilde{\mathbf{P}}_\ell^{EE} \mathbf{Y}^\dagger + C_\ell^{BB} \mathbf{Y} \tilde{\mathbf{P}}_\ell^{BB} \mathbf{Y}^\dagger \right]. \tag{1.58}$$

By observation of this expression, one can see in action the elements that determine the covariance matrix of the signal in pixel space: it is a sum of the contributions of the components of the power spectrum at different angular scales. The power of each component at each angular scale multiplies a kind of covariance matrix atoms in pixel space —we can see those matrices as a basis of the covariance in pixel space—. Therefore, the covariance is a weighted sum of covariance atoms.

The elements of each of the covariance matrix atoms are the result of a computation similar to that shown in eq. (1.44), but in which different columns of the matrix **Y** are combined. The *selection* matrix $\tilde{\mathbf{P}}_\ell^{XY}$ —the basis of the covariance matrix in harmonic space— determines the columns to mix. By this property, selection, the matrices $\tilde{\mathbf{P}}_\ell^{XY}$ play a crucial role in the efficient implementation of the power spectra estimator that we have developed (see Chapter 5).

Finally, with the help of the index i which includes the component of the power spectrum and the scale index ℓ, we get

$$\mathbf{S} = \sum_i C_i \mathbf{Y} \tilde{\mathbf{P}}_i \mathbf{Y}^\dagger. \tag{1.59}$$

In a real experiment, the observed data is a combination of signal and noise. Assuming that both components are independent, the data yielded by an experiment can be expressed as a sum of both components

$$\mathbf{d} = \mathbf{s} + \mathbf{n}, \tag{1.60}$$

where **d** is the data vector, **s** is the signal vector, and **n** is the noise vector .

Assuming $\langle \mathbf{s} \rangle = \langle \mathbf{n} \rangle = \mathbf{0}$, the covariance of the data is

$$\langle \mathbf{d}\mathbf{d}^t \rangle = \langle \mathbf{s}\mathbf{s}^t \rangle + 2\langle \mathbf{s}\mathbf{n}^t \rangle + \langle \mathbf{n}\mathbf{n}^t \rangle. \tag{1.61}$$

By statistical independence between signal and noise, the cross product vanishes, therefore, being **C** the covariance matrix of the data and **N** the covariance of the noise we have

$$\mathbf{C} = \mathbf{S} + \mathbf{N}. \tag{1.62}$$

Finally, by eq. (1.55), the covariance matrix of the data is

$$\mathbf{C} = \sum_i C_i \mathbf{P}_i + \mathbf{N}. \tag{1.63}$$

1.8.2 The need for an estimator

Equations. (1.4) and (1.27) allow us to straightforwardly calculate the values of the CMB maps in pixel space from the coefficients of the harmonic expansion. In this section, we will analyze the possibility of transforming the information in the data in the opposite direction, from pixel to harmonic space. We will see that in realistic situations it is not possible to do this directly, which means that we will have to give up the idea of a straightforward computation of the power spectrum as being defined in eq. (1.9), so we will have to resort to parameter estimation techniques.

The problem

Without loss of generality, let us refer to the only temperature case in this section. In an ideal situation, making use of the orthogonality property of the spherical harmonics, from eq. (1.4) it is easy to compute the coefficients $a_{\ell m}$ from a temperature map

$$a_{\ell m} \equiv \int d\Omega(\hat{r}) Y^*_{\ell m}(\hat{r}) \delta T(\hat{r}), \tag{1.64}$$

where the integral is done over the whole sphere. Having computed the $a_{\ell m}$'s, the power spectrum can be obtained from eq. (1.9).

In a real situation, there are two facts that make the above expressions impractical. Because of the high contamination of the CMB observations by emission from other types of microwave sources and because some experiments can only capture data from a fraction of the sky, the CMB data maps do not cover the whole sphere. Therefore the integral of eq. (1.64) can only be computed over a fraction of the sky. That means that the essential orthogonality property of the spherical harmonics is lost. Therefore, the $a_{\ell m}$'s computed from eq. (1.64) with partial sky coverage, $\tilde{a}_{\ell m}$, become a linear combination of the *true* coefficients of the map

$$\tilde{a}_{\ell m} = \sum_{\ell' m'} K_{\ell m}^{\ell' m'} a_{\ell' m'}. \tag{1.65}$$

The matrix $K_{\ell m}^{\ell' m'}$ gives account of the coupling of the harmonics over the observed fraction of the sky.

There is another issue that makes expression (1.64) unfeasible. As defined mathematically, the integral is a continuous sum, in this case over the spherical surface; but real CMB maps are made of sampling of data at $N_{\rm pix}$ locations. This means that, even if the data covered all the sky, the integral of eq. (1.64) cannot be computed as it is defined. The integral has to be transformed into a summation, and some approximations must be taken to solve the problem caused by the leap from the continuum to a discretized sum, as for example the ones that drive to the `HEALPix` [39] pixel window function. Because of the previous reasons, apparently, there is no way to compute the exact $a_{\ell m}$'s that describe an observed pixelized CMB map.

A possible solution

Arranging the values δT_i of each pixel of the observed temperature map in the vector **m**, the values of each spherical harmonic on each of the observed pixels as a column of the matrix **Y** and the coefficients $a_{\ell m}$ of the expansion in the vector **a**, eq. (1.4) can be expressed in matrix form

$$\mathbf{m} = \mathbf{Y}\mathbf{a}. \tag{1.66}$$

Multiplying by the left by the matrix product $(\mathbf{Y}^\dagger\mathbf{Y})^{-1}\mathbf{Y}^\dagger$ on both sides of the equation, we get

$$(\mathbf{Y}^\dagger\mathbf{Y})^{-1}\mathbf{Y}^\dagger\mathbf{m} = (\mathbf{Y}^\dagger\mathbf{Y})^{-1}\mathbf{Y}^\dagger\mathbf{Y}\mathbf{a}. \tag{1.67}$$

Therefore

$$\mathbf{a} = (\mathbf{Y}^\dagger\mathbf{Y})^{-1}\mathbf{Y}^\dagger\mathbf{m}. \tag{1.68}$$

According to the last expression, in principle, one can compute exactly the coefficients $a_{\ell m}$ of a given map using eq. (1.68) regardless of whether the map is masked or full-sky or whether the data is pixelized. The mechanism of this expression is simple. The product $\mathbf{Y}^\dagger\mathbf{m}$ is the matrix form of the inverse harmonic transform. Since the coefficients given by $\mathbf{Y}^\dagger\mathbf{m}$ are the true $a_{\ell m}$'s coupled because of the mask (i.e., the $\tilde{a}_{\ell m}$'s coefficient), one needs to disentangle them. The elements of the matrix product $\mathbf{Y}^\dagger\mathbf{Y}$ give account of the coupling of the spherical harmonics —they are the scalar product computed over the observed fraction of the sky—. Multiplying the coupled $a_{\ell m}$'s by the inverse of the coupling matrix gives place to the true $a_{\ell m}$'s. But this expression leads to two problems. The first is technical. One has to compute and invert the matrix product $\mathbf{Y}^\dagger\mathbf{Y}$, which can be very computationally demanding due to the size of the matrix **Y**. The second one appears when we look at the situation from the perspective of a system of linear equations: because the number of spherical harmonics needed to correctly describe a realistic signal is greater than the number of pixels in the map, to solve **a** from eq. (1.66) will be equivalent to solving a system of equations with more variables than equations. This implies that the matrix $\mathbf{Y}^{\dagger}\mathbf{Y}$ is not regular.

Since we cannot invert the product matrix, the mathematical problems we have encountered tell us that we cannot unravel the power leakage between the coefficients $a_{\ell m}$. We will see below that there are methods to compute the power spectrum without first computing the coefficients of the spherical harmonic expansion and that in some of them the leakage is completely disentangled.

On the other hand, the product matrix could be inverted if the observed signal were described by a small number of spherical harmonics —for example, a signal affected by a large beam—. In this case, if the number of variables is smaller than the number of equations, at least there would be mathematical options to solve the system.

The conclusion

This means that, since we cannot directly compute the power spectrum from the data in the map —in the sense of solving a set of equations— we have to resort to estimation methods.

1.9 Parameter estimation

Given a theory, it is assumed that one has a probabilistic model that describes the data yielded by an experiment. Usually, the model includes some parameters —in our case of physical significance— and it is supposed that we need a method to obtain the values of the parameters from the observed data.

For simplicity, let us consider a single parameter θ and let us assume we have a model that defines the probability distribution $f(\mathbf{x};\theta)$, that depends on θ, of the data samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ given by the experiment. Thus the probability of that an experiment yields a particular value to the sample $\mathbf{x}_i$ depends on the true value of θ. A single sample of data $\mathbf{x}_i$ can be just one single value or a collection of numbers —a vector—, as is the case of a CMB temperature map, in which a single sample is made up of the collection of values δT_j in the observed pixels.[7]

---

[7] In this section, we are trying to describe the size of the data sample with some generality. However, when applying these techniques to a realistic CMB experiment the sample size would be one —our universe is the only realization of the underlying probability distribution, and the anisotropies we see on the

Conceptually, an estimator $\hat{\Theta}$ is a mathematical operation on random variables $\mathbf{X}_i$ whose output is a numerical value of the parameter θ

$$\hat{\Theta} = h(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n). \tag{1.69}$$

In practice, an experiment yields the data samples $\mathbf{x}_i$ of the random variables $\mathbf{X}_i$ and the estimator $\hat{\Theta}$, operating on the values $\mathbf{x}_i$ yields a particular value $\hat{\theta}$ of the parameter θ

$$\hat{\theta} = h(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n). \tag{1.70}$$

Note that according to this definition, classical estimators in cosmology (maximum likelihood estimator, quadratic maximum likelihood, Pseudo-C_ℓ methods, etc.) fall into this category. In all of them, after a certain pre-processing of the data, calculations are made on the data given by an experiment that can be described as a mathematical function h.

If we perform the experiment again, it is expected that another data sample would be measured and, therefore, the estimator would likely catch another value. Since the data sample is made up of random numbers, the estimated parameter is also a random number. In this context, let us make the simple assumption that the random variables $\mathbf{x}_i$ yielded by repeated experiments are statistically independent.

There are some criteria that can be used to define the quality of an estimator. A desirable property is that, if it were to operate on all possible data to estimate the value of a parameter of the probability distribution, the result would be the true numerical value of the parameter driving the probability distribution, whatever the true value of θ is. Consequently, an estimator $\hat{\Theta}$ is said to be unbiased if the expected value of $\hat{\theta}$ is θ when the estimator operates on the totality of possible values, taking into account the probability density of the values. Being $h(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ the calculation performed by the $\hat{\Theta}$ operator to determine the estimated value $\hat{\theta}$ of the parameter θ given the data samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$, under the assumption of statistical independence, for the estimator to be unbiased it is expected that

$$\theta = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} h(\mathbf{x}_1, \ldots, \mathbf{x}_n) f(\mathbf{x}_1; \theta) \ldots f(\mathbf{x}_n; \theta) d\mathbf{x}_1 \ldots d\mathbf{x}_n. \tag{1.71}$$

In practice, for example, when performing simulations, when we sum and take the means of the estimated parameters, we do not apply weights given by the probability distribution but assume the elementary principle that the most probable realizations

---

last scattering surface are the only sample we can measure—. This is, again, the limitation that leads to the concept of cosmic variance.

are the most frequent. From this point of view, if the number of realizations is sufficiently large for the probability law to be fulfilled, an estimator is unbiased when

$$\langle\hat{\Theta}\rangle = \theta. \tag{1.72}$$

Thus, this property is associated with the mean of the estimations on a representative set of realizations.

Another desirable property is that the values given by the estimator tend to the true value as the sample size grows, a large-sample concept. An estimator is said to be consistent when

$$\lim_{n\to\infty} \hat{\Theta}(\mathbf{x}_1,\ldots,\mathbf{x}_n) \to \theta. \tag{1.73}$$

Therefore, the consistency property is associated with the behavior of the succession of estimations as the sample size increases. By definition, a consistent estimator is unbiased when the estimation is calculated from a sufficiently large sample. In principle, no further claims can be made related to one property implying the other.

Since the estimated value is a random number, it is assumed to be affected by random fluctuations. It is to be expected that for a well-defined estimator the variance of the parameter depends on the intrinsic fluctuations of the data and on the sample size. There is a surprisingly general property applicable to all unbiased estimators of the form of eq. (1.70), the Cramér-Rao inequality, which sets a *minimum variance bound* to the variance of the estimated parameter. Assuming eq. (1.71) it can be shown that (see e.g. [40, 41])

$$\mathrm{Var}(\hat{\Theta}) \geq \left[n \left\langle \left(\frac{\partial \ln f(\mathbf{x};\theta)}{\partial \theta} \right)^2 \right\rangle \right]^{-1}. \tag{1.74}$$

Where the average is made as in eq. (1.71). Note that since the bound on the variance does not depend on the function h, it is valid for all unbiased estimators. Note also that it shows that the variance decreases with the sample size n.

In the case in which the estimator gives the value of multiple parameters, being $\boldsymbol{\theta}$ a vector of parameters θ_1, θ_2, ..., θ_m, the inverse of the covariance matrix of the parameters is related to the expectation value shown in the expression

$$\left[\mathrm{Cov}^{-1}(\hat{\Theta})\right]_{ij} \leq n \left\langle \frac{\partial \ln f(\mathbf{x};\boldsymbol{\theta})}{\partial \theta_i} \frac{\partial \ln f(\mathbf{x};\boldsymbol{\theta})}{\partial \theta_j} \right\rangle, \quad i,j = 1,2\ldots m. \tag{1.75}$$

The minimum bound to the covariance can be computed when the matrix whose elements are given by the expression at the right side of the inequality is regular. In this case, it would be limited by

$$\mathrm{Cov}(\hat{\Theta}) \geq \frac{1}{n}\Lambda^{-1}, \tag{1.76}$$

where Λ is a matrix whose values are the expectations of eq. (1.75).

It is common to find in the texts the previous expressions given in terms of the *likelihood* function. For a collection $\boldsymbol{\theta}$ of parameters and independent data samples $\mathbf{x}_1$, $\mathbf{x}_2, \ldots, \mathbf{x}_n$ the likelihood function L is the product of the probability densities

$$L \equiv L(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n;\boldsymbol{\theta}) = f(\mathbf{x}_1;\boldsymbol{\theta})f(\mathbf{x}_2;\boldsymbol{\theta})\ldots f(\mathbf{x}_n;\boldsymbol{\theta}). \tag{1.77}$$

The Cramér-Rao inequality for several parameters in terms of the likelihood is of the form

$$\left[\mathrm{Cov}^{-1}(\hat{\Theta})\right]_{ij} \leq \left\langle \frac{\partial \ln L}{\partial \theta_i}\frac{\partial \ln L}{\partial \theta_j}\right\rangle = -\left\langle \frac{\partial^2 \ln L}{\partial \theta_i \partial \theta_j}\right\rangle, \quad i,j = 1,2\ldots m \tag{1.78}$$

The expectations of the last expression are the elements of the well-know matrix in theory of information, the so-called *Fisher information matrix*

$$\mathbf{F}_{ij} = \left\langle \frac{\partial^2 \mathcal{L}}{\partial \theta_i \partial \theta_j}\right\rangle, \tag{1.79}$$

where $\mathcal{L} \equiv -\ln L$. If $\mathbf{F}$ is regular, for any unbiased estimator $\Delta\theta_i \geq \sqrt{(\mathbf{F}^{-1})_{ii}}$.

In statistics jargon, a BUE estimator, which stands for *Best Unbiased Estimator*, is an unbiased estimator and of minimal variance, that is, unbiased and efficient.

1.9.1 On the flux of the information

In this section, we want to analyze how information flows, particularly whether it is conserved, when we estimate a set of parameters directly from the data or from intermediate parameters estimated from the data.

We will measure the amount of information in terms of the Fisher matrix. In particular, we will analyze whether we can compute the Fisher information matrix expressed in terms of some parameters —for example, λ— once we have the matrix expressed in terms of other parameters —for example, θ.

The Fisher matrix defined in terms of the parameters λ is

$$\mathbf{F}^{\lambda}_{ij} \equiv -\left\langle \frac{\partial^2}{\partial \lambda_i \partial \lambda_j} \ln L\right\rangle. \tag{1.80}$$

Since we want to analyze whether the matrix can be computed in terms of the parameters λ from the matrix expressed in terms of the parameters θ, we will transform the derivatives of the likelihood with respect to λ into derivatives with respect to θ. By the chain rule

$$\frac{\partial \ln L}{\partial \lambda_j} = \sum_{\alpha} \frac{\partial \ln L}{\partial \theta_\alpha}\frac{\partial \theta_\alpha}{\partial \lambda_j}. \tag{1.81}$$

In the last expression there are two implicit conditions to be mentioned: we are assuming that the likelihood can be expressed completely in terms of the variables θ and we are assuming that we have complete knowledge of the dependence of the variables θ on the variables λ. When we apply the results of this section to the CMB we should check whether these conditions are met.

By applying the chain rule to the expression within the expected value in eq. (1.80), we get

$$\frac{\partial^2 \ln L}{\partial\lambda_i \partial\lambda_j} = \sum_{\alpha\beta} \left\{ \frac{\partial^2 \ln L}{\partial\theta_\alpha \partial\theta_\beta} \frac{\partial\theta_\beta}{\partial\lambda_i} \frac{\partial\theta_\alpha}{\partial\lambda_j} + \frac{\partial \ln L}{\partial\theta_\alpha} \frac{\partial^2\theta_\alpha}{\partial\lambda_i \partial\lambda_j} \right\}. \tag{1.82}$$

The Fisher matrix is the expected value of the second derivative, so let us analyze the effect of computing the average. Since the likelihood is a probability

$$\int L(\mathbf{x};\lambda) d\mathbf{x} = 1 \tag{1.83}$$

Taking the derivative with respect to λ of the last expression and taking into account that the derivative affects neither the space nor the integration variables —it affects only the dependence of the likelihood on the λ variable—, we get

$$\frac{\partial}{\partial\lambda} \int L(\mathbf{x};\lambda) d\mathbf{x} = \int \frac{\partial L(\mathbf{x};\lambda)}{\partial\lambda} d\mathbf{x} = \int \frac{\partial \ln L(\mathbf{x};\lambda)}{\partial\lambda} L(\mathbf{x};\lambda) d\mathbf{x} = \left\langle \frac{\partial \ln L(\mathbf{x};\lambda)}{\partial\lambda} \right\rangle = 0. \tag{1.84}$$

Computing the expected value of eq. (1.82) and taking into account that the relationship between λ and θ is independent of the data values $\mathbf{x}$, so that the derivatives of the variables θ can come out of the integrals, we get

$$\left\langle \frac{\partial^2 \ln L}{\partial\lambda_i \partial\lambda_j} \right\rangle = \sum_{\alpha\beta} \left\langle \frac{\partial^2 \ln L}{\partial\theta_\alpha \partial\theta_\beta} \right\rangle \frac{\partial\theta_\beta}{\partial\lambda_i} \frac{\partial\theta_\alpha}{\partial\lambda_j} + \left\langle \frac{\partial \ln L}{\partial\theta_\alpha} \right\rangle \frac{\partial^2\theta_\alpha}{\partial\lambda_i \partial\lambda_j}. \tag{1.85}$$

By eq. (1.84) the last term of eq. (1.85) vanishes. Therefore, we get

$$\mathbf{F}^{\lambda}_{ij} = \sum_{\alpha\beta} \mathbf{J}_{\beta i} \mathbf{F}^{\theta}_{\alpha\beta} \mathbf{J}_{\alpha j} \tag{1.86}$$

Expressed in matrix form, where the matrix $\mathbf{J}$ is the jacobian of the transformation, and taking into account that the Fisher matrix is symmetric, we get

$$\mathbf{F}^{\lambda} = \mathbf{J}^t \mathbf{F}^{\theta} \mathbf{J} \tag{1.87}$$

The interesting thing about this result is that $\mathbf{F}^{\lambda}$ can be computed from $\mathbf{F}^{\theta}$, which means that information is preserved when parameters are changed.

Applied to the CMB, if λ represents the cosmological parameters and θ, the power spectrum, the previous result means that if it is easier or faster to estimate the power spectrum instead of the cosmological parameters, we can efficiently obtain knowledge about them from the power spectrum data without losing information, because the Fisher matrix expressed in terms of the cosmological parameters can be calculated from the matrix expressed in terms of the power spectrum, as if it were calculated directly.

For this to be possible, two conditions must be fulfilled: we need to know the Jacobian of the transformation, i.e., the dependence of the power spectrum on the cosmological parameters. In addition, the probability must be fully described in terms of the power spectrum. Assuming that the fluctuations are Gaussian, the second condition is satisfied: the probability density of the data is a multivariate Gaussian distribution in which the covariance matrix is determined by the power spectrum —plus the noise model—, as shown in Section 1.8.1. Therefore, the theory describing the CMB fluctuations and their probability can be written in terms of these parameters.

On the other hand, if the Jacobian matrix is regular, eq. (1.87) allows us to compute $\mathbf{F}^{\theta}$ in terms of $\mathbf{F}^{\lambda}$. Applied to the CMB, one might wonder whether, in the case where we know the Fisher matrix expressed in terms of the cosmological parameters, one could calculate the corresponding matrix expressed in terms of the power spectrum. Due to the degeneracies in the estimation of these parameters from the power spectrum, one would expect the Jacobian matrix to become a singular matrix: therefore, it cannot be calculated.

1.9.2 Maximum likelihood estimator

Reading $f(\mathbf{x};\boldsymbol{\theta})$ for a fixed value of $\mathbf{x}$ as a function of $\boldsymbol{\theta}$ that drives the probability of the parameters given the data, we define the *likelihood* of the values of the parameters $\boldsymbol{\theta}$ as

$$L(\boldsymbol{\theta};\mathbf{x}) \equiv f(\mathbf{x};\boldsymbol{\theta}), \tag{1.88}$$

that can be computed by making use of the theory.

The *maximum likelihood estimator* attempts to find the parameter values that maximize the likelihood, that is, it is an estimator focused on finding the mode of the parameters distribution.

The likelihood in the case of having many independent data samples $\mathbf{x}_i$, $i = 1 \ldots n$ is[8]

$$L(\boldsymbol{\theta};\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n\}) \equiv f(\mathbf{x}_1;\boldsymbol{\theta})\cdot f(\mathbf{x}_2;\boldsymbol{\theta})\cdots f(\mathbf{x}_n;\boldsymbol{\theta}). \tag{1.89}$$

---

[8] The following expression is the same as eq. (1.77) but with slight changes in the notation.

Note that, although the latter expression is useful in general, when estimating the CMB parameters one has only a sample, so the likelihood reduces to that of eq. (1.88).[9]

By definition, being the vector parameters $\boldsymbol{\theta}$ made of m parameters θ_i, the maximum likelihood estimator reduces to solving a set of m simultaneous equations

$$\frac{\partial L(\hat{\boldsymbol{\theta}};\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n\})}{\partial\hat{\theta}_i}=0, \quad i=1,\ldots,m. \tag{1.90}$$

It is usually easier to solve the equivalent expression in terms of the logarithm of the likelihood

$$\frac{\partial \ln L(\hat{\boldsymbol{\theta}};\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n\})}{\partial\hat{\theta}_i}=0, \quad i=1,\ldots,m. \tag{1.91}$$

The expressions (1.90) and (1.91) encode a concrete example of the function h of Section 1.9, particularly the function h of the maximum likelihood estimator.

In some cases, the above systems of equations can be solved analytically. In other cases, it is necessary to resort to numerical methods to solve the equations. In addition, there are maximization techniques for finding the maximum likelihood in parameter space that are very useful in heavy computations, in a brute-force approach.

Regarding the properties (see e.g. [40, 41]) of the estimator, the MLE is *asymptotically efficient*. That is, if the sample size n tends to infinity, the estimator is unbiased and saturates the minimum variance bound inequality. Therefore, under this condition, the MLE is a BUE estimator. Moreover, the covariances can be easily calculated

$$\mathrm{Cov}^{-1}(\hat{\theta}_i,\hat{\theta}_j)=-\left.\frac{\partial^2\ln L}{\partial\theta_1\partial\theta_j}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}. \tag{1.92}$$

That is, the Fisher matrix can be computed without resorting to integrals by calculating the value of the second derivatives on the numerical values of the estimated parameters —which, being MLE unbiased when $n\to\infty$, are equal to the *true* values of the parameters.

Nevertheless, these are *large-sample* properties. Thus very little can be said when the sample size is small; MLE can be biased and not efficient in this case. Note that this is not a particular feature of the estimator but is due to the fact that a small sample may not be representative of the underlying distribution.

---

[9] It is important to note that if one wants to write a likelihood that incorporates data from different independent experiments, one cannot use the expression (1.89) because the intrinsic data sample is the same —the particular realization of our universe—; that is, the experiments may be independent, but the data are not.

CMB maximum likelihood

Turning to the estimation of the CMB power spectrum, assuming that the signal and noise are statistically independent and that both are Gaussian, two conditions are satisfied. First, the probability distribution depends exclusively on the covariance matrix of the data. Second, by eq. (1.63) the covariance matrix depends only on the angular power spectrum[10] (of course, plus the noise model). Therefore, the likelihood depends exclusively on the power spectrum.

Assuming that we have a CMB data map $\mathbf{x}$ drawn from a statistical distribution of mean zero and being $\mathbf{c}$ the vector power spectrum,[11] under the Gaussian assumption and without assuming any prior for the power spectrum, the likelihood is

$$L(\mathbf{c};\mathbf{x}) = \frac{1}{(2\pi)^{N_{\mathrm{pix}}/2}|\mathbf{C}|^{1/2}} \exp\left[-\frac{\mathbf{x}^t\mathbf{C}^{-1}\mathbf{x}}{2}\right], \tag{1.93}$$

where $\mathbf{C}$, the covariance matrix of the maps $\mathbf{C} = \langle \mathbf{x}\mathbf{x}^t \rangle$, introduces the dependence on the parameters $\mathbf{c}$ (see eq. (1.63)).

Referred to the error of the estimation, the elements of the Fisher matrix, eq. (1.79), of a likelihood given by a Gaussian probability distribution [42] are

$$\mathbf{F}_{ii'} = \frac{1}{2}\,\mathrm{tr}\left[\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial C_i}\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial C_{i'}}\right], \tag{1.94}$$

where the variables C_i are the elements of the vector $\mathbf{c}$. Of course, $\mathbf{C}$ and the partial derivatives have to be evaluated at the values of the estimated vector $\hat{\mathbf{c}}$ that maximizes the likelihood.

Combining equations (1.94), (1.62) and (1.55), we get

$$\mathbf{F}_{ii'} = \frac{1}{2}\,\mathrm{tr}\left[\mathbf{C}^{-1}\mathbf{P}_i\mathbf{C}^{-1}\mathbf{P}_{i'}\right]. \tag{1.95}$$

Determining the numerical values $\hat{\mathbf{c}}$ that maximize L is a gigantic task when the number of pixels is large. Some techniques have been developed to speed up the computation of the likelihood function. The Karhunen-Loève technique is based on a decomposition of the covariance matrix into a sum of modes followed by a selection of those that provide the most information in the sense of the signal-to-noise ratio [42].

---

[10] Note that the signal term of the covariance matrix depends on the power spectrum, whether the data are Gaussian or not.

[11] Although so far we have used $\boldsymbol{\theta}$ to refer to the vector of parameters, since in this section the parameters that fix the probability distribution are the multipoles C_i of the power spectrum, we will use $\mathbf{c}$ to refer to the vector of parameters. Therefore, in this section C_i is a parameter, $\mathbf{c}$ the vector of parameters and $\mathbf{C}$ the covariance matrix.

Although it allows to reduce the number of operations, it is not practical to apply this technique to maps with a large number of pixels. So the data that the current experiments are generating cannot be processed in this way. In a later chapter it will be shown that the maximun likelihood power spectrum can be found without resorting to brute-force maximization using a quadratic method on the data (QML) [43]. It should be noted, however, that although QML can be used to compute the spectrum of maximum likelihood by reducing the number of operations with respect to a brute-force maximization, the method is still very computationally intensive, so it cannot be applied to very large maps.

1.10 Pseudo-C_ℓ methods

Due to the lack of orthogonality of the spherical harmonics, when the maps are masked, the coefficients of the spherical harmonics expansion cannot be calculated directly using the inverse transform. However, an unbiased estimator can be defined by means of this operation. The estimator relates the observed power spectra to the true underlying one by means of a coupling matrix reflecting the effect of the cut-sky applied to the data. It is not of minimum variance but is computationally optimal. Let us focus initially on the estimation of the intensity power spectrum. Subsequently, we will show how the estimator operates in the case of intensity and polarization.

The field $\delta T(\hat{r})$ defined over the full-sky can be expanded as a sum of spherical harmonics weighted by the coefficients

$$a_{\ell m} = \int \delta T(\hat{r}) Y^*_{\ell m}(\hat{r}) d\Omega(\hat{r}). \tag{1.96}$$

In the case of CMB observations, (reliable) measurements in all directions of space are not available and, therefore, the above integral cannot be performed over the entire surface of the sphere. However, by introducing a function that assigns a weight to each direction, $W(\hat{r})$, we can formally retain the integral over the entire surface. In this condition, the coefficients we would compute are no longer the true coefficients of the signal map, but a kind of pseudo-coefficients

$$\tilde{a}_{\ell m} = \int W(\hat{r}) \delta T(\hat{r}) Y^*_{\ell m}(\hat{r}) d\Omega(\hat{r}). \tag{1.97}$$

Assuming that the *true* coefficients of the map are $a_{\ell m}$, we get

$$\tilde{a}_{\ell m} = \sum_{\ell' m'} a_{\ell' m'} \int W(\hat{r}) Y_{\ell' m'}(\hat{r}) Y^*_{\ell m}(\hat{r}) d\Omega(\hat{r}), \tag{1.98}$$

where the integral defines the coupling coefficients $K_{\ell m}^{\ell' m'}$ of eq. (1.65). The pseudo power spectrum can be defined as

$$\tilde{C}_\ell = \frac{1}{2\ell+1}\sum_m |\tilde{a}_{\ell m}|^2. \tag{1.99}$$

Our goal is to find the relationship between the pseudo power spectrum thus measured and the power spectrum of the signal. Since the function that determines the weighting is itself a field on the sphere surface, it can be expanded on spherical harmonics with coefficients

$$w_{\ell m} = \int W(\hat{r}) Y_{\ell m}^*(\hat{r}) d\Omega(\hat{r}), \tag{1.100}$$

where the last integral can be calculated over the entire surface. Therefore, the weighting function has a power spectrum

$$\mathcal{W}_\ell = \frac{1}{2\ell+1}\sum_m |w_{\ell m}|^2. \tag{1.101}$$

Introducing eq. (1.100) into eq. (1.98) we get

$$\tilde{a}_{\ell m} = \sum_{\ell' m' \ell'' m''} a_{\ell' m'} w_{\ell'' m''} \int Y_{\ell'' m''}(\hat{r}) Y_{\ell' m'}(\hat{r}) Y_{\ell m}^*(\hat{r}) d\Omega(\hat{r}). \tag{1.102}$$

Therefore, we get

$$\tilde{a}_{\ell m} = \sum_{\ell' m'} a_{\ell' m'} K_{\ell m \ell' m'}(W), \tag{1.103}$$

where $K_{\ell m \ell' m'}$ is the coupling kernel resulting from the sky weighting, W

$$K_{\ell m \ell' m'}(W) = \sum_{\ell'' m''} w_{\ell'' m''} \int Y_{\ell'' m''}(\hat{r}) Y_{\ell' m'}(\hat{r}) Y_{\ell m}^*(\hat{r}) d\Omega(\hat{r}). \tag{1.104}$$

Note that we have managed to take the dependence on the weighting function in the kernel out of the integral, so that what remains are integrals over the complete surface, whose values can be computed independently of the weighting scheme, and are related to the Wigner 3-j symbols (or Clebsch-Gordan coefficients).

It can be shown (see details in [44]) that the assemble averaged power spectrum is

$$\langle \tilde{C}_\ell \rangle = \frac{1}{2\ell+1}\sum_m \sum_{\ell'} \langle C_{\ell'} \rangle \sum_{m'} K_{\ell m \ell' m'}[W], \tag{1.105}$$

that can be expressed as

$$\langle \tilde{C}_\ell \rangle = \sum_{\ell'} \mathcal{M}_{\ell \ell'} \langle C_{\ell'} \rangle. \tag{1.106}$$

Making use of the orthogonality relation of the Wigner symbols and the definition of the power spectrum of the weighting function, the coupling kernel reads

$$\mathcal{M}_{\ell\ell'} = \frac{2\ell'+1}{4\pi}\sum_{\ell''}(2\ell''+1)\mathcal{W}_{\ell''}\begin{pmatrix}\ell & \ell' & \ell'' \\ 0 & 0 & 0\end{pmatrix}. \tag{1.107}$$

If we assume that the map given by $\delta T(\hat{r})$ contains signal and noise and also the statistical independence of both components, eq. (1.106) becomes

$$\langle\tilde{C}_\ell\rangle = \sum_{\ell'}\mathcal{M}_{\ell\ell'}\langle C_{\ell'}\rangle + \langle\tilde{N}_\ell\rangle, \tag{1.108}$$

where $\langle\tilde{N}_\ell\rangle$ is the average pseudo power spectrum of the noise that has to be determined by Monte Carlo simulations.

If the $\mathcal{M}$ kernel were non-singular, the next unbiased estimator could be defined

$$\widehat{C}_\ell = \sum_{\ell'}(\mathcal{M}^{-1})_{\ell\ell'}(\tilde{C}_{\ell'} - \langle\tilde{N}_{\ell'}\rangle). \tag{1.109}$$

Either to minimize the correlations or because the $\mathcal{M}$ kernel is singular, it is convenient to bin the power spectrum in ℓ. In this case, an unbiased binned power spectrum estimator can be defined (see details in [44]).

Regarding polarization, similar expressions to eq. (1.102) can be found for the coefficients $\tilde{a}^E_{\ell m}$ and $\tilde{a}^B_{\ell m}$, which mix $a^E_{\ell m}$ and $a^B_{\ell m}$. The ratios are given by the coefficients of the weighting function of the polarization map, $w^P_{\ell m}$, and integrals of products of the spin 0 and spin ±2 spherical harmonics (see details in [45, 46]). That is

$$\begin{aligned}\tilde{a}^E_{\ell m} &= \frac{1}{2}\sum_{\ell'm'\ell''m''} w^P_{\ell''m''}\left[a^E_{\ell'm'}\Phi_{\ell m\ell'm'\ell''m''} + a^B_{\ell'm'}\Psi_{\ell m\ell'm'\ell''m''}\right] \\ \tilde{a}^B_{\ell m} &= \frac{1}{2}\sum_{\ell'm'\ell''m''} w^P_{\ell''m''}\left[a^B_{\ell'm'}\Phi_{\ell m\ell'm'\ell''m''} - a^E_{\ell'm'}\Psi_{\ell m\ell'm'\ell''m''}\right]\end{aligned} \tag{1.110}$$

where

$$\Phi_{\ell m\ell'm'\ell''m''} = \int Y_{\ell''m''}(\hat{r})\left({}_2Y_{\ell'm'}(\hat{r})\,{}_2Y^*_{\ell m}(\hat{r}) + {}_{-2}Y_{\ell'm'}(\hat{r})\,{}_{-2}Y^*_{\ell m}(\hat{r})\right)d\Omega(\hat{r}) \tag{1.111}$$

and

$$\Psi_{\ell m\ell'm'\ell''m''} = i\int Y_{\ell''m''}(\hat{r})\left({}_2Y_{\ell'm'}(\hat{r})\,{}_2Y^*_{\ell m}(\hat{r}) - {}_{-2}Y_{\ell'm'}(\hat{r})\,{}_{-2}Y^*_{\ell m}(\hat{r})\right)d\Omega(\hat{r}). \tag{1.112}$$

As for the intensity case, eq. (1.106), the pseudo-C_ℓ spectra measured from a finite region of the sky are related to the full-sky power spectra by a kernel coupling

$$\langle\tilde{C}^{XY}_\ell\rangle = \sum_{X'Y'\ell'}\mathcal{M}^{XYX'Y'}_{\ell\ell'}\langle C^{X'Y'}_{\ell'}\rangle, \tag{1.113}$$

were the pairs XY are related to the six components of the power spectrum. The elements of the blocks of the kernel can be found in [46] and are again related to the Wigner 3-j symbols.

The covariance matrix of the estimated power spectrum can be estimated by Monte Carlo simulations [45] or computed analytically [46].

As defined, the estimator is unbiased, but not of minimum variance because nothing guarantees that it saturates the Cramér-Rao inequality. Moreover, the covariance and the estimation itself depend on the intensity and polarization weighting functions. The function can take values 0 and 1 to discriminate between observed and unobserved pixels, although the usual way to avoid aliasing is to introduce apodization, a smooth transition between observed and unobserved areas. When it is specifically adapted to analyze polarization, it ends up with a vanishing value of the first derivative at the boundary of the cut sky region.

The fact that the estimator is not of minimum variance is associated with power leakage between the different multipoles and components of the estimated power spectrum. Although it cancels on average, it contributes to increasing the variances. To minimize this, E/B mode purification techniques have been developed. They are based on the application on the measured Q and U fields of projection functions within orthogonal spaces with respect to the E and B contributions to the polarization field (see details in [47]).

1.11 On the motivation of this work

During the last decades, Cosmic Microwave Background observations have provided very valuable information to put together our current picture of the universe. In particular, among many other efforts, the *Planck* satellite has obtained the best full-sky CMB data in intensity and polarization over a large range of frequencies (30-857 GHz) up to date, allowing to impose constraints, in many cases at sub-percent level, over the cosmological parameters [10].

Given that CMB experiments usually produce a huge amount of data in the form of pixelized maps (T for intensity and the Q and U Stokes parameters for polarization), a crucial step in their analysis is the compression of this information in a more tractable way. In particular, since CMB fluctuations are expected to be nearly-Gaussian, most of their statistical information is contained in the 2-point correlation function (or equivalently in the power spectrum). Therefore, estimating the power spectrum is a key point in order to extract all the valuable cosmological information encoded in the CMB.

Different approaches have been developed for power spectra estimation, which differ in their efficiency and computational cost. In particular, maximum-likelihood based-methods (e.g. [48, 49]) provide optimal results in the sense that the estimator is unbiased and of minimum variance, but they are computationally very expensive and can not be implemented for high-resolution data. A particular case of this type of methods is the Quadratic Maximum Likelihood (QML), first introduced by [43] for intensity and extended to deal with polarization by [50] (see also [48, 51, 52, 53, 54]). Another popular approach are the pseudo-C_ℓ algorithms (see e.g. [55, 56, 44, 57, 58, 59] and references therein), which are much faster than maximum-likelihood methods and can therefore be used at the resolutions provided by current and planned experiments. They are also unbiased and their efficiency is comparable to that of the optimal methods at high multipoles, but not at large scales.

Although the utility of the pseudo-C_ℓ methods is out of discussion, the use of estimators which are optimal at large and intermediate scales is becoming increasingly important since they are critical for the detection of the primordial CMB polarization B-mode, whose main contribution is present at those scales. Note that having a QML method that can cover the full range of the reionization and the recombination peaks of the B-mode (even if a pseudo-spectrum method could be close to optimal in a part of this range) will provide not only a consistent optimal estimation of the relevant multipole range of the spectra but also of the corresponding full covariance matrix. Detection of primordial B-modes, which are parametrized by the tensor-to-scalar ratio r, would be a major breakthrough in Cosmology, since they are sourced by tensor perturbations, and its detection would constitute a definitive proof of the existence of a background of primordial gravitational waves, as predicted by inflationary models [60, 61, 62]. The best current constraint is given by $r_{0.05}<0.032$ at 95% CL obtained by combining *Planck*, BICEP/Keck2018 and BAO [38] (see also [63, 64, 65]), showing the faintness of the signal and the difficulty of its detection. Indeed, a large number of B-mode polarization experiments are currently ongoing or planned, such as for example the BICEP array [66], the Simons Observatory [67], the CMB-S4 experiment [68] or the JAXA LiteBIRD satellite [69], whose goal is to reach a sensitivity $\sigma_r(r=0) \leq 10^{-3}$.

This work presents `ECLIPSE` (Efficient Cmb poLarization and Intensity Power Spectra Estimator), an efficient implementation of the QML algorithm in `Fortran`, that allows us to compute the full power spectra with a very significant reduction of computational time, allowing us to work at higher resolution than before. To illustrate the performance of the method, we present results for a space-based B-mode mission and for a typical ground-based experiment. Although the QML method is well-known, there are several practical issues that should be taken into account when applying it to real data, such as the regularity of the covariance matrix of the data (of interest in many astrophysical problems in addition to the quadratic estimator), the regularity of the Fisher matrix, the choice of an initial guess for the power spectra or the performance of an iterative scheme. In particular, depending on the observed sky fraction, the Fisher matrix can become singular. To solve this problem, we construct a binned version of the QML estimator to be used when the Fisher matrix is not invertible.

CHAPTER 2

THE QUADRATIC MAXIMUM LIKELIHOOD ESTIMATOR

2.1 Motivation

The goal of this section is to show that an optimal estimator of the CMB power spectra and its covariance matrix, which operates in the space of pixels without appealing to the Fourier transform —thus, well suited to deal with incomplete sky coverage—, can be found. To do that, we will follow the steps shown in [43], offering both an in-depth description of the mathematical development and a didactic focus.[1]

2.2 The approach

Before starting, it is important to note that formally the deductions and the final formulations of the method for both the only intensity case and the intensity and polarization case are identical. This is because the covariance matrices of the maps for both cases are also formally identical. By eq. (1.63), in the intensity case the matrix is

$$\mathbf{C} = \sum_{\ell} C_{\ell} \mathbf{P}_{\ell} + \mathbf{N} \tag{2.1}$$

and the matrix for intensity and polarization is

$$\mathbf{C} = \sum_{\mu} C_{\mu} \mathbf{P}_{\mu} + \mathbf{N}, \tag{2.2}$$

where μ is an index that, if we compute the spectrum from $\ell = 2$ to $\ell = \ell_{\text{max}}$, takes, in principle, $6 \times (\ell_{\text{max}} - 1)$ values; that is, it runs through the six components of the power spectrum and all the values of ℓ of each component.

---

[1] An alternative derivation of the quadratic maximum likelihood estimator is given in [48]. It is based on truncating in the vicinity of the maximum the Taylor series of the maximum likelihood function up to order two and in approximating the curvature matrix by the Fisher matrix.

For simplicity, in the next sections we are going to deduce the method as if it were an only intensity case. The method will be derived in two steps: first, we will try to find an estimator that satisfies very restricted conditions. We will check that a BUE can not be found under those conditions. In a second step, taking advantage of what we have already learned in the previous step, we will show how to find a BUE if we are less restrictive about how the estimator operates.

2.3 First tentative

Our goal in this section is to find an unbiased quadratic estimator of minimum variance. We are going to try an estimator of the form

$$\widehat{C_\ell} = \mathbf{x}^t \mathbf{E}^\ell \mathbf{x} - b_\ell, \tag{2.3}$$

where $\mathbf{x}$ is the CMB map, $\mathbf{E}^\ell$ is a symmetric matrix to be determined and b_ℓ is a term that corrects the bias introduced by the noise.

Taking averages over the realizations, we get

$$\langle \widehat{C_\ell} \rangle = \langle \mathbf{x}^t \mathbf{E}^\ell \mathbf{x} \rangle - \langle b_\ell \rangle. \tag{2.4}$$

The noise bias is determined by our model of the noise of the experiment, and it does not depend of the realizations themselves, therefore $\langle b_\ell \rangle = b_\ell$.

Analyzing the quadratic function operating on the maps, we get

$$\mathbf{x}^t \mathbf{E}^\ell \mathbf{x} = \sum_{ij} x_i \mathbf{E}^\ell_{ij} x_j = \sum_{ij} \mathbf{E}^\ell_{ij} (x_i x_j) = \sum_{ij} \mathbf{E}^\ell_{ij} (x_j x_i) = \sum_i \Big[\sum_j \mathbf{E}^\ell_{ij} (x_j x_i) \Big], \tag{2.5}$$

and the sum on i in the last expression is the sum of the diagonal elements of the product –sum on j– of the $\mathbf{E}^\ell$ matrix by the matrix $(\mathbf{x}\mathbf{x}^t)$. Thus

$$\mathbf{x}^t \mathbf{E}^\ell \mathbf{x} = \operatorname{tr}\Big[\mathbf{E}^\ell (\mathbf{x}\mathbf{x}^t) \Big]. \tag{2.6}$$

Moving back to the averages

$$\langle \mathbf{x}^t \mathbf{E}^\ell \mathbf{x} \rangle = \operatorname{tr}\Big[\langle \mathbf{E}^\ell (\mathbf{x}\mathbf{x}^t) \rangle \Big] = \operatorname{tr}\Big[\mathbf{E}^\ell \langle \mathbf{x}\mathbf{x}^t \rangle \Big], \tag{2.7}$$

where we have taken into account that, being part of the estimator, the $\mathbf{E}^\ell$ matrices are independent of the realizations. Since the average of the cross-product of the pixels of the maps is the covariance matrix, we get

$$\langle \mathbf{x}^t \mathbf{E}^\ell \mathbf{x} \rangle = \operatorname{tr}\Big[\mathbf{E}^\ell \mathbf{C} \Big]. \tag{2.8}$$

At this point, it is important to highlight an essential aspect: in the last expression, $\langle \mathbf{x}\mathbf{x}^t \rangle$ has been replaced by $\mathbf{C}$, which means that $\mathbf{C}$ is the mathematical modelization of the information that one has about the statistical characteristics of the map to be analyzed. On the other hand, since $\mathbf{S}$, $\mathbf{N}$ and C_ℓ must be known to compute $\mathbf{C}$, it is required to dispose of an *a priori* model of the signal and the noise that characterizes our map.

Once the model is fixed, QML is going to provide an estimation of the power spectrum based on this model.[2] In the rest of this chapter we assume that we know the *true spectrum* —and that we have a correct model of the noise—. Once the deduction of the method is finalized, we will offer some considerations regarding what might arise when, previously to apply QML, we do not have a precise knowledge of the characteristics of the statistical distribution from which our map derives from (see more details about this question in Section 6.5.1).

Now we substitute eq. (2.8) in eq. (2.4), and it is found that the averages on the estimated spectrum are

$$\langle \widehat{C_\ell} \rangle = \mathrm{tr}\left[\mathbf{E}^\ell \mathbf{C}\right] - b_\ell. \tag{2.9}$$

Introducing the sum of signal and noise matrices in the eq. (2.9), yields

$$\langle \widehat{C_\ell} \rangle = \mathrm{tr}\left[\mathbf{E}^\ell (\mathbf{S}+\mathbf{N})\right] - \langle b_\ell \rangle = \mathrm{tr}\left[\mathbf{E}^\ell \mathbf{S}\right] + \mathrm{tr}\left[\mathbf{E}^\ell \mathbf{N}\right] - b_\ell, \tag{2.10}$$

and we have found the value that we can assign to the noise bias

$$b_\ell = \mathrm{tr}\left[\mathbf{E}^\ell \mathbf{N}\right]. \tag{2.11}$$

Therefore, by substituting eq. (2.11) in eq. (2.10), we find an expression in which all that rest is CMB signal

$$\langle \widehat{C_\ell} \rangle = \mathrm{tr}\left[\mathbf{E}^\ell \mathbf{S}\right]. \tag{2.12}$$

Writing the explicit form of $\mathbf{S}$, we get

$$\langle \widehat{C_\ell} \rangle = \mathrm{tr}\left[\mathbf{E}^\ell \mathbf{S}\right] = \mathrm{tr}\left[\mathbf{E}^\ell \sum_{\ell'} C_{\ell'} \mathbf{P}_{\ell'}\right], \tag{2.13}$$

thus

$$\langle \widehat{C_\ell} \rangle = \sum_{\ell'} C_{\ell'} \,\mathrm{tr}\left[\mathbf{E}^\ell \mathbf{P}_{\ell'}\right] = \sum_{\ell'} \mathbf{W}_{\ell\ell'} C_{\ell'}. \tag{2.14}$$

---

[2] In essence, this is the same as occurs in other situations, as with the *Maximum Likelihood Estimator*, in which we need to dispose of an *a priori* model that describes our map; or with the Wiener filter, in which we need to know the statistical characteristics of the observed maps previously, that is, the matrix $\mathbf{C}$.

Therefore the estimated power spectrum, in this way, is a linear combination of the elements of the *true spectrum*, being the *Window function*, $\mathbf{W}_{\ell\ell'}$

$$\mathbf{W}_{\ell\ell'} = \mathrm{tr}\left[\mathbf{E}^{\ell}\mathbf{P}_{\ell'}\right] \tag{2.15}$$

the mixing matrix. Therefore, for the estimator to be unbiased, it must be true that

$$\mathbf{W}_{\ell\ell'} = \delta_{\ell\ell'} \tag{2.16}$$

and, to be optimal, the covariance matrix of the estimated spectra

$$\mathbf{V}_{\ell\ell'} \equiv \langle \widehat{C}_{\ell}\widehat{C}_{\ell'}\rangle - \langle \widehat{C}_{\ell}\rangle\langle \widehat{C}_{\ell'}\rangle \tag{2.17}$$

must be equal to the inverse of the Fisher information matrix given by the Gaussian likelihood that describes the properties of the maps, which are encoded in $\mathbf{C}$.

Our goal now is to find the matrices $\mathbf{E}^{\ell}$ that minimize the variance $\mathbf{V}_{\ell\ell}$. To do that, we need to write the eq. (2.17) in terms of $\mathbf{E}^{\ell}$.

Before inserting the $\mathbf{E}^{\ell}$ matrices expressly, let us go back to the eq. (2.3), that is telling us that the value of the estimated multipole is the difference between the total power of the map minus the noise power. Using G_{ℓ} to describe the power in the map, $G_{\ell} = \mathbf{x}^{t}\mathbf{E}^{\ell}\mathbf{x}$, we will show that the covariance matrix of the estimated spectrum does not depend on the terms of the noise bias. Writing $\widehat{C}_{\ell} = G_{\ell} - b_{\ell}$, we get

$$\mathbf{V}_{\ell\ell'} = \langle (G_{\ell} - b_{\ell})(G_{\ell'} - b_{\ell'})\rangle - \langle G_{\ell} - b_{\ell}\rangle\langle G_{\ell'} - b_{\ell'}\rangle. \tag{2.18}$$

Expanding the products and taking into account that the noise bias term b_{ℓ} does not depend on a particular realization of the maps, we get

$$\begin{aligned}
\mathbf{V}_{\ell\ell'} &= \langle G_{\ell}G_{\ell'} - G_{\ell}b_{\ell'} - G_{\ell'}b_{\ell} + b_{\ell}b_{\ell'}\rangle - (\langle G_{\ell}\rangle - b_{\ell})(\langle G_{\ell'}\rangle - b_{\ell'}) \\
&= \langle G_{\ell}G_{\ell'}\rangle - \langle G_{\ell}\rangle b_{\ell'} - \langle G_{\ell'}\rangle b_{\ell} + b_{\ell}b_{\ell'} - \langle G_{\ell}\rangle\langle G_{\ell'}\rangle + \langle G_{\ell}\rangle b_{\ell'} + \langle G_{\ell'}\rangle b_{\ell} - b_{\ell}b_{\ell'} \\
&= \langle G_{\ell}G_{\ell'}\rangle - \langle G_{\ell}\rangle\langle G_{\ell'}\rangle.
\end{aligned} \tag{2.19}$$

Therefore, we can compute the covariances leaving aside the noise term. Let us now insert the $\mathbf{E}^{\ell}$ matrices in the expression that gives the covariances. From eq. (2.5) we get

$$\mathbf{V}_{\ell\ell'} = \left\langle\left(\sum_{ij}\mathbf{E}^{\ell}_{ij}(x_i x_j)\right)\left(\sum_{uv}\mathbf{E}^{\ell'}_{uv}(x_u x_v)\right)\right\rangle - \left\langle\sum_{ij}\mathbf{E}^{\ell}_{ij}(x_i x_j)\right\rangle\left\langle\sum_{uv}\mathbf{E}^{\ell'}_{uv}(x_u x_v)\right\rangle. \tag{2.20}$$

From the distributive property, the fact that averages and sums commute and taking into account that the $\mathbf{E}^{\ell}$ matrices are independent of the realizations, we get

$$\mathbf{V}_{\ell\ell'} = \sum_{ijuv}\mathbf{E}^{\ell}_{ij}\mathbf{E}^{\ell'}_{uv}\langle x_i x_j x_u x_v\rangle - \left(\sum_{ij}\mathbf{E}^{\ell}_{ij}\langle x_i x_j\rangle\right)\left(\sum_{uv}\mathbf{E}^{\ell'}_{uv}\langle x_u x_v\rangle\right). \tag{2.21}$$

The averages of the kind $\langle x_i x_j \rangle$ are the elements of the covariance matrix, independently of which is the statistical distribution of the CMB. The four-point covariances are more complicated. Nevertheless, under the assumption of Gaussian fluctuations with vanishing mean, they become a function of the two-point covariances

$$\langle x_i x_j x_u x_v \rangle = \langle x_i x_j \rangle \langle x_u x_v \rangle + \langle x_i x_u \rangle \langle x_j x_v \rangle + \langle x_i x_v \rangle \langle x_j x_u \rangle. \tag{2.22}$$

Inserting eq. (2.22) in eq. (2.21) and substituting the averages at two points by the covariance matrix, yields

$$\mathbf{V}_{\ell\ell'} = \sum_{ijuv} \mathbf{E}^{\ell}_{ij} \mathbf{E}^{\ell'}_{uv} \left[\mathbf{C}_{ij} \mathbf{C}_{uv} + \mathbf{C}_{iu} \mathbf{C}_{jv} + \mathbf{C}_{iv} \mathbf{C}_{ju} \right] - \left(\sum_{ij} \mathbf{E}^{\ell}_{ij} \mathbf{C}_{ij} \right) \left(\sum_{uv} \mathbf{E}^{\ell'}_{uv} \mathbf{C}_{uv} \right). \tag{2.23}$$

By the distributive property applied to the last product, we get

$$\mathbf{V}_{\ell\ell'} = \sum_{ijuv} \mathbf{E}^{\ell}_{ij} \mathbf{E}^{\ell'}_{uv} \left[\mathbf{C}_{ij} \mathbf{C}_{uv} + \mathbf{C}_{iu} \mathbf{C}_{jv} + \mathbf{C}_{iv} \mathbf{C}_{ju} \right] - \sum_{ijuv} \mathbf{E}^{\ell}_{ij} \mathbf{E}^{\ell'}_{uv} \mathbf{C}_{ij} \mathbf{C}_{uv}. \tag{2.24}$$

Simplifying, we find the final expression of the covariances

$$\mathbf{V}_{\ell\ell'} = \sum_{ijuv} \mathbf{E}^{\ell}_{ij} \mathbf{E}^{\ell'}_{uv} \left[\mathbf{C}_{iu} \mathbf{C}_{jv} + \mathbf{C}_{iv} \mathbf{C}_{ju} \right]. \tag{2.25}$$

The last expression can be simplified taking into account that the matrices $\mathbf{E}^{\ell}$ are symmetric. Breaking it up into two summations

$$\mathbf{V}_{\ell\ell'} = \sum_{ijuv} \mathbf{E}^{\ell}_{ij} \mathbf{E}^{\ell'}_{uv} \mathbf{C}_{iu} \mathbf{C}_{jv} + \sum_{ijuv} \mathbf{E}^{\ell}_{ij} \mathbf{E}^{\ell'}_{uv} \mathbf{C}_{iv} \mathbf{C}_{ju}. \tag{2.26}$$

Interchanging the dummy indices u and v in the second summation and making use of the symmetric property of the matrices $\mathbf{E}^{\ell}$, we get

$$\mathbf{V}_{\ell\ell'} = 2 \sum_{ijuv} \mathbf{E}^{\ell}_{ij} \mathbf{E}^{\ell'}_{uv} \mathbf{C}_{iu} \mathbf{C}_{jv}. \tag{2.27}$$

Reordering the matrices and making use again of the symmetric property, we find

$$\mathbf{V}_{\ell\ell'} = 2 \sum_{ijuv} \mathbf{C}_{ui} \mathbf{E}^{\ell}_{ij} \mathbf{C}_{jv} \mathbf{E}^{\ell'}_{vu}. \tag{2.28}$$

And we get that the covariances are related to the sum of the diagonal elements of the product matrix $\mathbf{E}^{\ell}\mathbf{C}\mathbf{E}^{\ell'}\mathbf{C}$. That is

$$\mathbf{V}_{\ell\ell'} = 2\,\mathrm{tr}\left[\mathbf{C}\mathbf{E}^{\ell}\mathbf{C}\mathbf{E}^{\ell'} \right]. \tag{2.29}$$

We can, now, find the matrix $\mathbf{E}^\ell$ that minimizes the variance. To do that, in the case $\ell' = \ell$, the derivative of $\mathbf{V}_{\ell\ell}$ with respect to E^ℓ must vanish. But because of the form in which the variance depends on $\mathbf{E}^\ell$ in the eq. (2.29), the unique solution to set the derivative to zero, i.e.,

$$2 \times 2 \times \mathbf{C}\mathbf{E}^\ell\mathbf{C} = 0 \tag{2.30}$$

is $\mathbf{E}^\ell = 0$, which it is the minimum variance trivial solution —the power spectrum and their variances are zero.

To find an interesting solution, let us introduce a Langrange multiplier which adds an extra addend to the derivative. According to eqs. (2.14), (2.15) and (2.16), we recall that the estimator must satisfy the following condition to be unbiased

$$\mathbf{W}_{\ell\ell} = \mathrm{tr}[\mathbf{E}^\ell\mathbf{P}_\ell] = 1. \tag{2.31}$$

Therefore, being λ a parameter to be determined, in this case we can write

$$\mathbf{V}_{\ell\ell} = 2\,\mathrm{tr}\left[\mathbf{C}\mathbf{E}^\ell\mathbf{C}\mathbf{E}^\ell\right] - 4\lambda\left(\mathrm{tr}\left[\mathbf{E}^\ell\mathbf{P}_\ell\ \right] - 1\right). \tag{2.32}$$

By taking the derivative and set it to zero, we get

$$4\mathbf{C}\mathbf{E}^\ell\mathbf{C} - 4\lambda\mathbf{P}_\ell\ = 0. \tag{2.33}$$

In this way, the term $4\lambda\mathbf{P}_\ell$ introduced by the Lagrange multiplier makes the solution different from the trivial one. The equation leaves to the —as expected— symmetric matrix

$$\mathbf{E}^\ell = \lambda\mathbf{C}^{-1}\mathbf{P}_\ell\mathbf{C}^{-1}. \tag{2.34}$$

And it only remains to find the value of λ that satisfies the eq. (2.31), that is, substituting $\mathbf{E}^\ell$

$$\lambda\,\mathrm{tr}[\mathbf{C}^{-1}\mathbf{P}_\ell\mathbf{C}^{-1}\mathbf{P}_\ell] = 1. \tag{2.35}$$

Therefore, we have the matrix that makes the estimator of minimum variance

$$\mathbf{E}^\ell = \frac{1}{\mathrm{tr}[\mathbf{C}^{-1}\mathbf{P}_\ell\mathbf{C}^{-1}\mathbf{P}_\ell]}\mathbf{C}^{-1}\mathbf{P}_\ell\mathbf{C}^{-1}. \tag{2.36}$$

Having arrived at this point, we have two pending questions. First, we can not be sure whether the estimator is unbiased. Let us recall that for this condition to be fulfilled, eq. (2.16) must be accomplished. As it has been defined, it is true when $\ell' = \ell$. But when they are different, we get

$$\mathbf{W}_{\ell\ell'} = \frac{1}{\mathrm{tr}[\mathbf{C}^{-1}\mathbf{P}_\ell\mathbf{C}^{-1}\mathbf{P}_\ell]}\,\mathrm{tr}[\mathbf{C}^{-1}\mathbf{P}_\ell\mathbf{C}^{-1}\mathbf{P}_{\ell'}]. \tag{2.37}$$

Equation (1.95) indicates us that the trace in the last expression is related to the elements of the Fisher information matrix

$$\mathbf{F}_{\ell\ell'} = \frac{1}{2}\operatorname{tr}\left[\mathbf{C}^{-1}\mathbf{P}_{\ell}\mathbf{C}^{-1}\mathbf{P}_{\ell'}\right]. \tag{2.38}$$

Therefore

$$\mathbf{W}_{\ell\ell'} = \frac{\mathbf{F}_{\ell\ell'}}{\mathbf{F}_{\ell\ell}}. \tag{2.39}$$

And there is no guarantee that the elements outside the diagonal of the Fisher matrix are equal to zero. Note that this would actually mean that the covariances of a hypothetical BUE defined in this form would be equal to zero, a desirable but unlikely property.

Second, looking back to the eq. (2.29) and computing the covariances of the estimator, we get

$$\mathbf{V}_{\ell\ell'} = \frac{\mathbf{F}_{\ell\ell'}}{\mathbf{F}_{\ell\ell}\mathbf{F}_{\ell'\ell'}}, \tag{2.40}$$

thus the covariances do not match the inverse of the Fisher matrix, but they are related to the elements of this matrix.

Therefore, we have found that an estimator as defined in eq. (2.3) does not match a BUE. However, making use of the knowledge acquired, the next section will be devoted to find an estimator that accomplishes all the required properties.

2.4 Second tentative: quadratic BUE

Instead of estimating the power spectrum in just one step, as in expression (2.3)

$$\widehat{C}_{\ell} = \mathbf{x}^t\mathbf{E}^{\ell}\mathbf{x} - b_{\ell}, \tag{2.41}$$

where we tried to compute each of the $\widehat{C}_{\ell}$ directly, we will construct an estimator defined *in two steps*.

Keeping the formal structure of the last expression, we will assume that our operator on the maps does not supply the values of the power spectrum directly, but some intermediary magnitudes in the harmonic space, that we will call y_{ℓ}. Next, we will analyze the relation of those magnitudes with the power spectrum. That is, we are going to define

$$y_{\ell} \equiv \mathbf{x}^t\mathbf{E}^{\ell}\mathbf{x} - b_{\ell}. \tag{2.42}$$

In the previous section we have found the formal structure of the $\mathbf{E}_{\ell}$ matrices, that translate the power of the anisotropies in the pixel space to the harmonic space

$$\mathbf{E}^{\ell} \propto \mathbf{C}^{-1}\mathbf{P}_{\ell}\mathbf{C}^{-1}. \tag{2.43}$$

Let us consider in this section $\mathbf{E}^\ell$ matrices with that structure but introducing a 1/2 factor —we will show that this drives to a BUE—. That is, let us define

$$\mathbf{E}^\ell \equiv \frac{1}{2}\mathbf{C}^{-1}\mathbf{P}_\ell\mathbf{C}^{-1}. \tag{2.44}$$

Having made this choice, eq. (2.42) becomes

$$y_\ell \equiv \frac{1}{2}\mathbf{x}^t\mathbf{C}^{-1}\mathbf{P}_\ell\mathbf{C}^{-1}\mathbf{x} - b_\ell. \tag{2.45}$$

Independently of the form of the $\mathbf{E}^\ell$ matrices, we know how to eliminate the noise bias, eq. (2.11). Taking averages and following the steps of the previous section, we find an expression similar to (2.14), but in which instead of $\widehat{C}_\ell$ we find y_l, that is

$$\langle y_\ell \rangle = \sum_{\ell'} C_{\ell'}\,\mathrm{tr}\left[\mathbf{E}^\ell\mathbf{P}_{\ell'}\right]. \tag{2.46}$$

Substituting our $\mathbf{E}_\ell$ matrix, we get

$$\langle y_\ell \rangle = \sum_{\ell'} C_{\ell'}\frac{1}{2}\,\mathrm{tr}\left[\mathbf{C}^{-1}\mathbf{P}_\ell\mathbf{C}^{-1}\mathbf{P}_{\ell'}\right], \tag{2.47}$$

where we can recognize the elements of the Fisher information matrix. Therefore

$$\langle y_\ell \rangle = \sum_{\ell'} \mathbf{F}_{\ell\ell'}C_{\ell'}. \tag{2.48}$$

Arranging the elements y_ℓ in the vector $\mathbf{y}$ and the C_ℓ values of the *true spectrum* in the vector $\mathbf{c}$, we get

$$\langle \mathbf{y} \rangle = \mathbf{F}\mathbf{c}. \tag{2.49}$$

And we can define the unbiased estimator

$$\widehat{\mathbf{c}} = \mathbf{F}^{-1}\mathbf{y}. \tag{2.50}$$

Certainly

$$\langle \hat{\mathbf{c}} \rangle = \langle \mathbf{F}^{-1}\mathbf{y} \rangle = \mathbf{F}^{-1}\langle \mathbf{y} \rangle = \mathbf{F}^{-1}\langle \mathbf{F}\mathbf{c} \rangle = \mathbf{F}^{-1}\mathbf{F}\mathbf{c} = \mathbf{c}. \tag{2.51}$$

Let us now analyze the covariances of the vector $\mathbf{y}$ and, later, the covariances given by our estimator. Going back to the eq. (2.17) and substituting in it C_ℓ by y_ℓ, using the eq. (2.29) and substituting the matrices $\mathbf{E}_\ell$, we get

$$\langle y_\ell y_{\ell'} \rangle - \langle y_\ell \rangle\langle y_{\ell'} \rangle = 2\,\mathrm{tr}[\mathbf{C}\mathbf{E}^\ell\mathbf{C}\mathbf{E}^{\ell'}] = \frac{1}{2}\,\mathrm{tr}[\mathbf{P}_\ell\mathbf{C}^{-1}\mathbf{P}_{\ell'}\mathbf{C}^{-1}]. \tag{2.52}$$

By the invariance of the trace of the product of matrices under cyclic permutations, the last term is the element $\ell\ell'$ of the Fisher matrix. Written in matrix form, the last expression is

$$\langle \mathbf{y}\mathbf{y}^t \rangle - \langle \mathbf{y} \rangle \langle \mathbf{y}^t \rangle = \mathbf{F}. \tag{2.53}$$

Making use of the last expression, we can compute the covariance of our estimator. By the definition of covariance, the definition of our estimator and the symmetric property of the Fisher matrix, we get

$$\langle \hat{\mathbf{c}}\hat{\mathbf{c}}^t \rangle - \langle \hat{\mathbf{c}} \rangle \langle \hat{\mathbf{c}}^t \rangle = \mathbf{F}^{-1} \left[\langle \mathbf{y}\mathbf{y}^t \rangle - \langle \mathbf{y} \rangle \langle \mathbf{y}^t \rangle \right] \mathbf{F}^{-1} = \mathbf{F}^{-1}\mathbf{F}\mathbf{F}^{-1} = \mathbf{F}^{-1}. \tag{2.54}$$

And we have found that the covariances of our estimator match the inverse of the Fisher matrix. That is, the estimator defined by the eq. (2.50) plus the eq. (2.42), (2.11), (2.44) and (2.38) is a BUE.

It is important to note that this shows that there are no estimators of a superior order to the quadratic that could yield better estimations than those given by the QML estimator because the latter produces estimations that reach the limit given by the Fisher-Chramér-Rao inequality. On the other hand, being a BUE, QML matches the maximum likelihood estimator.

2.5 Description in terms of alternative variables

When estimating the power spectra, one usually needs to explicitly consider the effects of the instrumental beam and of the pixel window function. In addition, it is also quite common to describe the angular power spectra per logarithmic interval as $D_\ell = \ell(\ell+1)C_\ell/2\pi$. Therefore, it may be convenient to implement the QML method in terms of the D_ℓ variables and/or including the instrumental resolution effects. This can be easily done by introducing some additional factors in the $\mathbf{P}_i$ matrices of eqs. (2.1) and (2.2).

Let us first denote by B_ℓ the beam and pixel instrumental effects, such that the harmonic coefficients of eq. (1.32) of the observed signal are given by $a_{\ell m}^{\text{Observed}} = B_\ell a_{\ell m}^{\text{Signal}}$. Analogously, let us define $W_i = B_\ell^X B_\ell^Y$, which encodes these effects in the power spectra. In this way, we can write the covariance matrix of the observed (smoothed) signal as (see eq. (1.55))

$$\mathbf{S} = \sum_i C_i W_i \mathbf{P}_i. \tag{2.55}$$

We can also write the previous equation in terms of the D_i variables

$$\mathbf{S} = \sum_i D_i \frac{2\pi}{\ell(\ell+1)} W_i \mathbf{P}_i = \sum_i D_i \check{\mathbf{P}}_i, \tag{2.56}$$

where we have defined the new matrices $\check{\mathbf{P}}_i$. It becomes apparent that by replacing C_i and $\mathbf{P}_i$ by D_i and $\check{\mathbf{P}}_i$, respectively, in the equations of the previous section, we have an implementation of the method such that the D_i quantities are estimated. Let us remark that these estimated spectra are corrected from the beam and pixel effects.

Of course, one could also easily write the equivalent expressions to estimate the power spectra in terms of the variables C_i or D_i and/or including the experimental beam. Along this Thesis we will use the QML in terms of different variables, as convenient, but note that this does not imply any loss of generality since all results can be straightforwardly obtained for any of the previously considered variables.

2.6 Discussion

Once we have found the method, it is worth making some remarks regarding the assumptions that we have made to develop it as well as discussing some issues that may arise when applying it.

2.6.1 Assumptions

We recall that the following assumptions were made to develop the method:

- **Gaussianity**: we have assumed that the fluctuations are Gaussian, which is expected in standard cosmology. Therefore, the method is applicable to CMB maps and other Gaussian maps. But, in principle and strictly speaking, it is applicable only to Gaussian fluctuations.

- **Absence of privileged directions**: we have assumed that the covariance matrix is characterized by the power spectrum, which means that there are not privileged directions in the space. In principle, difficulties can arise if we want to apply the method to maps of anisotropic signals —such as Galactic emissions—, because neither their statistical characteristics nor their covariance matrices will be given by eq. (2.1) or eq. (2.2).

- **Noise properties**: we have assumed that signal and noise are uncorrelated and that the noise is Gaussian.

In practice, QML can be applied to signals that do not completely fulfill those requirements, but one should take into account that the method is not working under the ideal conditions, especially if the data place us far from the previous assumptions. Therefore, in that case, the results have to be carefully interpreted.

2.6.2 Issues

It is also worth mentioning some issues that one needs to face when applying the methodology. First of all, we need to compute the inverse of the covariance matrix. However, there is no guarantee that this matrix will be regular under all circumstances; in Chapter 3 we analyze the conditions that must be accomplished for the matrix to be regular.

From the definition of the estimator, eq. (2.49), it becomes apparent that the inverse of the Fisher matrix needs to be computed. Again, there is no guarantee whether this matrix will be regular under all the considered circumstances. In Chapter 4 we analyze the issues associated with the lack of regularity of **F**. We will also show that the method can be used to estimate a binned power spectrum and that, choosing carefully the way to estimate the power in the bins, we can define a BUE for the binned power spectrum.

The eqs. (2.50), (2.42), (2.11), (2.44) and (2.38) contain the essence of the computations needed to apply the method. In Chapter 5 we will show that they can be transformed into equivalent expressions such that the number of required mathematical operations is significantly reduced. In Section 5.5 we describe ECLIPSE, a parallelized implementation of the efficient version of QML written in `Fortran` language designed to take advantage from the capabilities of a supercomputer to tackle large scale problems.

2.6.3 On the fiducial models of power spectrum and noise

According to equations (2.1) and (2.2), we need the power spectrum of the maps to compute the covariance matrix, but the power spectrum is precisely what we want to find using QML. In practice, we compute **C** from a *fiducial* power spectrum that we introduce by making use of the previous knowledge that we have of the map we want to analyze. Let us show that deviations of the fiducial model from the underlying true power spectrum will not introduce a relevant bias.

The coupled power of the anisotropies in the harmonic space is

$$y_\ell \equiv \mathbf{x}^t \mathbf{E}^\ell \mathbf{x} - b_\ell. \tag{2.57}$$

Since we are assuming that we start from a fiducial model to apply QML that differs to the model in the maps, we must differentiate the *true* covariance matrix of maps, $\mathbf{C}_m$, from the covariance matrix given by our fiducial, $\mathbf{C}_f$. The matrices $\mathbf{E}_\ell$ are defined in eq. (2.44) and taking into account that they are computed from the fiducial model, in fact, they become

$$\mathbf{E}^{\ell} \equiv \frac{1}{2}\mathbf{C}_f^{-1}\mathbf{P}_{\ell}\mathbf{C}_f^{-1}. \tag{2.58}$$

Inserting eq. (2.58) into eq. (2.57) we get

$$y_{\ell} \equiv \frac{1}{2}\mathbf{x}^t\mathbf{C}_f^{-1}\mathbf{P}_{\ell}\mathbf{C}_f^{-1}\mathbf{x} - b_{\ell}. \tag{2.59}$$

Taking averages and since $\langle x^t x\rangle = \mathbf{C}_m$, we get

$$\langle y_{\ell}\rangle \equiv \frac{1}{2}\,\mathrm{tr}\left[\mathbf{C}_f^{-1}\mathbf{P}_{\ell}\mathbf{C}_f^{-1}\mathbf{C}_m\right] - b_{\ell}. \tag{2.60}$$

Since the *true* covariance matrix of the maps is the sum of signal and noise, we get

$$\langle y_{\ell}\rangle \equiv \frac{1}{2}\,\mathrm{tr}[\mathbf{C}_f^{-1}\mathbf{P}_{\ell}\mathbf{C}_f^{-1}(\mathbf{S}_m + \mathbf{N})] - b_{\ell}, \tag{2.61}$$

where $\mathbf{S}_m$ is the true signal covariance matrix. Later on, we will make some considerations referring to the noise matrix but, for the time being, let us continue to refer to it only as $\mathbf{N}$, without distinguishing the *true* and *fiducial* noise matrices. From the last expression, as we did before, one can find the expression that cancels the noise bias

$$b_{\ell} = \frac{1}{2}\,\mathrm{tr}[\mathbf{C}_f^{-1}\mathbf{P}_{\ell}\mathbf{C}_f^{-1}\mathbf{N}]. \tag{2.62}$$

Writing the explicit form of $\mathbf{S}_m$ in terms of the *true* power spectrum, we get

$$\langle y_{\ell}\rangle = \sum_{\ell'} C_{\ell'}^{\mathrm{true}}\frac{1}{2}\,\mathrm{tr}\left[\mathbf{C}_f^{-1}\mathbf{P}_{\ell}\mathbf{C}_f^{-1}\mathbf{P}_{\ell'}\right]. \tag{2.63}$$

In the last expression one can recognize the element of the Fisher matrix given by our fiducial model, then

$$\langle y_{\ell}\rangle = \sum_{\ell'} C_{\ell'}^{\mathrm{true}}\mathbf{F}_{\ell\ell'}^{\mathrm{fid}}. \tag{2.64}$$

In vector form

$$\langle \mathbf{y}\rangle = \mathbf{F}^{\mathrm{fid}}\mathbf{c}^{\mathrm{true}}. \tag{2.65}$$

Therefore, from our fiducial, we can compute the estimator

$$\widehat{\mathbf{c}} = \left(\mathbf{F}^{\mathrm{fid}}\right)^{-1}\mathbf{y}, \tag{2.66}$$

that is unbiased

$$\langle\widehat{\mathbf{c}}\rangle = \left(\mathbf{F}^{\mathrm{fid}}\right)^{-1}\langle\mathbf{y}\rangle = \left(\mathbf{F}^{\mathrm{fid}}\right)^{-1}\mathbf{F}^{\mathrm{fid}}\mathbf{c}^{\mathrm{true}} = \mathbf{c}^{\mathrm{true}}. \tag{2.67}$$

In eq. (2.62) we have assumed that $\mathbf{N}$ is the *true* noise matrix. Otherwise, the b_{ℓ} term will not cancel the bias. That is, while there is some freedom in choosing the starting fiducial model, in principle, there is no freedom with respect to the noise.

On the other hand, QML is not optimal when the fiducial differs from the model on the maps. By distinguishing $\mathbf{C}_m$ from $\mathbf{C}_f$, eq. (2.52) becomes

$$\langle y_\ell y_{\ell'} \rangle - \langle y_\ell \rangle \langle y_{\ell'} \rangle = \frac{1}{2} \mathrm{tr}[\mathbf{C}_m \mathbf{C}_f^{-1} \mathbf{P}_\ell \mathbf{C}_f^{-1} \mathbf{C}_m \mathbf{C}_f^{-1} \mathbf{P}_{\ell'} \mathbf{C}_f^{-1}]. \tag{2.68}$$

Since the covariances of the y_l do not fit the elements of the Fisher matrix, the covariances of the estimated power spectrum do not fit the elements of the inverse of the Fisher matrix, so the estimator is unbiased but not of minimum variance when the fiducial differs from the model in the maps.

In Chapter 6, we analyze the effect of the choice of the fiducial in the recovered power spectra. In particular, we have studied how the estimations change when varying the fiducial model and, applying an iterative scheme, we have analyzed the convergence to a final and stable estimation starting from different initial models.

Most of the work presented in the next chapters is part of the results published in [70] and [71].

Chapter 3

The regularity of the covariance matrix

3.1 Motivation

With the advent of the precision era of Cosmology and ‘big data’ astrophysical experiments, data analysis techniques have risen to a prominent role in modern Astronomy. In particular, the treatment of very large matrices is a challenge from the point of view of algebraic and numerical methods, data storage and software implementation. For example, many astrophysical problems require the manipulation of large covariance matrices and their inverses. But it is often the case that such matrices are ill-conditioned and their inverse matrices cannot be calculated; when this happens, the problem must be attacked either by means of clever algorithms that calculate the pseudo-inverse of the matrix (see, e.g.,[72]) or by *ad hoc* regularizers such as the addition of a small amount of uncorrelated noise to the diagonal of the covariance matrix.

The literature is rich in situations where the regularity of the covariance matrix plays a fundamental role. For example, the inverse of the covariance matrix is necessary for the study of the statistics of the Cosmic Microwave Background (CMB), the Quadratic Maximum Likelihood (QML) power spectrum estimator [43], the maximum likelihood cosmological parameter estimation [73, 74], to study the topology of the universe [75] and for many CMB foreground removal/component separation methods (see, e.g., [76, 77, 78, 79]). Beyond the mere characterization of the second-order statistics of the CMB temperature or polarization fluctuations, covariance matrices are also fundamental for the study of non-Gaussianity [80, 81] and the statistical analysis of CMB anomalies such as the Cold Spot [82]. But the covariance matrix is often ill-conditioned even for low-resolution sky maps.

The typical solution consists on the regularization ‘by hand’ of the covariance matrix. For example, a small level of white noise is added to the noise covariance matrix of the CMB temperature data involved in the construction of the low-multipole *Planck* likelihood [74]. Another possibility is to deal with ill-conditioned covariance matrices by using a principal component analysis approach to remove the lowest degenerate eigenvalues, as is done for instance to study the multi-normality of the CMB [83] or in the estimation of primordial non-Gaussianity using wavelets [81]. The main problem with this kind of approaches is the *ad hoc* nature of the regularization. For example, the amount of uncorrelated artificial noise to be added to the covariance matrix must be carefully chosen: if the level of the noise is too small, it may not suffice to make the matrix regular, but if it is too large, the quality of the data is sorely compromised. One must find the correct amount of noise by trial and error.

A more fundamental question is why and how covariance matrices become ill-conditioned. By definition, all covariance matrices should be positive-semidefinite and symmetric. Therefore, a covariance matrix can be singular if it has, at least, one eigenvalue equal to zero. However, in practice it is often assumed that if a covariance matrix arising from CMB data is singular, it must be so because of numerical issues related to the way the inverse is computed and to limits in computer precision. In this chapter we will focus on the interesting case of observations on the sphere, and we will make a comprehensive study on how the regularity and rank of the covariance matrix depend on how it is built, the pixelization scheme, the sky coverage and the regularizing noise. We will show that, even with arbitrary high precision, CMB covariance matrices can be singular due to the way the CMB is sampled. As we will see in the following sections, the way the sphere is pixelized can introduce symmetries that affect the regularity of covariance matrices. This is a purely algebraic effect that, as far as we know, has not been explored in the literature before. We investigate these effects for five different pixelizations that have been used in the context of Cosmic Microwave Background (CMB) data analysis: `Cube`, `Icosahedron`, `Igloo`, `GLESP` and `HEALPix`, finding that, at least in the considered cases, the `HEALPix` pixelization tends to provide a covariance matrix with a rank closer to the maximum expected theoretical value than the other pixelizations. The effect of the propagation of numerical errors in the regularity of the covariance matrix is also studied for different computational precisions, as well as the effect of adding a certain level of noise in order to regularize the matrix. In addition, we investigate the application of the previous results to a particular example that requires the inversion of the covariance matrix: the estimation of the CMB temperature power spectrum through the Quadratic Maximum Likelihood algorithm. Finally, some general considerations in order to achieve a regular covariance matrix are also presented.

For simplicity, most of the chapter is focused on an only temperature case, and some considerations related to the full case —intensity and polarization– are made.

The structure of the chapter is as follows: Section 3.2 enunciates a general constraint in the rank of the covariance matrix. Section 3.3 presents further upper limits on the rank under the presence of some specific symmetries of the pixelization. These constraints are tested for five different pixelization schemes in Section 3.4. Section 3.5 studies the effect of introducing a realistic CMB model when computing the covariance matrix as well as the degrading effect of numerical precision. Section 3.6 investigates the addition of noise as a regularizer of the covariance matrix. The effect of the presence of a mask in the rank of the covariance matrix is addressed in 3.7. As an example, Section 3.8 applies the results of the previous sections to the Quadratic Maximum Likelihood method, which, as shown in Chapter 2, requires the calculation of the inverse of the covariance matrix. Concluding remarks are offered in Section 3.9. Finally, technical points related to the covariance matrix are presented in the appendices A, B and C.

3.2 The rank of the covariance matrix

If a discretized scalar field on the sphere arises from isotropic random fluctuations, the elements of the covariance matrix $\mathbf{C} \equiv \langle \mathbf{x}\mathbf{x}^t \rangle$ can be written in terms of the angular power spectrum C_ℓ as (see eqs. (1.46) and (1.44))

$$\mathbf{C}_{ij} = \sum_{\ell=0}^{\infty} C_\ell \sum_{m=-\ell}^{\ell} Y_{\ell m}(\theta_i, \phi_i) Y^*_{\ell m}(\theta_j, \phi_j), \tag{3.1}$$

where $Y_{\ell m}$ are the spherical harmonics given by

$$Y_{\ell m}(\theta, \phi) = \sqrt{\frac{2\ell+1}{4\pi}\frac{(l-m)!}{(l+m)!}} P_{\ell m}(\cos\theta) e^{im\phi}, \tag{3.2}$$

and $P_{\ell m}$ are the associated Legendre polynomials. In real-life applications, the sum over multipoles does not extend to infinity, but it has a cutoff at some ℓ_{max} instead. If the value of ℓ_{max} is not sufficiently large, the covariance matrix can be singular. This is a problem for all numerical applications that require the inversion of $\mathbf{C}$. On the other hand, a too large value of ℓ_{max} can lead to extremely heavy computational costs. Besides, there is a limit in the spatial resolution that can be explored, which is determined by the lowest angular distance between points in the pixelization, as established by the Nyquist's theorem. Therefore, one must be very careful with the choice of ℓ_{max}. In this section we will make a thorough study of the rank of $\mathbf{C}$ as a function of ℓ_{max}.

Let $\ell_{\rm min}$ and $\ell_{\rm max}$ be the summation limits in the finite version of eq. (3.1). Let us also consider a pixelization of the sky such that the unit sphere is sampled at a set of n positions $\{\hat{r}_i\}$ and angular coordinates $\{(\theta_i,\phi_i)\}$, $i=1,\ldots,n$. In order to simplify the notation in the following expressions, we introduce the univocal index change $(\ell,m)\leftrightarrow\mu$ for the spherical harmonics such that

$$\mu(\ell,m)=\sum_{k=\ell_{\rm min}}^{\ell-1}(2k+1)+\ell+m+1=\ell^2-\ell_{\rm min}^2+\ell+m+1. \tag{3.3}$$

This change assigns a unique μ to each pair (ℓ,m) in ascending m order, that is,

$$\mu(\ell_{\rm min},-\ell_{\rm min})=1,\quad \mu(\ell_{\rm min},-\ell_{\rm min}+1)=2 \tag{3.4}$$

and so on. The index μ runs from 1 to N, the total number of spherical harmonics, given by

$$N=\sum_{\ell=\ell_{\rm min}}^{\ell_{\rm max}}(2\ell+1)=\ell_{\rm max}^2-\ell_{\rm min}^2+2\ell_{\rm max}+1. \tag{3.5}$$

We will also use the same index μ for the power spectrum, such that C_μ corresponds to the power of the multipole ℓ obtained as

$$\ell=\text{floor}\left(\sqrt{\ell_{\rm min}^2-1+\mu}\right), \tag{3.6}$$

where floor(x) is the largest integer not greater than x. Using this notation, we can write

$$Y_{\ell m}(\theta_i,\phi_i)=Y_\mu(\theta_i,\phi_i)=Y_{\mu i}, \tag{3.7}$$

and thus

$$\mathbf{C}_{ij}=\sum_{\ell=\ell_{\rm min}}^{\ell_{\rm max}}C_\ell\sum_{m=-\ell}^{\ell}Y_{\ell m}(\theta_i,\phi_i)Y^*_{\ell m}(\theta_j,\phi_j)=\sum_{\mu=1}^{N}C_\mu Y_{\mu i}Y^*_{\mu j}, \tag{3.8}$$

where μ makes reference to the harmonic indexes, and i and j make reference to the pair of points on the sky at which the sum is being calculated. Using this notation, the $j^{\rm th}$ vector column of $\mathbf{C}$ is

$$\mathbf{C}_{\texttt{Col}_j}=\sum_{\mu_j}C_{\mu_j}\begin{pmatrix}Y_{\mu_j 1}\\ Y_{\mu_j 2}\\ \vdots\\ Y_{\mu_j n}\end{pmatrix}Y^*_{\mu_j j}, \tag{3.9}$$

and therefore **C** is

$$\mathbf{C}=\begin{pmatrix} \sum_{\mu_1} C_{\mu_1}\begin{pmatrix} Y_{\mu_1 1} \\ \vdots \\ Y_{\mu_1 n}\end{pmatrix} Y^*_{\mu_1 1} & \cdots & \sum_{\mu_n} C_{\mu_n}\begin{pmatrix} Y_{\mu_n 1} \\ \vdots \\ Y_{\mu_n n}\end{pmatrix} Y^*_{\mu_n n}\end{pmatrix}. \tag{3.10}$$

Using eq. (3.10) and the properties of matrix determinants, we can write the determinant of **C** as

$$|\mathbf{C}|=\sum_{\mu_1\cdots\mu_n=1}^{N}\left(\prod_{i=1}^{n} C_{\mu_i} Y^*_{\mu_i i}\right)\begin{vmatrix} Y_{\mu_1 1} & \cdots & Y_{\mu_n 1} \\ \vdots & & \vdots \\ Y_{\mu_1 n} & \cdots & Y_{\mu_n n}\end{vmatrix}. \tag{3.11}$$

We are interested in knowing whether or not $|\mathbf{C}| = 0$. A sufficient condition for this determinant to be zero is that all the determinants of the previous equation are null.

If we calculate **C** on n points on the sphere using N spherical harmonics, the number of elements of the sum given in eq. (3.11) is N^n. Any of those determinants can be different from zero if its columns are linearly independent. Since the necessary condition for this to be achieved is that each column corresponds to a different spherical harmonic, we have a first constraint on $\ell_{\max}$: N must be equal or greater than n. In other words, the number of spherical harmonics must be at least as large as the number of considered pixels on the sphere. The rank of the covariance matrix **C** is thus constrained by

$$\mathrm{rank}(\mathbf{C}) \leq \min\left(n, \ell_{\max}^2 - \ell_{\min}^2 + 2\ell_{\max} + 1\right). \tag{3.12}$$

Hereinafter we will refer to this constraint as R_0.

According to eq. (3.11), if one determinant in the sum is not null, there will be other $n! - 1$ terms that are not zero as well, which correspond to the permutations of its columns. However, we may wonder if the sum of all these elements could be zero. In Appendix A we show that the sum of these terms is a real positive number and, therefore, if there exists at least one non-null determinant, then $|\mathbf{C}| > 0$ is satisfied. Nonetheless, note that having $N \geq n$ does not guarantee that this is fulfilled. In particular, even if all the columns of the determinants correspond to different spherical harmonics, they can still be linearly dependent. A particular case when this can happen is under the presence of certain symmetries in the considered pixelization of the sphere. This is studied in more detail in the next section.

Referred to the full case —intensity and polarization— by inspecting eqs. (1.42) and (1.59), it is apparent that the intensity and polarization covariance matrices are of the same structure. This means that we can apply the same procedure to analyze the lower bound on the range. Note that, in this case, the columns of the determinants are taken from the the columns of the matrix **Y** of eq. (1.31) and eq. (1.32). Therefore, for the matrix **S** to be regular, the number of columns of the spherical harmonics matrix **Y** must be greater or equal than the size of the covariance matrix. Since the number of columns of **Y** is three times the number of columns of the matrix of spherical harmonics of intensity (see eq. (1.32)), and the size of the covariance matrix of the full case, eq. (1.40), is three times —T, Q and U parts— the size of the covariance matrix of intensity, the conclusion is that the rank is limited, again, by eq. (3.12).

3.3 The effect of symmetries on the rank of C

It is possible that the rank of **C** is reduced if the positions where the covariance matrix is calculated have certain symmetries. In this section we will study several of these symmetries, which appear in some commonly used pixelizations of the sphere, and their effect on the rank of **C**.

As detailed in Section 3.2, the determinant of **C** can be expanded as the sum of the determinants of several matrices according to eq. (3.11). It is easy to see that many of these matrices will have a null determinant but, if the sum runs up to a sufficiently large ℓ, some of them might have non-zero determinants. Rather than inspecting the rank of each individual matrix, it is possible to determine the existence (or non existence) of full-rank matrices in eq. (3.11) by inspecting the (non necessarily square) matrix, whose elements are given by

$$\mathbf{Y} = \begin{pmatrix} Y_{1,1} & \cdots & Y_{N,1} \\ \vdots & & \vdots \\ Y_{1,n} & \cdots & Y_{N,n} \end{pmatrix}, \tag{3.13}$$

where the harmonics are calculated for all the n pixels and N is the number of spherical harmonics. Therefore, **Y** is an $n \times N$ matrix, with as many rows as pixels and as many columns as spherical harmonics. For convenience, the value of the element ij of **Y** is the value of the spherical harmonic of index $\mu = j$ on the pixel $\hat{r}_i$, $\mathbf{Y}_{ij} = Y_{ji} = Y_j(\hat{r}_i)$.

It is easy to see that rank(**C**) = rank(**Y**). In particular, if $N \geq n$, the maximum possible rank of **Y** will be n. If this rank is achieved, this implies that there exists at least n linearly independent columns in **Y** and, thus, there is at least one determinant in eq. (3.11) together with its permutations which are different from zero.

With the help of the matrix **Y** and defining a diagonal signal matrix **S**, $\mathbf{S}_{\mu\nu} = C_\mu \delta_{\mu\nu}$, we can get to the same conclusion by a different route. The expression (3.8) can be written as a product of matrices, $\mathbf{C} = \mathbf{Y}\mathbf{S}\mathbf{Y}^{\dagger}$.[1] Since the rank of the product of matrices is lower than or equal to the minimum rank of the factors, for $\mathbf{C}$ to be regular the number of columns of $\mathbf{Y}$ (the number of spherical harmonics) has to be equal to or greater than the dimensions of $\mathbf{C}$ (the number of pixels).

In the next subsections we will study how the rank of $\mathbf{Y}$ is affected by some specific symmetries.

3.3.1 S_I symmetry: $\hat{r} \to -\hat{r}$

First of all, we will consider a pixelization such that all points on the sphere have a diametrically opposed point. For full-sky coverage, one would expect that this condition is in general fulfilled since it is reasonable that two symmetrical hemispheres are pixelized in the same way. However, this may not be the case when only partial coverage of the sky is considered.

Let us make the variable change $\cos\theta = z$ in $Y_{\ell m}(\theta,\phi)$. The spherical harmonics $Y_{\ell m}(z,\phi)$ have a well-defined parity with respect to the change $(z,\phi) \to (-z,\phi+\pi)$ (i.e., $\hat{r} \to -\hat{r}$)

$$Y_{\ell m}(-z,\phi+\pi) = (-1)^{\ell} Y_{\ell m}(z,\phi). \tag{3.14}$$

Assuming that we have a pixelization and sky coverage such that for each pixel with direction $\hat{r}$ there exists a pixel in the direction $-\hat{r}$, eq. (3.14) allows us to transform the matrix of eq. (3.13) into a block-diagonal matrix with the same rank as the matrix $\mathbf{Y}$

$$\mathbf{Y} \to \begin{pmatrix} \mathbf{Y}_e & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_o \end{pmatrix}, \tag{3.15}$$

where $\mathbf{Y}_e$ is a block of spherical harmonics of even ℓ calculated on half of the points of the pixelization, and $\mathbf{Y}_o$ contains the spherical harmonics with odd ℓ (see Appendix B.1 for details).

Therefore, we can write $\mathrm{rank}(\mathbf{Y}) = \mathrm{rank}(\mathbf{Y}_e) + \mathrm{rank}(\mathbf{Y}_o)$. Since the rank of each block is limited by the minimum value of the number of rows and the number of columns, the following constraint must be satisfied

$$\mathrm{rank}(\mathbf{Y}) \le \min(N_e, n/2) + \min(N_o, n/2), \tag{3.16}$$

where

$$N_e = \sum_{\substack{\ell=\ell_{\min} \\ (\ell\ \mathit{even})}}^{\ell_{\max}} (2\ell+1), \qquad N_o = \sum_{\substack{\ell=\ell_{\min} \\ (\ell\ \mathit{odd})}}^{\ell_{\max}} (2\ell+1). \tag{3.17}$$

---

[1] This expression can also be found in eq. (1.41).

Hereinafter we will refer to the relationship (3.16) as R_1. As will be shown below, this additional constraint can lead in certain cases to the reduction of the maximum rank that the covariance matrix can achieve. In particular, this could yield to a singular covariance matrix even in the case $N \geq n$.

3.3.2 S_{II} symmetry: $\phi \to \phi + \pi$

Let us assume that we have a pixelization that, in addition to the previous symmetry S_I, satisfies that for each point with coordinates (z, ϕ) there is a point with coordinates $(z, \phi + \pi)$.

Taking into account the following expression of the spherical harmonics

$$Y_{\ell m}(z, \phi + \pi) = (-1)^m Y_{\ell m}(z, \phi), \tag{3.18}$$

we can further expand the matrix **Y** in blocks (see Appendix B.2 for details)

$$\mathbf{Y} \to \begin{pmatrix} \begin{matrix} \mathbf{Y}_{eo} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_{ee} \end{matrix} & \mathbf{0} \\ \mathbf{0} & \begin{matrix} \mathbf{Y}_{oo} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_{oe} \end{matrix} \end{pmatrix}, \tag{3.19}$$

where, for example, $\mathbf{Y}_{eo}$ denotes the block made up of the spherical harmonics of even ℓ and odd m. The size of the blocks depends on the number of pixels at $z = 0$ in the pixelization. The reason is that the pair of points at $z = 0$ that satisfy the symmetry S_{II} also satisfy S_I, so they have to be accounted for in a slightly different way. In particular, Appendix B.2 shows that if the map contains u pixels at $z = 0$ and $0 \leq \phi < \pi$, and k pixels at $z > 0$ (note that $k + u = n/2$), then the blocks $\mathbf{Y}_{eo}$ and $\mathbf{Y}_{oe}$ have $k/2$ rows each, and the blocks $\mathbf{Y}_{ee}$ and $\mathbf{Y}_{oo}$ have $k/2 + u$ rows each (note that if $u = 0$, the number of rows of each block defaults to $n/4$). Therefore, the rank of the matrix will be limited by the following expression (R_2)

$$\begin{aligned} \operatorname{rank}(\mathbf{Y}) \leq{}& \min(N_{eo}, k/2) + \min(N_{ee}, k/2 + u) \\ &+ \min(N_{oo}, k/2 + u) + \min(N_{oe}, k/2), \end{aligned} \tag{3.20}$$

where N_{eo} is the number of spherical harmonics of even ℓ and odd m, N_{ee} is the number of spherical harmonics of even ℓ and even m, and so on.

3.3.3 S_{III} symmetry: $\phi \to \phi + \pi/2$

Let us assume that we have a pixelization that satisfies the symmetry S_I and that for each pixel at (θ, ϕ) there is a corresponding pixel at $(\theta, \phi + \pi/2)$. Note that, under these conditions, S_{II} is also fulfilled. Under the transformation $\phi \to \phi + \pi/2$ we have

$$Y_{\ell m}(\theta, \phi + \pi/2) = i^m Y_{\ell m}(\theta, \phi), \tag{3.21}$$

which can be used to transform each non-null block in eq. (3.19) into a new block-diagonal structure (see Appendix B.3). For example, the block $\mathbf{Y}_{eo}$ can be transformed

$$\mathbf{Y}_{eo} \to \begin{pmatrix} \mathbf{Y}_{e1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_{e3} \end{pmatrix}. \tag{3.22}$$

At the end of this process we have decomposed matrix $\mathbf{Y}$ into a matrix with eight blocks in the diagonal

$$\{\mathbf{Y}_{e1}, \mathbf{Y}_{e3}, \mathbf{Y}_{e0}, \mathbf{Y}_{e2}, \mathbf{Y}_{o1}, \mathbf{Y}_{o3}, \mathbf{Y}_{o0}, \mathbf{Y}_{o2}\}. \tag{3.23}$$

The corresponding rank is limited by the expression (R_3)

$$\begin{aligned} \mathrm{rank}(\mathbf{Y}) \leq {} & \min(N_{e1}, k/4) + \min(N_{e3}, k/4) \\ & + \min(N_{e0}, k/4 + u/2) + \min(N_{e2}, k/4 + u/2) \\ & + \min(N_{o1}, k/4 + u/2) + \min(N_{o3}, k/4 + u/2) \\ & + \min(N_{o0}, k/4) + \min(N_{o2}, k/4), \end{aligned} \tag{3.24}$$

where k and u have the same meaning as in Section 3.3.2. N_{eq} is the number of spherical harmonics of even ℓ and $q = \mathrm{mod}(m, 4)$.[2] N_{oq} is the number of spherical harmonics of odd ℓ and $q = \mathrm{mod}(m, 4)$. Note that the sum of the eight N_{pq} adds up the total number of spherical harmonics

$$\sum_{q=0}^{3} N_{eq} + \sum_{q=0}^{3} N_{oq} = \sum_{\ell=2}^{\ell_{\max}} 2\ell + 1. \tag{3.25}$$

Similarly, the sum of the pixels involved in the eight terms of eq. (3.24) adds up to $2k + 2u$, i.e., the total number of pixels n.

3.4 Results for different pixelization schemes

The particular way in which the sphere is pixelized gives rise to different symmetries and sets of pairs (θ, ϕ), and therefore will have an impact on the rank of the covariance matrix. In this section we will compare the theoretical ranks imposed by the

---

[2] $\mathrm{mod}(i, j) = i - j \times \mathrm{floor}(i/j)$ denotes the modulo of i and j. In our case, $q = \mathrm{mod}(m, 4)$ can take four different values corresponding to $\{0, 1, 2, 3\}$. Note that for positive m, this quantity is simply given by the rest of m divided by 4.

N_{pix}	12			48			192			768		
ℓ_{max}	2	3	4	5	6	7	12	13	14	26	27	28
Cons.												
R_0	5	12	12	32	45	48	165	192	192	725	768	768
R_1	5	11	12	32	42	48	165	186	192	725	761	768
R_2	5	11	12	32	42	48	165	186	192	725	761	768
R_3	5	11	12	32	42	48	165	186	192	725	761	768

N_{pix}	3072				12288			49152		
ℓ_{max}	53	54	55	56	109	110	111	220	221	222
Cons.										
R_0	2912	3021	3072	3072	12096	12288	12288	48837	49152	49152
R_1	2912	3018	3072	3072	12096	12246	12288	48837	49106	49152
R_2	2912	3017	3071	3072	12096	12246	12288	48837	49106	49152
R_3	2912	3017	3071	3072	12096	12246	12288	48837	49106	49152

Table 3.1: Theoretical maximum ranks R_{th} of the covariance matrix, for a generic pixelization of the sphere, imposed by the constraints given in the previous section. Different number of pixels (N_{pix}) and maximum multipole (ℓ_{max}) are considered.

constraints introduced in the previous sections with those obtained for different pixelization schemes. In particular, we will consider five different pixelizations that have been used to analyze CMB data: `Cube` [84], `Icosahedron` [85], `Igloo` [86], `GLESP` [87] and `HEALPix` [39].

However, before considering these particular pixelization schemes, let us study the case of a generic pixelization that successively fulfills symmetries S_I, S_{II} and S_{III}. We can easily calculate the maximum rank for the covariance matrix for a certain maximum multipole ℓ_{max} given by the corresponding constraints. Table 3.1 shows the theoretical maximum ranks achieved by the covariance matrix for different values of N_{pix} and ℓ_{max}, depending on which symmetries are satisfied. In particular, for all cases, the minimum ℓ_{max} necessary to achieve a non-singular matrix (i.e., whose rank is at least as large as N_{pix}) is given. To allow for an easier comparison, we have chosen values for N_{pix} equal to the number of pixels of the first resolutions of the `HEALPix` and `Igloo` pixelizations (see sections 3.4.5 and 3.4.3 for details). In addition, the constraints imposed by the S_{II} and S_{III} symmetries also depend on the particular number of pixels at $z = 0$. For the sake of simplicity, we have decided this generic pixelization to have the same number of pixels at $z = 0$ as `HEALPix`. As one would expect, including symmetries in the pixelization implies, at least in certain cases, that a higher ℓ_{max} is needed in order to achieve a regular covariance matrix.

Although the ranks given in table 3.1 correspond to the maximum ranks that the covariance matrix can achieve under the presence of the corresponding symmetries, in practice, a specific pixelization may have additional properties or cancellations. This reduces even further the rank for a given multipole, which implies that we need a higher $\ell_{\max}$ for this matrix to become regular. In the following subsections we will study the actual rank of the covariance matrix for five specific pixelizations, considering different resolutions. To carry out this analysis, we have calculated numerically the rank of **C** using the `MatrixRank` function of the symbolic computation software Mathematica. Hereinafter we will refer to this value as R_N. For comparison, this value is confronted with that obtained from the theoretical constraints, taking into account the symmetries present in each of the considered pixelizations, which we will denote R_{th}.

It is worth noting that the particular elements of the covariance matrix depend on the power spectrum of the fluctuations but, according to eq. (3.11), its rank does not (except in the null case of a blank image).[3] This allows us to choose the C_ℓ's and, for the sake of simplicity, in this section we will compute **C** for a flat spectrum $C_\ell = 1$. For this simplified case, making use of the addition theorem for spherical harmonics, eq. (3.8) takes the form

$$\mathbf{C}_{ij} = \sum_{\ell=2}^{\ell_{\max}} \frac{2\ell+1}{4\pi} P_\ell(\hat{r}_i \cdot \hat{r}_j) = \sum_{\ell=2}^{\ell_{\max}} \mathbf{P}_\ell(\hat{r}_i \cdot \hat{r}_j). \tag{3.26}$$

As expected, in all the considered cases we have found $R_N \equiv \text{rank}(\mathbf{C}) = \text{rank}(\mathbf{Y})$.

3.4.1 `Cube`

The `Cube` pixelization [84], represented in figure 3.1, maps the pixels from the surface of a cube to the unit sphere in such a way that their areas are approximately equal. The successive resolution levels are attained by recursive subdivision of the projected faces of the cube, controlled by the resolution parameter res, which can take positive integer values. The number of pixels is given by $N_{\text{pix}} = 6 \times 4^{\text{res}-1}$. The minimum resolution, six pixels at the centers of the six faces of the cube, is given by res = 1. This pixelization has been extensively used during the processing of the COBE experiment [88].

In this pixelization all pixels have a S_I-symmetric pixel. They also satisfy the S_{III} symmetry, except for two pixels on the poles for the first resolution level. Table 3.2 shows the values of the rank of the covariance matrix obtained numerically (R_N) and the maximum theoretical rank (R_{th}) as a function of $\ell_{\max}$. The pixels of resolutions

---

[3] Note also that, in practice, very small values of the power spectrum can introduce numerical errors in the calculation of the rank of **C**, giving rise to ill-conditioned matrices in cases expected to be non-singular from theoretical arguments (see Section 3.5 for details).

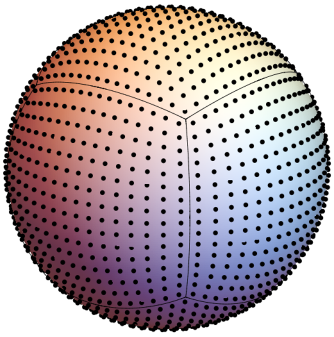


Figure 3.1: Cube pixelization corresponding to res = 5 and $N_{\text{pix}} = 1536$.

(res, N_{pix})	(1,6)			(2, 24)				(3,96)						
$\ell_{\max}$	2	3	4	2	3	4	5	5	6	7	8	9	10	11
R_N	2	5	6	5	12	19	24	32	45	60	75	88	94	96
R_{th}	3	6	6	5	12	19	24	32	45	60	77	92	96	96

Table 3.2: Numerically calculated rank for the covariance matrix (R_N) compared to the maximum expected theoretical rank (R_{th}) for different configurations of the Cube pixelization.

2 and 3 satisfy the S_{III} symmetry, thus the maximum theoretical rank is given by R_3. For res = 1, the theoretical ranks have been calculated taking into account that the S_I symmetry is satisfied and that there are two pairs of pixels that also satisfy S_{III}.

The table shows that for some $\ell_{\max}$ we have $R_N \leq R_{\text{th}}$. In order to understand why the maximum range is not always achieved, let us consider as a workable example the case res = 1, $\ell_{\max} = 2$, where we find $R_N = 2$ versus $R_{\text{th}} = 3$. For this resolution level we have six pixels located at $(\pm1,0,0),(0,\pm1,0),(0,0,\pm1)$ in Cartesian coordinates. Since **C** is calculated using the harmonics of $\ell = 2$, **Y** is a matrix with 6 rows (number of pixels) and 5 columns (number of harmonics). Following the arguments of the previous section, it is convenient to order the spherical harmonics following the sequence $\{e1, e3, e0, e2\}$ (note that odd values of ℓ are not considered since $\ell_{\min} = \ell_{\max} = 2$). For our case, this corresponds to an ordering of m of $\{1,-1,0,-2,2\}$. Pixels are ordered (0, 0, 1), (1, 0, 0), (0, 1, 0), (-1, 0, 0), (0, -1, 0), (0, 0, -1). In this way, carrying out the corresponding operations, we get the matrix **Y**

$$\mathbf{Y} = \begin{pmatrix} 0 & 0 & \frac{\sqrt{\frac{5}{\pi}}}{2} & 0 & 0 \\ 0 & 0 & -\frac{\sqrt{\frac{5}{\pi}}}{4} & \frac{1}{4}\sqrt{\frac{15}{2\pi}} & \frac{1}{4}\sqrt{\frac{15}{2\pi}} \\ 0 & 0 & -\frac{\sqrt{\frac{5}{\pi}}}{4} & -\frac{1}{4}\sqrt{\frac{15}{2\pi}} & -\frac{1}{4}\sqrt{\frac{15}{2\pi}} \\ 0 & 0 & -\frac{\sqrt{\frac{5}{\pi}}}{4} & \frac{1}{4}\sqrt{\frac{15}{2\pi}} & \frac{1}{4}\sqrt{\frac{15}{2\pi}} \\ 0 & 0 & -\frac{\sqrt{\frac{5}{\pi}}}{4} & -\frac{1}{4}\sqrt{\frac{15}{2\pi}} & -\frac{1}{4}\sqrt{\frac{15}{2\pi}} \\ 0 & 0 & \frac{\sqrt{\frac{5}{\pi}}}{2} & 0 & 0 \end{pmatrix}. \tag{3.27}$$

The first two columns of zeros occur because Y_2^1 and Y_2^{-1} contain the product $\sin\theta\cos\theta$, which becomes null for the 6 considered pixels. All points are symmetric in the sense $\hat{r} \to -\hat{r}$, and therefore we can use eq. (3.14) and transform **Y** into a matrix with 3 rows of zeros. Choosing the pixels $(0,0,1)$, $(1,0,0)$ and $(0,1,0)$ when applying the symmetry we get

$$\mathbf{Y} \to \begin{pmatrix} 0 & 0 & \frac{\sqrt{\frac{5}{\pi}}}{2} & 0 & 0 \\ 0 & 0 & -\frac{\sqrt{\frac{5}{\pi}}}{4} & \frac{1}{4}\sqrt{\frac{15}{2\pi}} & \frac{1}{4}\sqrt{\frac{15}{2\pi}} \\ 0 & 0 & -\frac{\sqrt{\frac{5}{\pi}}}{4} & -\frac{1}{4}\sqrt{\frac{15}{2\pi}} & -\frac{1}{4}\sqrt{\frac{15}{2\pi}} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3.28}$$

Then, the maximum rank we could ever achieve is 3. Note that $Y_2^2 = Y_2^{-2} = 0$ at $(0,0,1)$, so the first row in $\mathbf{Y}_e$ has only one non zero term, at the column that corresponds to Y_2^0. Taking into account that the other two points are S_{III} symmetric, we can get more zero blocks in $\mathbf{Y}_e$. Therefore, the first three rows of the matrix become

$$\mathbf{Y}_e \to \begin{pmatrix} 0 & 0 & \frac{\sqrt{\frac{5}{\pi}}}{2} & 0 & 0 \\ 0 & 0 & -\frac{\sqrt{\frac{5}{\pi}}}{4} & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{4}\sqrt{\frac{15}{2\pi}} & -\frac{1}{4}\sqrt{\frac{15}{2\pi}} \end{pmatrix}, \tag{3.29}$$

which yields to a matrix of rank 2, a unit of rank lower than expected taking into account the size of Y_e after applying S_I. The reason for this lower rank for this particular pixelization is that two columns of spherical harmonics, Y_2^{-1} and Y_2^1, are zeros and the columns of Y_2^2 and Y_2^{-2} are equal. This illustrates how in some cases the rank can be lower than expected by the theoretical expressions when computing the rank of matrices **Y** calculated on particular sets of pixels.

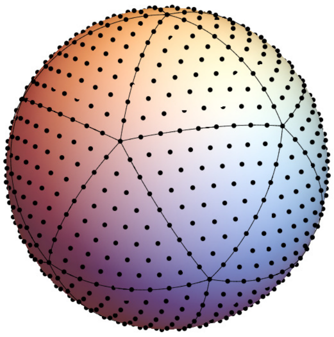


Figure 3.2: `Icosahedron` pixelization corresponding to res = 5 and N_{pix}=812.

(res, N_{pix})	(1, 12)					(2, 92)									
$\ell_{\max}$	2	3	4	5	6	3	4	5	6	7	8	9	10	11	12
R_N	5	8	8	11	12	12	21	32	45	60	77	87	88	91	92
R_{th}	5	11	12	12	12	12	21	32	45	60	77	90	92	92	92

Table 3.3: Numerically calculated rank for the covariance matrix (R_N) compared to the maximum expected theoretical rank (R_{th}) for different configurations of the `Icosahedron` pixelization.

3.4.2 `Icosahedron`

The `Icosahedron` pixelization [85], shown in figure 3.2, is constructed by subdividing the faces of an icosahedron into a regular triangular grid, starting from a first resolution of 12 pixels situated at the vertices of the icosahedron. The pixels are projected onto the sphere in such a way that their areas are approximately equal. Similarly to the `Cube` pixelization, the resolution of the pixelization is controlled by an integer positive parameter res, being the number of pixels given by $N_{\text{pix}} = 40 \times \text{res} \times (\text{res} - 1) + 12$.

This pixelization satisfies the S_I symmetry. However, since the vertices of the Icosahedron, which are the starting point of the subsequent resolutions, do not fulfill symmetries S_{II} or S_{III}, no further symmetries are expected to be found. In particular, we have tested that this is the case up to resolution res = 9. Table 3.3 shows the values of the rank of the covariance matrix obtained numerically (R_N) and the maximum theoretically expected range (R_{th}) as a function of $\ell_{\max}$ for two different resolutions. Note that for the shown resolutions the S_I symmetry is satisfied and, therefore, $R_{\text{th}} = R_1$. As for the `Cube` pixelization, $R_N \leq R_{\text{th}}$ is always satisfied. The reason why the maximum rank is not always achieved can be easily seen in the lowest resolution (res = 1, $N_{\text{pix}} = 12$).

If we sum up to $\ell_{\max} = 2$, the rank of the matrix is limited by the number of spherical harmonics, which is 5, and we have found $R_N = 5$. If we sum up to $\ell_{\max} = 3$, there are 7 new harmonics, thus the rank could increase, in principle, up to 12. However, due to the presence of the S_I symmetry, we have that the maximum theoretical rank is $R_{\text{th}} = 11$. Nevertheless, the numerical rank R_N is found to be eight. One of the reasons for the reduction of the rank is that the z-coordinate of all pixels take one of the values $z = \{-1, 1, -\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{5}}\}$ and the Legendre polynomials for $\ell = 3$ and $m = \pm 1$,

$$P_3^{-1}(z) = \frac{1}{8}\sqrt{1-z^2}\left(5z^2 - 1\right), \qquad P_3^1(z) = -\frac{3}{2}\sqrt{1-z^2}\left(5z^2 - 1\right), \tag{3.30}$$

happen to cancel at these points, which leads to two zero columns in the $\mathbf{Y}_o$ block. Thus we are left with only five non-zero columns instead of seven for this block, which means that the covariance matrix rank could be ten as maximum. However, due to the particular positions of the pixels in this pixelization scheme, other columns of this block happen to be linearly dependent. In particular, the column that corresponds to $m = 0$ is linearly independent of the other four columns, but the $m = -3$ column is proportional to the $m = -2$ column and the same relation applies to columns $m = 3$ and $m = 2$. Therefore, the rank of the odd block is reduced to three, and adding up the rank of the even block, which is five, we finally get rank eight for the covariance matrix as found. Note that adding multipoles up to $\ell_{\max} = 4$ does not change the situation, but going up to $\ell_{\max} = 5$ raises the rank to 11. Finally, adding multipoles up to $\ell_{\max} = 6$ provides enough independent columns to get a non-singular covariance matrix.

3.4.3 `Igloo`

The `Igloo` pixelation [86] divides the sphere into bands of variable width in θ. Each band is divided into a different number of pixels at constant intervals of ϕ, and the interval varies from band to band. Thus, the edges of the pixels are of constant latitude and longitude. Among the different options, the one presented here consists of a base resolution of twelve pixels, three on each of the poles and six on the central band. The angles have been chosen in such a way that the resultant pixels are of equal area. The subsequent resolutions were obtained by recurrent subdivisions of each pixel into other four pixels of equal area, always having three pixels at each pole.

The number of pixels is $N_{\text{pix}} = 12 \times N_{\text{side}}^2$. The resolution is controlled by the N_{side} parameter, which can take as values integers that are a power of 2. Figure 3.3 shows the `Igloo` pixelization for $N_{\text{side}} = 2$. The Igloo pixels are symmetric in the S_I sense. Moreover, for all resolutions there are couples of S_{II}-symmetric pixels in the central area.

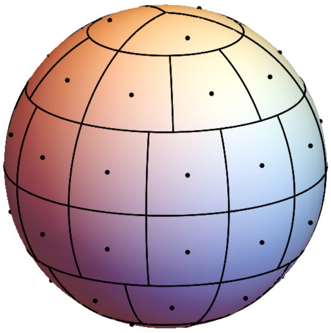


Figure 3.3: `Igloo` pixelization corresponding to N_{side}=2 and N_{pix}=48.

$(N_{\text{side}}, N_{\text{pix}})$	(4, 192)														
$\ell_{\max}$	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
R_N	5	12	21	32	45	60	77	96	117	140	162	179	187	191	192
R_{th}	5	12	21	32	45	60	77	96	117	140	165	186	192	192	192

Table 3.4: Numerically calculated rank for the covariance matrix (R_N) compared to the maximum expected theoretical rank (R_{th}) for different configurations of the `Igloo` pixelization.

For resolutions 2 and higher, there are also some pixels with S_{III} symmetry. Taking into account these particular symmetries, we find that $R_N = R_{\text{th}} = R_1$ for all $\ell_{\max}$ for resolutions $N_{\text{side}} = \{1, 2\}$. For $N_{\text{side}} \geq 4$ the numerical rank of the covariance matrix is found to be smaller than the maximum theoretical rank in some cases. Table 3.4 shows the results for $N_{\text{side}} = 4$ and different values of $\ell_{\max}$.

3.4.4 GLESP

The Gauss-Legendre Sky Pixelization (GLESP) [87] (figure 3.4) makes use of the Gaussian quadratures to evaluate numerically the integral with respect to z of the expression

$$a_{\ell m} = \int_{-1}^{1} dz \int_{0}^{2\pi} d\phi \Delta T(z, \phi) Y_{\ell m}^{*}(z, \phi), \tag{3.31}$$

which can be formally expressed with this method in an exact form as a weighted finite sum. The surface of the sphere is divided into N rings of trapezoidal pixels with values of θ at the centers of the pixels according to the Gauss-Legendre quadrature method. There is some degree of freedom to fix the number and size of the pixels on the rings, but preferentially they are defined to make the equatorial ones roughly square. The

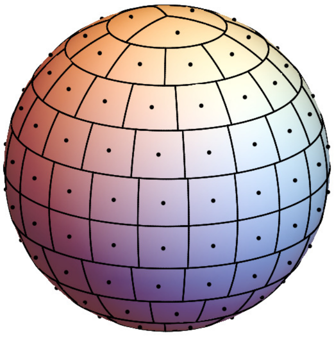


Figure 3.4: GLESP pixelization corresponding to a resolution parameter $N = 10$. In this figure the rings have been rotated to make the pixels S_I-symmetric.

number of pixels on the rest of the rings is chosen in such a way that all the pixels have nearly equal area.[4] Finally, polar pixels are triangular. Note that this pixelization is not hierarchical.

Regarding the presence of symmetries, those pixels belonging to rings which have been divided into a number of pixels which is a multiple of two or four will satisfy S_{II} or S_{III}. However, this is not always the case and, therefore, the GLESP pixelization does not fulfill these symmetries as a whole. Referring to the S_I-symmetry, it can be fulfilled in certain cases if the rings are rotated with respect to the z-axis (note that this rotation does not change the essence of the pixelization). In particular, if the number of rings is even, each of the rings on one hemisphere can be positioned with respect to the opposite ring on the other hemisphere in such a way that the S_I-symmetry is satisfied. If N is an odd number, there is an unmatched ring on the equator, and the symmetry is only fulfilled if this ring is divided into an even number of pixels. Fig. 3.4 shows the GLESP pixelization for $N = 10$ in a S_I-symmetric configuration.

Table 3.5 shows the values of the rank for the case with four rings. The number of pixels is 22, eight on each of the central rings and three on each of the poles. For the sake of simplicity and in order to allow for a better comparison with the other considered pixelizations, we have rotated the rings in such a way that all the pixels have their S_I-symmetric pair. The pixels on the two central rings have a S_I, S_{II} and S_{III}-

---

[4] It is interesting to note that to study the CMB polarization and with the aim of obtaining a better evaluation of the ${}_{0,\pm2}a_{\ell,m}$-coefficients, two additional over-pixelization versions of this scheme have been proposed [89]. In this case, there is a larger number of pixels on the rings than in the nearly equal area scheme considered in this chapter, with decreasing area of the pixels on the rings as they get closer to the polar caps. Since our work is oriented to the scalar field case, we will study the nearly equal area version of the pixelization.

(N, N_{pix})	(4, 22)				
$\ell_{\max}$	2	3	4	5	6
R_N	5	12	17	21	22
R_{th}	5	12	18	22	22

Table 3.5: Numerically calculated rank for the covariance matrix (R_N) compared to the maximum expected theoretical rank (R_{th}) for the `GLESP` pixelization for $N = 4$.

symmetric partner, but the pixels on the poles do not have a S_{II} and S_{III}-symmetric one. In this situation, the R_{II} and R_{III} expressions are not strictly valid. However, if we find that $R_I = R_{II} = R_{III}$, i.e., that the matrix rank is not reduced under the presence of S_{II} and S_{III}, we can conclude that the same is valid when the symmetries are only partially fulfilled, as for the `GLESP` pixelization. For the considered case, we have indeed found $R_I = R_{II} = R_{III}$ for all the values of $\ell_{\max}$. Table 3.5 shows that for $\ell_{\max} = 4$ and $\ell_{\max} = 5$ the rank of $\mathbf{C}$ is one unit lower than the theoretical rank. Let us try to explain where this unit is lost. Since the S_I-symmetry is fulfilled, we have two diagonal blocks in $\mathbf{Y}$. For $\ell_{\max} = 4$, the block of odd ℓ has 11 rows and seven columns, and its rank is seven. The block of even ℓ, 11 rows and 14 columns, and its rank is ten. Due to the way in which the values of θ are chosen in this pixelization scheme, the roots of the Legendre polynomial with $\ell = 4$ in the case of $N = 4$, the column of $\mathbf{Y}$ that corresponds to the harmonic $Y_{4,0}$ is made up of zeros. Aside from this column, among the other 13 columns there are only 10 linearly independent columns. We have found that the group of ten that corresponds to the five harmonics of values of $\ell = 2$ and the other five of $\ell = 3$ and values of m : -4, -3, -1, 1 and 3 are linearly independent. When $\ell_{\max} = 5$, we get 11 new columns on the odd block of $\mathbf{Y}$, and the rank in this block saturates to 11, but the block of even ℓ still has 11 rows and 14 columns but only ten linearly independent columns; thus the rank is a unit lower than the theoretical value. When $\ell_{\max} = 6$ we get new linearly independent columns in the block of even ℓ, and the rank saturates.

3.4.5 `HEALPix`

The most extensively used pixelization for CMB analysis is `HEALPix`, which stands for Hierarchical Equal Area iso-Latitude Pixelization [39]. `HEALPix` divides the sphere into twelve spherical diamond-shaped pixels on three rings, one of them on the equator and the other two towards the poles. The four pixels forming each ring have the same

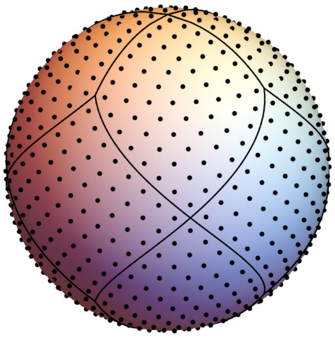


Figure 3.5: `HEALPix` pixelization corresponding to $N_{\rm side} = 8$ and $N_{\rm pix}$=768.

$(N_{\rm side}, N_{\rm pix})$	(8, 768)			
$\ell_{\rm max}$	25	26	27	28
R_N	672	724	760	768
$R_{\rm th}$	672	725	761	768

Table 3.6: Numerically calculated rank for the covariance matrix (R_N) compared to the maximum expected theoretical rank ($R_{\rm th}$) for the `HEALPix` pixelization for $N_{\rm side} = 8$ in the multipole regime in which **C** becomes regular.

latitude and each diamond is then subdivided recursively in order to get the pixels for the different resolution levels. The total number of pixels is $N_{\rm pix} = 12 \times N_{\rm side}^2$, where $N_{\rm side}$ is the resolution parameter, which is always an integer power of 2. Figure 3.5 shows the `HEALPix` pixelization for $N_{\rm side}$=8. By construction all the pixels satisfy the S_I, S_{II} and S_{III} symmetries. We find that for the three lowest resolutions ($N_{\rm side} = 1, 2$ and 4), the covariance matrix always achieve the maximum possible rank, i.e., $R_N = R_{\rm th} = R_3$ for all $\ell_{\rm max}$. The first reduction of rank occurs for $N_{\rm side} = 8$ and $\ell_{\rm max} = 26$ (see table 3.6). In particular, when $\ell_{\rm max} = 26$ we have $R_{\rm th} = 725$ and $R_N = 724$. Let us focus on this case in more detail. We have found that the loss of rank of the covariance matrix is found to lie in block $\mathbf{Y}_{e2}$. The number of rows in block $\mathbf{Y}_{e2}$ is determined by the quantity $k/4 + u/2$, while the number of columns is given by the number of spherical harmonics with even ℓ and $mod(m,4) = 2$ (see Appendix B.3 for details). In particular, for $N_{\rm side} = 8$ we have a total of 768 pixels. Among these, 32 have $z = 0$, one half of them with $0 \leq \phi < \pi$ (i.e. $u = 16$). We also have 368 pixels with $z > 0$ (i.e., $k = 368$). Therefore, the block $\mathbf{Y}_{e2}$ has $k/4 + u/2 = 100$ rows. For $\ell_{\rm max} = 26$, the number of spherical harmonics with ℓ even and $mod(m,4) = 2$ is 98. So the block has size 100×98, but rank 97. We have also checked that the rank of all the sub-blocks that can be formed by removing any particular column of $\mathbf{Y}_{e2}$ is 97. This means that any group of 97 out of the 98 spherical harmonics are linearly independent, but the 98 columns taken as a whole are linearly coupled. This shows, again, how some units of rank can be lost by properties related to the particular values of the pixels of the chosen pixelization.

So far we have demonstrated, for five different pixelization schemes, that for a given resolution parameter, it is possible to determine the lowest $\ell_{\max}$ needed to make $\mathbf{C}$ regular. Furthermore, a detailed analysis of the pixel properties allows us to detect linear combinations of spherical harmonics that lead to reductions of the maximum theoretical rank. When this happens, it is necessary to raise $\ell_{\max}$ in order to regain regularity.

Among the five pixelization schemes we have considered, `HEALPix` is particularly interesting because of two nice properties. On the one hand, from the tests we have presented, it seems to be the pixelization in which the reduction of the rank of the covariance matrix with respect to the maximum expected theoretical value is lowest. On the other hand, since it presents the three types of considered symmetries for all the resolution levels, it is easy to divide the $\mathbf{Y}$ matrix into eight blocks, which in turn makes it easier to study the linear independence of the columns of the matrix. For these reasons, together with the fact that `HEALPix` is the most commonly used pixelization for CMB analysis, we will only consider this pixelization along the rest of the chapter.

The discussion so far has focused on an idealized case. In particular, there is a number of real-life issues we have not taken into account yet:

- Until now, we have used $C_\ell = 1$ and neglected the window function B_ℓ.
- In many real-life applications, $\mathbf{C}$ is calculated for a masked area of the sky.
- We have not considered the effect of noise.

We will discuss the impact of these effects on the regularity of $\mathbf{C}$ in the next sections.

3.5 Effect of the power spectrum and the beam transfer function

Up to now we have studied the rank of $\mathbf{C}$ according to eq. (3.8) using a simplified angular power spectrum $C_\ell = 1$. In a realistic CMB analysis, we would use the corresponding

ℓ_{max}	24	25	26	27	28	29	30	31
$R_N(1)$	621	672	724	760	768	768	768	768
$R_N(2)$	621	671	709	735	752	761	766	768

Table 3.7: Numerically calculated ranks of **C** for a `HEALPix` resolution of $N_{\text{side}} = 8$ and different values of ℓ_{max}. $R_N(1)$ is the rank of **C** using $C_\ell = 1$ and not including any beam window function. $R_N(2)$ is the rank of the same matrix when considering a realistic CMB power spectrum and beam.

C_ℓ's and the beam transfer function B_ℓ which encodes the response of the beam of the experiment (see Section 2.5). Therefore

$$\mathbf{C}_{ij} = \sum_\ell \frac{2\ell+1}{4\pi} C_\ell B_\ell^2 P_\ell(\hat{r}_i \cdot \hat{r}_j). \tag{3.32}$$

The beam transfer function usually drops quickly for high multipoles and the sum of eq. (3.32) is dominated by low-ℓ terms. As we mentioned before, theoretically this should not affect the rank of **C**. However, in practice, since the sums are computed numerically, it is possible that the regularity of **C** is affected by machine precision limitations. In table 3.7 we compare the rank of **C** as calculated in the previous section (i.e., C_ℓ=1 for all multipoles and not including any window function) with that calculated using a realistic CMB power spectrum, for the `HEALPix` pixelization with resolution $N_{\text{side}} = 8$. In particular, we have considered the *Planck* ΛCDM best-fit model that includes also information from the *Planck* lensing power spectrum reconstruction and Baryonic Acoustic Oscillations [90],[5] a Gaussian beam with a FWHM corresponding to 2.4 times the pixel size at the considered resolution (FWHM=17.6 degrees for $N_{\text{side}} = 8$) and we have also taken into account the corresponding `HEALPix` pixel window function. This model will be used along the chapter, unless otherwise stated. As before, we have estimated the rank of the covariance matrix using MATHEMATICA with its default machine precision (which corresponds to the standard double precision). From the results of table 3.7, we see that the rank estimated for the covariance matrix can be significantly smaller when working under realistic conditions for the power spectrum, indicating that numerical errors may have an important effect in real-life applications. We can further study the effect of numerical precision on the effective rank of **C** by means of the MATHEMATICA software. MATHEMATICA can operate the sums of spherical

---

[5] More specifically we have used model 2.30 described in the document https://wiki.cosmos.esa.int/planckpla2015/images/f/f7/Baseline_params_table_2015_limit68.pdf from the Planck Explanatory Supplement

$\ell_{\max}$	Machine	50	200	400	800
26	-7.70×10^{-615}	2.95×10^{-2713}	-2.49×10^{-9329}	1.44×10^{-18046}	1.05×10^{-35734}
27	-9.50×10^{-390}	-4.58×10^{-759}	4.90×10^{-1979}	-1.74×10^{-3545}	-4.80×10^{-6767}
28	6.91×10^{-255}	3.12×10^{-256}	3.12×10^{-256}	3.12×10^{-256}	3.12×10^{-256}
29	1.64×10^{-174}	2.62×10^{-174}	2.62×10^{-174}	2.62×10^{-174}	2.62×10^{-174}
30	5.54×10^{-136}	5.66×10^{-136}	5.66×10^{-136}	5.66×10^{-136}	5.66×10^{-136}
31	4.39×10^{-122}	4.40×10^{-122}	4.40×10^{-122}	4.40×10^{-122}	4.40×10^{-122}

Table 3.8: Determinant of **C** computed with different precisions and $\ell_{\max}$.

harmonics symbolically and convert the result into numerical format with any desired number of decimals. Those quantities which are available with a limited precision (e.g. the power spectrum or the pixel window function) can be rationalized in MATHEMATICA, which helps to control the propagation of numerical errors.[6] In this way, calculating the covariance matrix given by eq. (3.32) with a precision of 200 decimals we recover the ranks given by $R_N(1)$ in table 3.7. This confirms that the loss of rank observed in table 3.7 is, in fact, a matter of numerical errors.

The effect of numerical precision can be observed in more detail in table 3.8. We have calculated the determinant of **C** for a realistic power spectrum for different values of $\ell_{\max}$ and different numerical precisions: the machine native precision and 50, 200, 400 and 800 decimals. For $\ell_{\max} = 26$ and $\ell_{\max} = 27$ the determinant must be zero because the theoretical rank is lower than 768. This fact is easily seen at higher numerical precisions, where the value of the determinant quickly drops to zero. Conversely, at $\ell_{\max} = 28$, 29 and 30, the determinant is non-zero, and, even if very small, we see that its value is stable when sufficiently high precision is used. In particular, the default machine precision is not enough to get the correct value of the determinant (considering a precision of two decimal places), although it gets closer to the correct value as $\ell_{\max}$ increases. This relates to our previous finding showing that the estimated rank is lower than expected for these values of $\ell_{\max}$ when using the default machine precision, and it becomes correct for higher numerical precisions.

3.6 Effect of noise

Another common complexity on CMB data is the presence of instrumental noise. Let us now consider the effect of adding a zero-mean noise $\mathbf{n}$ to the signal $\mathbf{s}$. Given that

---

[6] Of course this comes at the cost of increasing the computational time.

noise and signal are expected to be statistically independent, the covariance matrix of the data $\mathbf{x} = \mathbf{s} + \mathbf{n}$ is simply given by the sum of the signal ($\mathbf{S}$) and noise ($\mathbf{N}$) covariance matrices. Moreover, if the noise is not spatially correlated, the matrix $\mathbf{N}$ is diagonal. In that case

$$\mathbf{C}_{ij} = \mathbf{S}_{ij} + \mathbf{N}_{ij} = \sum_{\ell=\ell_{\min}}^{\ell_{\max}} \frac{2\ell+1}{4\pi} C_\ell B_\ell^2 P_\ell(\hat{r}_i \cdot \hat{r}_j) + \sigma_i^2 \delta_{ij}. \tag{3.33}$$

Taking into account that each column of $\mathbf{C}$ is given by the sum of two columns, we can expand by columns its determinant. In this way, the expansion takes the form of the sum of 2^n determinants, one of them containing only columns from $\mathbf{S}$, a second one containing only columns from $\mathbf{N}$ and the rest containing columns from both matrices. The determinants that contain columns from $\mathbf{S}$ can be either positive or null, depending on the value of $\ell_{\max}$ and the pixel configuration, but the term that depends only on columns from $\mathbf{N}$ is always positive. It can also be easily shown[7] that the terms with columns from both matrices are also greater or equal than zero. Therefore, we have

$$\det \mathbf{C} \geq \det \mathbf{N} > 0. \tag{3.34}$$

This shows that the presence of noise regularizes the covariance matrix.[8] This is a well known fact, and in many applications it is common to use a small amount of artificial noise to help regularizing a numerically ill-behaved covariance matrix [74, 91]. The question arising in these cases is what is the optimal noise level to be introduced in order to make the matrix regular without degrading too much the quality of the data.

We have carried out some tests in order to illustrate the regularizing effect of adding noise to the covariance matrix for four different resolutions ($N_{\text{side}} = 4$, 8, 16 and 32) of the `HEALPix` pixelization. We have used the same *Planck* ΛCDM model as in the previous section for the CMB power spectrum and a Gaussian beam with FWHM equal to 2.4 times the pixel size at each resolution. The pixel window function given by `HEALPix` has also been taken into account. For each case we have performed the summations up to the $\ell_{\max}$ value for which $\mathbf{C}$ is regular according to the constraint R_3. We have also added different levels of isotropic white noise according to $\sigma^2 = 10^{-f_n}$ $(\mu\text{K})^2$, to be compared with the CMB variance given in the fourth column of table 3.9. As in the previous section, the rank of the covariance matrix for each of the considered cases has been obtained using MATHEMATICA with its default machine precision and is given in table 3.9.

---

[7] By interchanging rows and columns and computing the resultant determinant.

[8] Note that although this reasoning has been made for the only intensity case, it also holds for the full —intensity and polarization— case, because it is still true that $\mathbf{C} = \mathbf{S} + \mathbf{N}$.

N_{side}	N_{pix}	l_{max}	$\mathbf{S}_{ii}$	No noise	$f_n = 9.5$	9.0	8.5	8.0	7.5	7.0
4	192	14	487	**192**	192	192	192	192	192	192
8	768	28	1057	752	752	756	**768**	768	768	768
16	3072	56	1655	2937	2937	2937	2938	**3072**	3072	3072
32	12288	111	2308	11518	11518	11518	11518	11518	**12288**	12288

Table 3.9: Rank (R_N) of **C** for several `HEALPix` resolutions after adding different levels of noise. The columns show, from left to right: the resolution parameter, the corresponding number of pixels, the ℓ_{max} used for the calculation of **C** (which is the theoretical multipole that makes the matrix regular), the value of the diagonal element $\mathbf{S}_{ii}$ (i.e., the CMB variance, in units of $(\mu\mathrm{K})^2$), the matrix rank without noise and the one obtained after adding noise with different values of the f_n parameter such that $\sigma^2 = 10^{-f_n}$. The boldface indicates the level of noise at which the matrix **C** becomes regular.

We find that for $N_{\mathrm{side}} = 4$, the covariance matrix is regular even without the presence of noise, whereas for higher resolutions it becomes necessary to add a certain level of noise for the matrix to be regular. Note that the required level of noise grows with N_{side}, although it is always very small in comparison to the value of the CMB variance and, therefore, it is not expected to compromise the quality of the data.

In many scientific applications, as for example the estimation of the power spectrum through the Quadratic Maximum Likelihood method (Chapter 2) it is necessary to perform operations with **C** or its inverse. Such operations propagate numerical errors and may lead to instabilities even if one starts with a regular matrix. In figure 3.6 we show the result of the operation $-\log\left(\mathrm{abs}\left(1-|\mathbf{C}\mathbf{C}^{-1}|\right)\right)$, with **C** calculated using the theoretical ℓ_{max} value that makes the matrix regular (third column in table 3.9), as a function of the noise level. Positive values of this quantity indicate approximately the decimal place at which the determinant $|\mathbf{C}\mathbf{C}^{-1}|$ departs from unity, while negative values show how many orders of magnitude this determinant (in absolute value) is greater than unity. This gives an indication of the level of noise required to operate with the covariance matrix and its inverse within the required precision.

3.7 Effect of masking

The first effect of masking is a reduction of the number of pixels over which the covariance matrix is computed, and therefore a smaller dimension of **C**. Consequently, a lower value of ℓ_{max} will be needed in general to obtain a regular matrix. A second more subtle effect is the possible breaking of pixel – spherical harmonic symmetries.

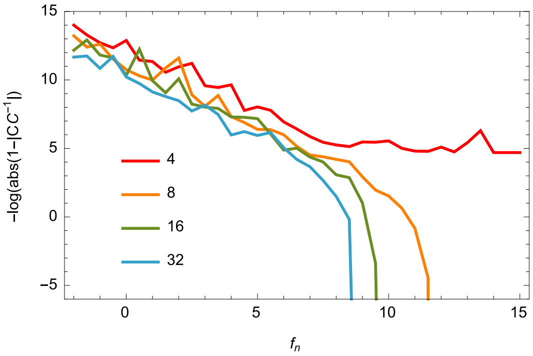


Figure 3.6: Measurement of the departure of the numerical value of $|\mathbf{CC}^{-1}|$ from unity versus the level of added noise, for $N_{\rm side}$= 4, 8, 16 and 32 (see text for details).

In Section 3.3 we studied the effect of three kinds of symmetries, S_I, S_{II} and S_{III}. We showed that, among them, S_I has the greatest impact on the rank of the covariance matrix. Therefore, for the sake of simplicity, in this section we will study the effect of masking only under the presence of the S_I symmetry.

Let us consider that, for a fixed resolution parameter and a given mask geometry, we have n valid pixels. Let us also consider that, among these pixels, n_p have their symmetric pixel inside the region allowed by the mask, while n_u pixels do not have it. Taking into account the symmetries of the n_p pixels the matrix $\mathbf{Y}$ can be transformed as described in Section 3.3

$$\mathbf{Y} \to \begin{pmatrix} \mathbf{Y}^{even}_{n_p/2\times N_e} & \mathbf{0}_{n_p/2\times N_o} \\ \\ \mathbf{0}_{n_p/2\times N_e} & \mathbf{Y}^{odd}_{n_p/2\times N_o} \\ \\ \mathbf{Y}^{even}_{n_u\times N_e} & \mathbf{Y}^{odd}_{n_u\times N_o} \end{pmatrix}, \tag{3.35}$$

where, for example, $\mathbf{Y}^{even}_{n_p/2\times N_e}$ stands for a block with $n_p/2$ rows and N_e columns, whose elements are the spherical harmonics of even ℓ calculated over the independent half of symmetrical pixels inside the mask. For the matrix of eq. (3.35) to be of maximum rank (see Appendix C for details), it is necessary that

$$N_e \geq n_p/2, \;\; N_o \geq n_p/2, \tag{3.36}$$

due to the pixel symmetry, and

$$N_e + N_o \geq n_p + n_u, \tag{3.37}$$

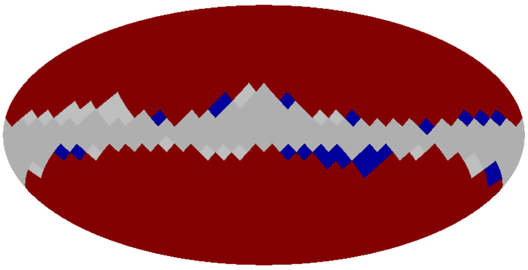


Figure 3.7: Location of the pixels outside the `SEVEM` mask for $N_{\rm side} = 8$. The pixels allowed by the mask with an S_I symmetrical partner are red, while those without a symmetric partner are blue. The light and dark gray pixels are discarded by the mask. In particular, the light gray pixels show the points symmetrical to the unpaired blue ones.

so that there are at least as many spherical harmonics as pixels. The previous expressions allow us to find the minimum $\ell_{\rm max}$ that makes the matrix regular in a case with mask and S_I symmetry.

We can also calculate the maximum rank of a matrix given the number of pixels, the mask, the number of pixels with and without a symmetric partner and the value of $\ell_{\rm max}$. Transforming the matrix of eq. (3.35) into diagonal blocks, in Appendix C we show that the rank matrix is constrained by

$$\mathrm{rank}(\mathbf{Y}) \leq \min(n_p/2, N_e) + \min(n_p/2, N_o) + \min(n_u, L), \tag{3.38}$$

were $L = N_e - \min(n_p/2, N_e) + N_o - \min(n_o/2, N_e)$. The expression (3.38), R_M hereafter, generalizes expression (3.16) for any mask configuration.

As an example, let us consider the case of $N_{\rm side} = 8$ in the `HEALPix` pixelization and the mask produced by the `SEVEM` component separation method [79] corresponding to the first *Planck* data release.[9] The `SEVEM` mask removes 154 pixels of the sky at this resolution level. Among the 614 remaining pixels, 590 have a S_I symmetrical pixel, whereas 24 do not have a symmetrical partner. Figure 3.7 shows the locations of paired and unpaired pixels for this case.

Table 3.10 shows the maximum expected rank $R_{\rm th}$ of the covariance matrix according to R_M and those calculated with Mathematica using a realistic CMB power spectrum and beam (R_N). In addition, two different precisions are considered: the native precision of the machine and a 100-digit precision. We find that the rank estimated with Mathematica coincides in all the considered cases with the maximum allowed

---

[9] All *Planck* products are publicly available at the Planck Legacy Archive, http://pla.esac.esa.int/pla.

$l_{\max}$	21	22	23	24	25	26
R_{th}	480	525	572	614	614	614
R_N (default)	480	525	571	602	611	614
R_N (100)	480	525	572	614	614	614

Table 3.10: Rank of **C** for the `HEALPix` pixelization at the $N_{\text{side}} = 8$ resolution using the `SEVEM` mask for different values of $\ell_{\max}$. R_{th} is the maximum theoretical rank according to R_M, and R_N is the rank of **C** calculated with Mathematica using the default and 100-digit precisions.

theoretical rank when using the higher precision, whereas numerical errors affect the calculation using the default precision.

3.8 Application to the QML method

As a working example, let us consider the case of `HEALPix` pixelization at resolution $N_{\text{side}} = 32$ and full-sky. The total number of pixels is 12288 and, according to the theoretical values given in table 3.1, it is necessary to sum up to $\ell_{\max} = 111$ in order to get the maximum rank of **C**. However, from table 3.9 we know that with this $\ell_{\max}$ the numerical matrix is not regular. A solution for this problem is to add a *small* amount of noise to make **C** regular. However, what does *small* mean in this context? If the artificial noise is too large, the performance of the QML method will suffer degradation, particularly at high multipoles, but if the noise level is too small the matrix will pose numerical instabilities. Therefore the choice of the correct noise level is not trivial. In this section we will study in detail the effect of noise addition and investigate the optimal *small* amount of noise to be added.

In particular, we will study the value of the determinant $|\mathbf{C}\mathbf{C}^{-1}|$, the statistical properties of the product $\eta = \mathbf{x}^t\mathbf{C}^{-1}\mathbf{x}$ (which, since both the simulated CMB and noise are Gaussian, should follow a χ^2 distribution with the same number of degrees of freedom as of pixels in the map), and the estimation and error bar of the power spectrum given by the QML method after the addition of different levels of noise. In order to do this, we first compute the matrix **S** in eq. (3.33) using the *Planck* best-fit model (and including the beam and pixel window function) with $\ell_{\max} = 111$ and, secondly, we add isotropic white noise with variance $\sigma^2 = 10^{-f_n}$; CMB maps are then simulated with the `HEALPix` package at $N_{\text{side}} = 32$, and noise is added following a normal distribution with dispersion parameterized by f_n.

f_n	% noise	$\langle\sigma_\ell^{QML}/\sigma_\ell^{CV}\rangle_\ell$	p-value	$\lvert\mathbf{CC}^{-1}\rvert$	$-\log\left(\text{abs}\left(1-\lvert\mathbf{CC}^{-1}\rvert\right)\right)$
0	2.08	2.54	0.63	1.00000000006	10.22
1	0.66	1.16	0.93	1.0000000007	9.13
2	0.21	1.02	0.66	1.000000003	8.49
3	6.6×10^{-2}	1.01	0.73	0.999999993	8.16
4	2.1×10^{-2}	1.01	0.73	1.000001	5.98
5	6.6×10^{-3}	1.01	0.73	0.999998	5.95
6	2.1×10^{-3}	1.01	0.73	1.000009	5.05
7	6.6×10^{-4}	1.01	0.60	1.0002	3.66
8	2.1×10^{-4}	1.01	0.11	0.97	1.50
9	6.6×10^{-5}	1.01	$\sim10^{-15}$	1.27×10^{36}	-36.10
9.25	4.9×10^{-5}	1.13	$\sim10^{-15}$	-1.12×10^{299}	-299.05
10	2.1×10^{-5}	$\sim7\times10^{5}$	$\sim10^{-14}$	∞	$-\infty$

Table 3.11: Summary of statistical tests obtained from 10000 CMB simulations (at $N_{\text{side}} = 32$) with different levels of regularizing noise. The first column shows the value of the noise parameter f_n, $\sigma^2_{noise} = 10^{-f_n}$. The second column gives the percentage of noise with respect to the dispersion of the CMB signal, i.e., $100\times\sigma_{noise}/\sqrt{S_{ii}}$. The third column shows the average over ℓ of the errors on the estimation of D_ℓ divided by those corresponding to the cosmic variance, eq. (3.39). The p-value corresponds to the probability that the quantity $\mathbf{x}^t\mathbf{C}^{-1}\mathbf{x}$ follows the expected χ^2 distribution according to the Cramér-von Mises test. The two last columns show the results in relation to the determinant of $\lvert\mathbf{CC}^{-1}\rvert$.

Table 3.11 shows the value of the determinant $\lvert\mathbf{CC}^{-1}\rvert$ (fifth column) for different levels of regularizing noise as well as the quantity $-\log\left(\text{abs}\left(1-\lvert\mathbf{CC}^{-1}\rvert\right)\right)$ (sixth column), introduced in Section 3.6. It is apparent that for very low levels of noise, the covariance matrix becomes singular and the determinant produces values which are very far from unity. Conversely, for values of $f_n \lesssim 5$, the determinant departs from its theoretical value approximately in the sixth decimal place (or even better). Note that $f_n = 5$ corresponds to a very small level of noise compared to the signal (second column) and, therefore, it is not expected to imply a degradation of the data. Of course, the more noise we include, the closer the determinant gets to unity, but at the price of degrading the data.

The product $\eta = \mathbf{x}^t\mathbf{C}^{-1}\mathbf{x}$ should follow a χ^2 distribution with 12288 degrees of freedom. However, if there are numerical instabilities in $\mathbf{C}^{-1}$, this quantity will depart from its theoretical distribution. In figure 3.8 we show two histograms of η obtained from 10000 simulations of CMB plus noise with two different levels of noise in comparison with the expected χ^2 distribution. The $f_n = 5$ noise results in a regular covariance

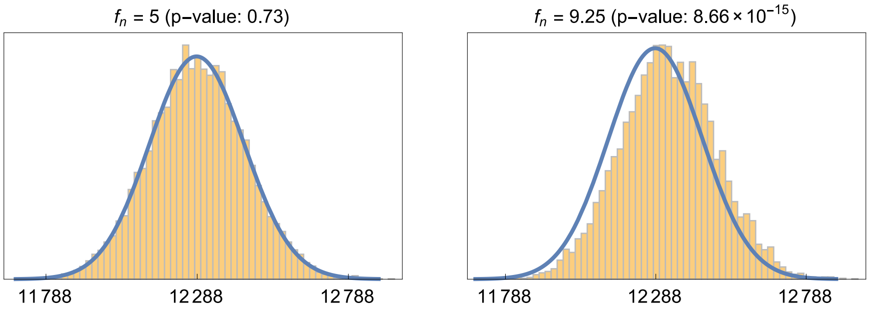


Figure 3.8: Distribution of numerical values of $\mathbf{x}^t\mathbf{C}^{-1}\mathbf{x}$ compared with the theoretical distribution, for noise levels $f_n = 5$ and 9.25 obtained from 10000 simulations.

matrix, and $f_n = 9.25$ is in the limit in which the tests fail dramatically. It clearly shows how the distribution of η departs from the theoretical distribution when the level of the noise is low. This can be quantified, for instance, by applying the Cramér-von Mises goodness-of-fit hypothesis test [92], which determines the p-value of the data to be consistent with the theoretical distribution. The p-values obtained for the different considered cases are given in table 3.11 (fourth column). As one would expect, the data are consistent with a χ^2 distribution only when the covariance matrix is regularized by adding a certain level of noise to the simulated data.

So far, we have studied two generic statistics directly related to the covariance matrix. We will now investigate how the possible singularity of $\mathbf{C}$ propagates through the QML estimator. Figure 3.9 shows the average and 1σ error for $\widehat{D_\ell}$ when considering three different levels of regularizing noise, namely, $f_n = 0$, 5 and 9.25. If the noise level is too small, the estimation of the low multipoles is systematically biased downwards. If the noise is too high, the performance of the estimator degrades at high ℓ and the error bars grow quickly. At intermediate noise levels, the estimator is unbiased and the performance of the method is basically limited only by cosmic variance, up to $\ell \approx 3N_{\text{side}}$.

To further quantify the performance of the QML method under the presence of different levels of regularizing noise, we have calculated the following quantity

$$\left\langle \frac{\sigma_\ell^{QML}}{\sigma_\ell^{CV}} \right\rangle = \frac{1}{3N_{\text{side}}-1} \sum_{\ell=2}^{3N_{\text{side}}} \frac{\sigma_\ell^{QML}}{\sigma_\ell^{CV}}, \tag{3.39}$$

where σ_ℓ^{QML} corresponds to the dispersion of the QML estimator and σ_ℓ^{CV} to that inferred from the cosmic variance. This average ratio gives information on how much

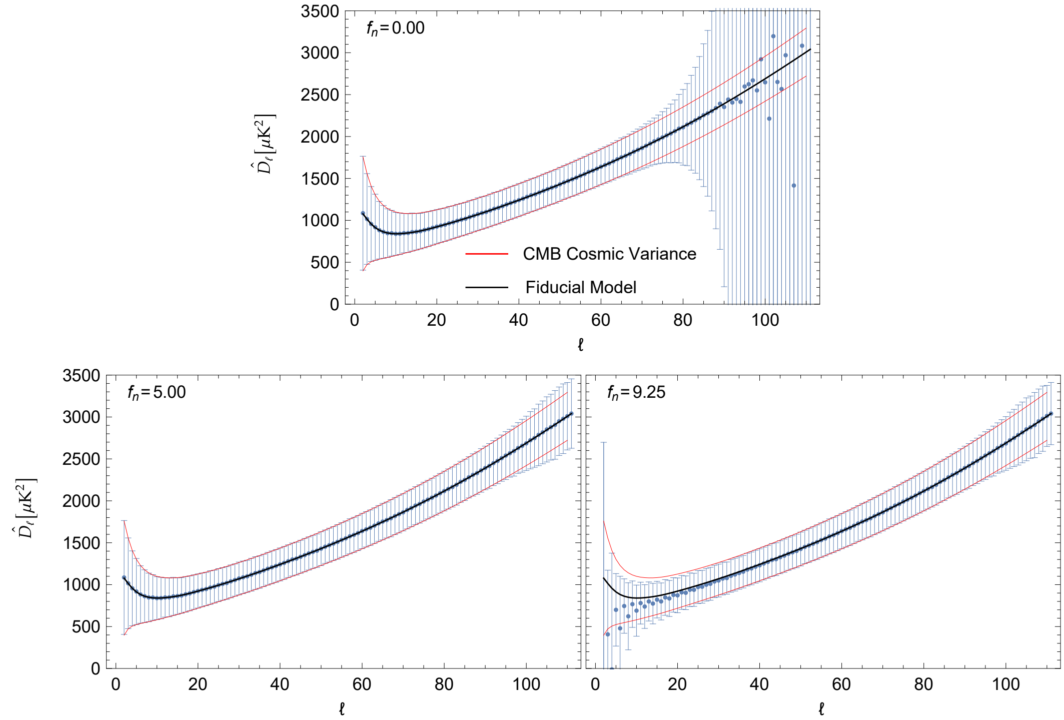


Figure 3.9: Power spectrum estimation given by the QML method for three different levels of noise, $f_n = \{0, 5, 9.25\}$.The average and 1σ error bars have been obtained from a set of 10.000 simulations.

the error in the estimation of the QML degrades with respect to the optimal case. Table 3.11 shows this ratio for different levels of added noise. For a very low level of noise, the performance of the estimator is clearly degraded due to the singularity of $\mathbf{C}$, while for a high level of noise, its effect in increasing the error of the estimator at high multipoles also becomes apparent. There is an intermediate region at which the error in the estimation is very close to the error expected by cosmic variance, i.e., $\langle \sigma_\ell^{QML} / \sigma_\ell^{CV} \rangle \approx 1$.

By looking simultaneously at the results for the three previous tests, we can get a better idea of which is the optimal level of noise to be added in order to regularize the covariance matrix without degrading the data for the particular case studied in this section. The hypothesis test based on $\mathbf{x}^t\mathbf{C}^{-1}\mathbf{x}$ and the determinant $|\mathbf{C}\mathbf{C}^{-1}|$ probe similar properties, since both of them focus directly on the inverse of $\mathbf{C}$. As one would expect, the values in table 3.11 show that the results from both statistics are consistent. In particular, when the noise is high, both tests show that the covariance matrix is regular, while, when the noise is reduced, the determinant departs from unity and the p-value of the Cramér-von Mises test gets close to zero. Note that the more noise is added, the better the discriminant is, but this does not indicate the level of noise at which the quality of the data starts to get compromised. Therefore, it is necessary to look at other complementary statistics (such as the one based on the QML method) in order to find the appropriate range of noise, sufficiently high as to regularize the covariance matrix but also as low as possible so that the data is not degraded. By studying the error bar in the estimation of the power spectrum with QML with respect to that expected by cosmic variance, we find that very low values of regularizing noise ($f_n = 9$) can give an apparent good result in this test but a bad value for the determinant. Thus, for the sake of consistency and being conservative, one should choose a value of noise that achieves both a good value for the determinant and a minimum error for the QML method. In this sense, table 3.11 shows that values of f_n between 3 and 5 are the most appropriate; the error bar criterion gives the lowest value for the error size, the determinant departs from unity at the sixth decimal position (in the worst case), and the hypothesis test shows consistency with the expected theoretical distribution.

For a generic application, one could give some general considerations to proceed in order to have a regular covariance matrix and to help establishing the appropriate level of regularizing noise, if necessary, in each case:

1. Taking into account the number of pixels, the symmetries and if there is a mask, calculate the minimal theoretical multipole ℓ_{tm} up to which is necessary to sum in the calculation of **C** in order to get a regular matrix, using the adequate theoretical expression among the ones given in this chapter.

2. If the real value up to which we sum in the calculation of **C**, $\ell_{\max}$, is lower than the previously determined multipole, ℓ_{tm}, noise certainly needs to be added. Even if it is not lower, we might need to add some regularizing noise due to numerical errors.

3. If necessary, add an amount of noise. Table 3.11 may help to choose an initial guess.

4. Calculate the determinant of $\mathbf{C}\mathbf{C}^{-1}$ and evaluate how much it deviates from unity. One should bear in mind that a very good determinant can also be the consequence of adding an excessive amount of noise.

5. If the determinant is good, study the performance of the considered application (in our case the QML method) for this covariance matrix and compare it with the expected theoretical result.

6. If either the value of the determinant or the performance of your application is not as good as expected, iterate by changing the amount of regularizing noise until both goals are achieved.

3.9 Conclusions

We have presented a comprehensive study of the regularity of the covariance matrix **C** of a discretized intensity field on the sphere. For a general case, we have shown that a necessary condition for the determinant of **C** to be different from zero is that the number of considered harmonics is equal or greater than the number of pixels. The presence of specific symmetries for the considered pixelization or the use of a mask that excludes some regions of the sphere also impose additional constraints on the rank of **C**. Along this work, five different expressions that establish limits on the maximum rank achieved by the covariance matrix are presented; they depend on the number of pixels, the number of spherical harmonics, the kind of symmetries of the considered pixelization and the presence of a mask.

When putting in practice these expressions for five different pixelizations proposed within the context of CMB analysis (`Cube`, `Icosahedron`, `Igloo`, `GLESP` and `HEALPix`), we have found that the particular properties of the location of the pixels can lead to a reduction of the rank of the covariance matrix with respect to the maximum rank allowed by the previously derived constraints. Interestingly, among the five studied pixelizations, `HEALPix` seems to present a lower reduction of the rank than the rest of pixelizations, which should help to achieve the regularity of the covariance matrix more easily.

We have also tested that numerical error propagation can give rise to a loss of rank, which would produce a singular covariance matrix in cases where this is not expected theoretically. By studying this effect with different numerical precisions, it is possible to differentiate whether the reduction of the rank is due to the propagation of numerical errors or to an intrinsically singular covariance matrix. As it is well known, a possible solution to mitigate the effect of numerical errors is to add some level of uncorrelated noise in order to regularize the covariance matrix. We have tested that even small levels of noise can be sufficient, although the particular value of the required noise will depend on the considered case.

We have also considered a practical case in which the calculation of the inverse of **C** is required, namely, the estimation of the CMB temperature power spectrum using the QML estimator. In particular, we have considered simulations containing CMB and different levels of noise using the `HEALPix` pixelization at resolution $N_{\rm side} = 32$. The optimal level of noise which needs to be included to obtain a regular matrix without degrading the quality of the data has been investigated in detail. We have studied particularly the behavior of three different statistics in order to evaluate the quality of the results versus different levels of regularizing noise: the value of the determinant of $\mathbf{CC}^{-1}$, a hypothesis test to study the distribution of the quantity $\eta = \mathbf{x}^t\mathbf{C}^{-1}\mathbf{x}$ (which should follow a χ^2 distribution if $\mathbf{x}$ represents Gaussian random fields) and the increment in the error of the estimation of the power spectrum with respect to the ideal case (given by the cosmic variance). The first conclusion from this particular exercise is that some level of noise, even if small, must be introduced in order to get a regular matrix, since numerical errors make the matrix singular even if this was not expected from the theoretical point of view. As the level of noise increases, the determinant and the hypothesis test indicate that the covariance matrix becomes more regular (for instance, it is observed that the determinant of $\mathbf{CC}^{-1}$ departs less from unity). However, increasing the level of noise also degrades the quality of the data, which is reflected in the increment of the error in the estimation of the CMB power spectrum with the QML estimator. Also note that the error of this estimator can give good results even if the determinant criterion is bad. Therefore, for the sake of consistency, it becomes necessary to look simultaneously at all the statistics in order to establish the optimal level of noise for the error in the QML estimation to be as small as possible, while ensuring the covariance matrix is well behaved. For this particular case, this can be achieved with a level of noise as small as ~ 0.01 per cent with respect to the CMB dispersion (see table 3.11), although the particular value will depend on the considered case as well as on the required precision for the covariance matrix.

Some general guidelines concerning the steps to follow to get a regular matrix for a generic application are given in Section 3.8.

Finally, referred to the full case —intensity and polarization— it can also be inferred that the rank of the covariance matrix is limited by $\ell_{\rm max}$ in the same fashion as the intensity matrix and that the addition of noise regularizes the matrix.

Chapter 4

The binned QML estimator

4.1 Motivation

It is well known that when working with an incomplete sky coverage, coupling appears between the different multipoles and the errors in the estimation of the power spectrum increase. In addition, there is a limit in the achieved angular resolution $\Delta\ell$ that depends on the size of the considered patch. As the mask grows, we are left with less information due to the pixels discarded. At some point, this leads to the Fisher matrix becoming singular. For instance, this is usually the case when applying QML to a typical ground configuration.

As shown in Section 2.4, the QML method consists on two stages: (i) compute the vector $\mathbf{y}$ from the anisotropies of the map that contains the coupled power of harmonic space (eq. 2.45) and (ii) decouple the mixing of power making use of the inverse of the Fisher matrix (i.e., $\widehat{\mathbf{c}} = \mathbf{F}^{-1}\mathbf{y}$). If this matrix is singular, the last step cannot be completed. However, we will show that an optimal estimator of a reduced number of variables (binned spectra) can still be defined. This can be understood as an extension of the method to estimate bandpower spectra.

4.2 Description of the estimator for binned spectrum

Let us assume that we have computed all the quantities in the harmonic space required by QML, but that we cannot invert the coupling in the $\mathbf{y}$ variables (see eq. (2.49)) because $\mathbf{F}$ is singular. From the point of view of a system of linear equations, eq. (2.49) defines a problem where $\mathbf{c}$ is the vector of variables to be solved. When $\mathbf{F}$ is singular, the system has more variables than linearly independent equations and, therefore, is under-determined. To reduce the number of variables, we can consider a set of bandpowers B_b^{XY}, which are linear combinations of the original D_ℓ^{XY} variables, and solve for them.

For the sake of simplicity, let us consider for the rest of the section the case of only temperature. Note that the process is the same in the full temperature and polarization case, but working with a higher dimension.

Let us take N_{bins}, indexed by b, with boundaries $\ell_{\text{low}}^b < \ell_{\text{high}}^b = \ell_{\text{low}}^{b+1} - 1$, and define $L_b = \{\ell_{\text{low}}^b, \ell_{\text{low}}^b + 1, \ldots, \ell_{\text{high}}^b\}$ the set of values of ℓ corresponding to bin b. To define the bandpower of a given bin, one usually calculates the weighted mean of the multipoles in the bin. If this is the case, the variables B_b will be the mean power of the multipoles in each one of the bins weighted by their theoretical errors [93], taking into account cosmic variance, noise and sky fraction (see Appendix D for a discussion on the validity of this error with regard to the sky fraction). In this case, we also have to compute the mean value of the $\ell \in L_b$ to find the position of the value representative of the bin, ℓ_b^*. Therefore, assuming that the fiducial is described in terms of the variables D_ℓ —and therefore, that we are implementing QML to get the estimation in terms of these variables— we can define the bandpowers of the binned fiducial as

$$B_b = \frac{\sum_{\ell \in L_b} \frac{D_\ell}{(\Delta D_\ell)^2}}{\sum_{\ell \in L_b} \frac{1}{(\Delta D_\ell)^2}} \tag{4.1}$$

where ΔD_ℓ corresponds to the theoretical error of the power spectrum. The position of the representative value of the bin is given by

$$\ell_b^* = \frac{\sum_{\ell \in L_b} \frac{\ell}{(\Delta D_\ell)^2}}{\sum_{\ell \in L_b} \frac{1}{(\Delta D_\ell)^2}}. \tag{4.2}$$

Let us note that other choices for B_b are possible. For example, one could pick the D_ℓ of the fiducial corresponding to the central value of ℓ in the bin ($B_b = D_{\ell_c}$) or use an unweighted mean of the values of the multipoles. As we will see, the binned QML constructed consistently with these different definitions will provide as output the considered binned power. Whichever the choice for the variables B_b, once they are fixed, the next step is to define a set of factors f_ℓ^b, making use of the information in the fiducial model, D_ℓ as

$$f_\ell^b = D_\ell / B_b. \tag{4.3}$$

Introducing the factors f_ℓ^b and the variables B_b in eq. (2.49) we get (assuming again that QML is implemented in terms of the variables D_ℓ)

$$\begin{aligned}\langle y_\ell \rangle &= \sum_{\ell'} F_{\ell\ell'} D_{\ell'} = \sum_{b'} \sum_{\ell' \in L_b'} F_{\ell\ell'} f_{\ell'}^{b'} B_{b'} \\ &= \sum_{b'} B_{b'} \sum_{\ell' \in L_b'} F_{\ell\ell'} f_{\ell'}^{b'} . \end{aligned} \tag{4.4}$$

Now this system of linear equations has $\ell_{\max} - 1$ equations and N_{bins} variables B_b. Thus it contains more equations than variables. To reduce the number of equations accordingly, we can simply combine them linearly. Although, as it will be shown later, the most efficient estimator can be obtained by a particular combination of the equations, for the purpose of illustration, let us combine them now by just summing together the $\ell_{\text{high}}^b - \ell_{\text{low}}^b + 1$ equations of each bin. This leads to N_{bins} equations

$$\begin{aligned}\sum_{\ell \in L_b} \langle y_\ell \rangle &= \sum_{\ell \in L_b} \sum_{b'} B_{b'} \sum_{\ell' \in L_{b'}} F_{\ell\ell'} f_{\ell'}^{b'} \\ &= \sum_{b'} B_{b'} \sum_{\ell \in L_b} \sum_{\ell' \in L_{b'}} F_{\ell\ell'} f_{\ell'}^{b'} \\ &= \sum_{b'} B_{b'} G_{bb'}, \end{aligned} \tag{4.5}$$

where in the last step we have defined the matrix $\mathbf{G}$. Defining $z_b \equiv \sum_{\ell \in L_b} y_\ell$, we get

$$\langle z_b \rangle = \sum_{b'} G_{bb'} B_{b'}, \tag{4.6}$$

or, arranging B_b and z_b, respectively, in the vectors $\mathbf{b}$ and $\mathbf{z}$, we can write the previous equation in matrix notation as

$$\langle \mathbf{z} \rangle = \mathbf{G}\mathbf{b}. \tag{4.7}$$

Note that the matrix $\mathbf{G}$ is square. Since we can reduce the number of variables as needed to make it regular, we can take as our estimator of the power in the bin

$$\widehat{\mathbf{b}} \equiv \mathbf{G}^{-1} \mathbf{z}, \tag{4.8}$$

that by eq. (4.7) and (4.8) is unbiased, i.e., $\langle \widehat{\mathbf{b}} \rangle = \mathbf{b}$.

Expressions (4.7) and (4.8) are analogous for the binned case to equations (2.45) and (2.50), respectively. It is important to note that the matrix $\mathbf{G}$ is not symmetric. Thus it is not the Fisher matrix expressed in terms of the variables B_b.

4.3 Covariance matrix for the binned estimator

To determine the covariance matrix of the binned estimator, let us define the rectangular matrix $\mathbf{A}$ of dimensions $N_{\text{bins}} \times (\ell_{max} - 1)$ as

$$\mathbf{A}_{b\ell} = \begin{cases} 1 & \text{if} \quad \ell \in L_b \\ 0 & \text{otherwise} \end{cases}, \tag{4.9}$$

such that for a given bin (row) it has non-null values only for those multipoles (columns) belonging to the considered bin b. Using this definition, the sum $z_b = \sum_{\ell \in L_b} y_\ell$ can be expressed as $\mathbf{z} = \mathbf{A}\mathbf{y}$. Therefore, combining this expression with eq. (2.45) and replacing the vector $\mathbf{c}$ by a vector $\mathbf{d}$ that contains the variables D_ℓ, we have

$$\langle \mathbf{z} \rangle = \mathbf{A}\langle \mathbf{y} \rangle = \mathbf{A}\mathbf{F}\mathbf{d}. \tag{4.10}$$

Let us also define a matrix $\mathbf{R}$ of dimensions $(\ell_{max} - 1) \times N_{\text{bins}}$ to transform the set $\{B_b\}$ into the set $\{D_\ell\}$

$$R_{\ell b} = \begin{cases} f_\ell^b & \text{if} \quad \ell \in L_b \\ 0 & \text{otherwise} \end{cases}, \tag{4.11}$$

so $\mathbf{d} = \mathbf{R}\mathbf{b}$. Expression (4.10) then becomes

$$\langle \mathbf{z} \rangle = \mathbf{A}\langle \mathbf{y} \rangle = \mathbf{A}\mathbf{F}\mathbf{R}\mathbf{b}. \tag{4.12}$$

The matrix $\mathbf{AFR}$ is a square matrix. By choosing an appropriate binning, this matrix is regular and therefore we can compute the estimator

$$\widehat{\mathbf{b}} \equiv [\mathbf{AFR}]^{-1}\mathbf{A}\mathbf{y}, \tag{4.13}$$

which is the same of eq. (4.8), and is unbiased. Finally, we can find the covariance matrix for the estimator

$$\begin{aligned} \langle (\widehat{\mathbf{b}} - \mathbf{b})(\widehat{\mathbf{b}} - \mathbf{b})^t \rangle &= [\mathbf{AFR}]^{-1}\mathbf{A}[\langle \mathbf{y}\mathbf{y}^t \rangle - \langle \mathbf{y} \rangle \langle \mathbf{y} \rangle^t]([\mathbf{AFR}]^{-1}\mathbf{A})^t \\ &= [\mathbf{A\,F\,R}]^{-1}\mathbf{AFA}^t[\mathbf{R}^t\,\mathbf{F}\,\mathbf{A}^t]^{-1}. \end{aligned} \tag{4.14}$$

4.4 Fisher matrix of the binned spectrum

In order to calculate the Fisher matrix corresponding to the B_b variables, let us first write the covariance matrix in terms of the binned power spectrum B_b and the factors f_ℓ^b

$$\mathbf{C} = \sum_\ell D_\ell \check{\mathbf{P}}_\ell + \mathbf{N} = \sum_b \sum_{\ell \in L_b} f_\ell^b B_b \check{\mathbf{P}}_\ell + \mathbf{N} = \sum_b B_b \sum_{\ell \in L_b} f_\ell^b \check{\mathbf{P}}_\ell + \mathbf{N}. \tag{4.15}$$

The Fisher matrix expressed in terms of the variables B_b is then given by[1]

$$\begin{aligned}\mathbf{F}^B_{bb'} &= \frac{1}{2}\operatorname{tr}\left[\mathbf{C}^{-1}\frac{\partial\mathbf{C}}{\partial B_b}\mathbf{C}^{-1}\frac{\partial\mathbf{C}}{\partial B_{b'}}\right] \\ &= \frac{1}{2}\operatorname{tr}\left[\mathbf{C}^{-1}\left(\sum_{\ell\in L_b} f^b_\ell \check{\mathbf{P}}_\ell\right)\mathbf{C}^{-1}\left(\sum_{\ell'\in L_{b'}} f^{b'}_{\ell'}\check{\mathbf{P}}_{\ell'}\right)\right] \\ &= \sum_{\ell\in L_b}\sum_{\ell'\in L_{b'}} f^b_\ell f^{b'}_{\ell'}\mathbf{F}_{\ell\ell'} = \mathbf{R}^t\mathbf{F}\mathbf{R}, \end{aligned} \tag{4.16}$$

where in the last step we have used the definition of matrix $\mathbf{R}$. To avoid confusion, it is important to note that the matrix $\mathbf{F}$ in the previous equation is the Fisher matrix corresponding to the power spectrum D_ℓ, while $\mathbf{F}^B$ is the Fisher matrix corresponding to the binned quantities B_b. For the covariance of the estimator to be minimum, the last term of (4.14) has to be equal to the inverse of $\mathbf{F}^B$, thus at first sight the estimator $\widehat{\mathbf{b}}$ does not have minimum variance.

4.5 Method of minimum variance

We recall that in the previous sections, we have simply added sets of linear equations eq. (4.5) to reduce the dimensionality of the problem. However, as we have seen, this leads to a non-optimal estimator. Therefore, we have to look for the appropriate way of combining the equations (i.e., to determine the appropriate $\mathbf{A}$) in order to obtain an estimator with minimum variance. In particular, if we choose $\mathbf{A} \equiv \mathbf{R}^t$, we find

$$\langle(\widehat{\mathbf{b}}-\mathbf{b})(\widehat{\mathbf{b}}-\mathbf{b})^t\rangle = [\mathbf{R}^t\mathbf{F}\mathbf{R}]^{-1}\mathbf{R}^t\mathbf{F}\mathbf{R}[\mathbf{R}^t\mathbf{F}\mathbf{R}]^{-1} = [\mathbf{R}^t\mathbf{F}\mathbf{R}]^{-1}, \tag{4.17}$$

where the last expression is the inverse of $\mathbf{F}^B$. Therefore, we find the optimal estimator for the binned case to be

$$\widehat{\mathbf{b}} = [\mathbf{R}^t\mathbf{F}\mathbf{R}]^{-1}\mathbf{R}^t\mathbf{y}. \tag{4.18}$$

In summary, to construct the optimal and unbiased binned version of QML, one first has to compute $\mathbf{F}$ and $\mathbf{y}$ as in the standard case. The next step is to define the set of bins as well as to obtain the value of the considered bandpowers from the full set of

---

[1] Since we are binning in terms of the variables D_ℓ, in the next expression we write explicitly the matrices $\mathbf{P}$ as $\check{\mathbf{P}}$. The Fisher matrix should also be written as $\check{\mathbf{F}}$ in that expression and the rest of the section, but we will keep the notation $\mathbf{F}$ for simplicity.

D_ℓ (usually as a weighted mean of the power spectrum in the bin). The binned QML will provide as output estimations of these bandpowers. Once this is established, one needs to compute accordingly the f_ℓ^b values using the information of the fiducial model, which will allow us to construct the **R** matrix (given by eq. (4.11)). Using this matrix, it is straightforward to estimate the binned power spectrum according to eq. (4.18). Finally, the Fisher matrix of the binned estimation is given by eq. (4.16).

4.6 Conclusions and limitations

This chapter shows that when the Fisher matrix becomes singular, by binning, the QML method can also be applied to get an optimal estimation of the power of the bins, but it does not show how to find the best binning scheme. In fact, as far as we know, there is not a simple answer to that question; this will depend on the specific characteristics of the data (sky coverage, noise, resolution) and also what the bins are going to be used for. A smaller number of bins will reduce the error bars in each of them and will reduce the correlation between bins, but at the price of possibly losing some specific features of the spectra because of the averaging. Therefore, this is an exercise that needs to be done carefully for each specific experiment and its objectives. In this work, we have tried to answer an equally important question: Given a fixed binning scheme (i.e., the number and size of bins are known), which is the best estimator of the binned power spectra? In particular, we have shown that our binned QML provides an unbiased and optimal estimation of the power of the bins. Moreover, this estimator provides a solution for the case in which the Fisher matrix is singular (as it can happen, for instance, for ground experiments with a small sky coverage), allowing to get an optimal estimation of binned power spectra even in this case. Once the Fisher matrix is computed, multiple binning schemes can be tried with very low additional computational cost.

On the other hand, given that the binned estimator is a BUE, one might wonder whether there is no loss of information when binning, which is counterintuitive. To analyze it, we would have to look at the expression (1.87) and replace the variables λ by the power spectrum D of this chapter and θ by the binned spectrum B. Using the expression (4.1) or the more general (4.3), we can compute the Jacobian of the transformation and, then, the Fisher matrix referred to the D variables from the Fisher matrix referred to the binning variables. So, apparently there is no loss of information. However, the binning —optimal or not— does not fulfill one of the prerequisites to arrive at the expression: we know that under the assumption that the CMB fluctuations are Gaussian all the information is contained in the power spectrum. But, even under the assumption of Gaussianity, the binned power spectrum variables no longer contain all the information. Therefore, they do not give a complete description of the probability distribution of the fluctuations. Hence, by binning we are losing information. This is another issue to take into account when choosing the binning scheme.

Chapter 5

Efficient implementation

5.1 Motivation

The QML method is computationally intensive. To implement it, it is necessary to compute numerous matrix operations, which can be time-consuming and memory-intensive when the size of the maps is large. Therefore, it may be useful to analyze the mathematical expressions in Chapter 2 to see whether they can be implemented in a way that reduces the number of operations to be performed and the memory requirements.

In this chapter, we show an approach to matrix operations that allows one to reduce very significantly the number of operations. Although the method will always remain computationally intensive, our formulation reduces significantly the computational costs associated to this technique, allowing one to estimate the power spectra up to higher multipoles than previous implementations. In particular, for a resolution of $N_{\text{side}} = 64$, $\ell_{\text{max}} = 192$ and a typical Galactic mask, the number of operations can be reduced by approximately a factor of 1000 in a full analysis including intensity and polarization with respect to an efficient direct implementation of the method.[1]

Once we have an optimal mathematical formulation, it is necessary to translate it into practice, embedded in a code that takes advantage of the power of today's supercomputers. Furthermore, it is very useful to make the code accessible to the entire scientific community and to provide it with a user's manual and some application examples. Our public code `ECLIPSE` (Efficient Cmb poLarization and Intensity Power

---

[1] We refer as "efficient direct implementation of the method" to an efficient implementation based on the equations in Chapter 2, i.e., an optimal implementation but carrying out the computations in the pixel space.

Spectra Estimator), an optimized implementation of the Quadratic Maximum Likelihood (QML) method,[2] implements a parallelized version of this implementation.

5.2 Different approaches and efficiency

A key element in an efficient implementation of the method is the connection between the pixel and harmonic domains, established by

$$\mathbf{x} = \mathbf{Y}\mathbf{a}, \tag{5.1}$$

where $\mathbf{x}$ are the data in the pixel domain, $\mathbf{a}$ are the data in the harmonic domain, and $\mathbf{Y}$ is a matrix that connects both (see Section 1.8.1). The vectors and matrices involved in this method can be constructed in any of these domains, and so the corresponding numerical implementation. Depending on which domain they are calculated, the time and memory required can be significantly different.

First, we will discuss the implementation in pixel space, which follows directly from the expressions given in Section 2.4. As we will see, some simple algebraic manipulation allows one to reduce the computational time in this case. However, the harmonic implementation, that we will outline below, is significantly more efficient.

A straightforward approach to implement the first step of the QML method, the calculation of the y_i vector, is to compute the matrices of eq. (2.2), (2.44) and (2.38) and, subsequently, the vectors of components b_i, $\mathbf{x}^t\mathbf{E}_i\mathbf{x}$ and y_i. However, it is easy to show that the quantity y_i can be obtained without the need of calculating explicitly the $\mathbf{E}_i$ matrices, reducing significantly the number of operations. In particular, from eqs. (2.11), (2.42) and (2.44), we can write

$$y_i = \frac{1}{2}\Big[(\mathbf{C}^{-1}x)^t\mathbf{P}_i(\mathbf{C}^{-1}x) - \mathrm{tr}((\mathbf{C}^{-1}\mathbf{N}\mathbf{C}^{-1})\mathbf{P}_i)\Big]. \tag{5.2}$$

For the second step of the method, we need to construct the Fisher matrix as well as its inverse. The calculation of the Fisher matrix is the part that involves the highest computational cost. From eq. (2.38), for each element of the matrix, one needs to compute the trace of a matrix product, having a total number of elements of the order of $(6\times(\ell_{\max}-1))^2/2$. Since the trace of the product of two matrices can be calculated without computing the product of the matrices (see eq. (5.15)), the number of operations can be considerably reduced, but one still has to compute —and keep stored in the memory of the computer— $6\times(\ell_{\max}-1)$ products of matrices of the kind $\mathbf{C}^{-1}\mathbf{P}_i$. If

---

[2] `ECLIPSE` is available at https://github.com/CosmoTool/ECLIPSE

the number of pixels in the map is large, this may require high computational resources regarding memory and CPU time (note that the number of operations to compute the product of two square matrices is of the order of the dimension of the matrices to the third power).

Alternatively, if we take into account in certain parts of the calculation the transformation from real to harmonic space given by eq. (5.1), it is possible to construct a significantly more efficient implementation of the QML, that we will refer as implementation in harmonic space, which is the base of our `ECLIPSE` code. The essence of the reduction in the number of operations is that, while in real space the $\mathbf{P}_i$ matrices are dense, their analogues in harmonic space are sparse, with a reduced number of ones in strategic locations. To take advantage of this property, we have performed a symbolic analysis of the results of the matrix operations involved and found analytical expressions of these results that can be easily implemented for a general case, reducing the number of operations tremendously. Moreover, the code can also be parallelized, further increasing the efficiency of the algorithm. We give the details of this approach in the next section and mention here only some of the improved aspects of the calculation.

First of all, it is possible to show that the y_i quantities can be computed avoiding the calculation of the $\mathbf{P}_i$ matrices. The quantities $\mathbf{x}^t\mathbf{E}_i\mathbf{x}$ can be obtained taking the vector product $\mathbf{Y}^\dagger(\mathbf{C}^{-1}\mathbf{x})$ and afterwards the sum of products of subsets of its elements. To get b_i one needs to compute the product $\mathbf{C}^{-1}\mathbf{Y}$ —a highly demanding operation that can not be avoided—. If the noise is spatially uncorrelated, the noise matrix is diagonal and the quantities b_i can be obtained through simple (non-matrix) operations with sub-blocks of elements of $\mathbf{C}^{-1}\mathbf{Y}$. Regarding the computation of the Fisher matrix, one only has to compute the product $\mathbf{Y}^\dagger(\mathbf{C}^{-1}\mathbf{Y})$ (actually only six square blocks out of the nine blocks of that matrix product). Once these blocks are calculated, the only work left in order to compute the Fisher matrix is collecting sub-blocks of the resultant blocks, multiply them and sum the elements of the product sub-block. This considerably reduces both the number of calculations —orders of magnitude— and the memory required to store the intermediate matrices. Therefore, the only highly demanding computer operations in our optimal implementation are the calculation of $\mathbf{C}$, $\mathbf{C}^{-1}$, $\mathbf{C}^{-1}\mathbf{Y}$ and six blocks of $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}$. The rest of operations consist in taking sub-blocks of the matrices, multiplying pairs of elements and summing the resulting numbers.

To quantify better the difference in the computational cost of the direct approach and the more efficient implementation of `ECLIPSE`, let us make an analysis of the number of operations involved in the calculation. For simplicity, we will focus only in the computation of the matrices needed to calculate the Fisher matrix, which is the most demanding step of the algorithm and will provide us with an approximated factor of the improvement gained with our approach.

In the straightforward implementation one has to compute the mentioned $\mathbf{C}^{-1}\mathbf{P}_i$ matrices multiplications. Since both matrices are of dimension $3N_{\text{pix}}$, the $6\times(\ell_{\text{max}}-1)$ multiplications $\mathbf{C}^{-1}\mathbf{P}_i$ require $6(\ell_{\text{max}}-1)(3N_{\text{pix}})^3$ operations. In fact, the number of operations would be even larger since, in this approach, one would need first to compute the $\mathbf{P}_i$ matrices.

In our efficient implementation one has to compute instead $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}$. The matrix $\mathbf{Y}$ has $3N_{\text{pix}}$ rows and $3L$ columns, where $L=\sum_{\ell=2}^{\ell_{\text{max}}}(2\ell+1)$. Therefore, the product $\mathbf{C}^{-1}\mathbf{Y}$ requires $3N_{\text{pix}}\times 3N_{\text{pix}}\times 3L$ operations. A complete computation of the last product, $\mathbf{Y}^\dagger(\mathbf{C}^{-1}\mathbf{Y})$, takes $3L\times 3N_{\text{pix}}\times 3L$ operations. Therefore the total number of operations is the sum, i.e., $27(N_{\text{pix}}^2L+N_{\text{pix}}L^2)$. In practice, the number of operations can be further reduced taking into account the structure of $\mathbf{Y}$ (see eq. (1.32)) and the fact that one needs to compute only six out of the nine blocks of this product. This takes the number of operations needed to compute the product $\mathbf{C}^{-1}\mathbf{Y}$ down to $15N_{\text{pix}}^2L$ while to compute $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}$ one needs in this case $18N_{\text{pix}}L^2$ operations. These expressions are summarized in table 5.1.

The specific number of required operations will depend on the values taken by the relevant parameters (resolution, number of pixels, maximum multipole) for the considered case. To have a better insight on how the different implementations scale with these parameters, let us consider the case $\ell_{\text{max}}=3N_{\text{side}}$ and $N_{\text{pix}}=12N_{\text{side}}^2$ (i.e., full sky), such that the number of operations depend only on N_{side} through a polynomial expression. It is straightforward to show that the leading term of the direct implementation scales as N_{side}^7 while the optimal implementation goes as N_{side}^6. Moreover, this term is multiplied by a different factor that increases even further the number of operations in the direct approach. This leading term including the specific factor is given in table 5.1. For comparison, an implementation in which only polarization is computed is also shown (this implementation is described in detail in Section 6.3).

The table also shows the number of operations in the case of a map at N_{side}=64, $\ell_{\text{max}}=192$ and a Galactic mask allowing for the use of $N_{\text{pix}}=29009$ pixels, corresponding to the space Configuration, for the direct and efficient approaches. As seen in the table, in order to compute the matrices needed to calculate the Fisher matrix, our efficient implementation requires around three orders of magnitude fewer operations than the straightforward approach for the considered case.

We can compare the performance of our code with those presented in the literature. Ref. [94] reports that it took roughly one day, using 16384 cores, to estimate the

	# operations (generic)	# operations $\ell_{\max} = 192$ $N_{\text{pix}} = 29009$	Ratio	# operations (leading term) $\ell_{\max} = 3N_{\text{side}}$ $N_{\text{pix}} = 12N_{\text{side}}^2$
Direct	$6(\ell_{\max}-1)(3N_{\text{pix}})^3$	7.55×10^{17}	634	$839808N_{\text{side}}^7$
Efficient TEB	$15N_{\text{pix}}^2 L + 18N_{\text{pix}} L^2$	1.19×10^{15}	1	$36936N_{\text{side}}^6$
Efficient EB	$8N_{\text{pix}}^2 L + 6N_{\text{pix}} L^2$	4.92×10^{14}	0.41	$16200N_{\text{side}}^6$

Table 5.1: Number of operations to compute the matrices needed to calculate the Fisher matrix in three different approaches: the direct estimation, our efficient implementation `ECLIPSE` and a case in which only polarization is computed also efficiently (see Section 6.3). The second column corresponds to a generic case while the third one refers to the particular case with $N_{\text{pix}} = 29009$ and $\ell_{\max} = 192$. The fourth column gives the ratio between the number of operations of the third column relative to our efficient approach for (T,E,B). Note that $L = 37245$ for the considered case. Finally, the last column shows the leading term of the polynomial expression that defines the number of operations in terms of the parameter N_{side} for the case $\ell_{\max} = 3N_{\text{side}}$ and full sky.

intensity power spectrum of 1000 maps at resolution $N_{\text{side}} = 64$ with a mask excluding around 20 per cent of the sky. In our case, we computed the six polarization power spectra up to $\ell_{\max} = 191$ of 1000 simulated maps at the same resolution and sky fraction in 90 minutes using 144 cores with a parallelized implementation in the Altamira[3] supercomputer at the Instituto de Física de Cantabria (IFCA). In [53] an efficient implementation of the QML method also based in the harmonic space is described, that computes the elements of the Fisher matrix as the trace of the product of two matrices of the kind $(\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y})\mathbf{I}_i$, where $\mathbf{I}_i$ represents a sparse matrix. Although this approach is similar to ours, our implementation takes advantage of the specific details of the previously mentioned traces and products of matrices, which consequently reduces very significantly the number of operations, allowing also a parallelization of the code. In particular, the implementation from [53] can compute an estimation of the polarization power spectra of a map at resolution $N_{\text{side}} = 16$ and $\ell_{\max} = 32$ in 2 CPU minutes. Our parallelized code can perform a similar computation in 7 seconds in a node of the Altamira supercomputer (16 cores), and it takes 24 seconds running on a single core of a laptop. Also note that, since our code is parallelized, it can compute problems of higher dimensions. We would like to emphasize that, as far as we know, `ECLIPSE` is the fastest available implementation of the QML estimator.

---

[3] https://www.res.es/en/res-sites/altamira

5.3 The efficient QML implementation

An implementation of the QML method requires to compute $\mathbf{C}$, its inverse, $\mathbf{x}^t\mathbf{E}_i\mathbf{x}$, b_i and the Fisher matrix $\mathbf{F}_{ii'}$. We calculate these quantities sequentially as shown below.

Computing C and C^{-1}

We recall that $\mathbf{C}$ is the sum of the signal and noise covariance matrices, of dimension $3N_{\rm pix}{\times}3N_{\rm pix}$. As shown in Section 1.8.1, the signal covariance matrix $\mathbf{S}$ can be efficiently computed using eq. (1.33). In particular, $\mathbf{S}$ has the structure

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_{TT} & \mathbf{S}_{TQ} & \mathbf{S}_{TU} \\ \mathbf{S}_{QT} & \mathbf{S}_{QQ} & \mathbf{S}_{QU} \\ \mathbf{S}_{UT} & \mathbf{S}_{UQ} & \mathbf{S}_{UU} \end{pmatrix}. \tag{5.3}$$

From eq. (1.33) and assuming $C_\ell^{TB} = C_\ell^{EB} = 0$, six of the blocks are given by

$$\begin{aligned}
\mathbf{S}_{TT} &= \mathbf{Y}_{TT}\tilde{\mathbf{S}}_{TT}\mathbf{Y}^\dagger_{TT} \\
\mathbf{S}_{TQ} &= \mathbf{Y}_{TT}\tilde{\mathbf{S}}_{TE}\mathbf{Y}^\dagger_{EQ} \\
\mathbf{S}_{TU} &= \mathbf{Y}_{TT}\tilde{\mathbf{S}}_{TE}\mathbf{Y}^\dagger_{EU} \\
\mathbf{S}_{QQ} &= \mathbf{Y}_{QE}\tilde{\mathbf{S}}_{EE}\mathbf{Y}^\dagger_{EQ} + \mathbf{Y}_{QB}\tilde{\mathbf{S}}_{BB}\mathbf{Y}^\dagger_{BQ} \\
\mathbf{S}_{QU} &= \mathbf{Y}_{QE}\tilde{\mathbf{S}}_{EE}\mathbf{Y}^\dagger_{EU} + \mathbf{Y}_{QB}\tilde{\mathbf{S}}_{BB}\mathbf{Y}^\dagger_{BU} \\
\mathbf{S}_{UU} &= \mathbf{Y}_{UE}\tilde{\mathbf{S}}_{EE}\mathbf{Y}^\dagger_{EU} + \mathbf{Y}_{UB}\tilde{\mathbf{S}}_{BB}\mathbf{Y}^\dagger_{BU},
\end{aligned} \tag{5.4}$$

where the matrices $\tilde{\mathbf{S}}_{XY}$ are the diagonal blocks of matrix $\tilde{\mathbf{S}}$, and the remaining blocks $\mathbf{S}_{QT}$, $\mathbf{S}_{UT}$ and $\mathbf{S}_{UQ}$ are the transpose of their symmetric partners of $\mathbf{S}$. There are some properties of the mathematical elements of eq. (5.4) that allow to reduce the number of computations. Since the matrices $\tilde{\mathbf{S}}_{XY}$ are diagonal, the products $\tilde{\mathbf{S}}_{XY}\mathbf{Y}^\dagger_{ZY}$ can be computed quickly without resorting to matrix multiplication. In addition, given that the elements of the covariance matrix are real numbers, one can reduce the number of calculations of eq. (5.4) by considering only those terms that produce real numbers (since the imaginary part will cancel). Once that the covariance matrix of the signal is computed, the covariance matrix of the noise has to be added.

The next step in the code is to compute the inverse of $\mathbf{C}$. We accomplish this step using efficient routines for symmetric definite positive matrices of the `ScaLAPACK` library [95].

Computing $\mathbf{x}^t\mathbf{E}_i\mathbf{x}$

As mentioned in Section 5.2, we can obtain $\mathbf{x}^t\mathbf{E}_i\mathbf{x}$ avoiding the direct calculation of the matrix $\mathbf{E}_i$ by computing instead the product $\mathbf{x}^t\mathbf{C}^{-1}\mathbf{P}_i\mathbf{C}^{-1}\mathbf{x}$. In particular, the matrices $\mathbf{P}_i$ can be computed from eq. (1.33) introducing a matrix basis $\tilde{\mathbf{P}}_i$ (more details in Section 1.8.1) such that

$$\tilde{\mathbf{S}} = \sum_i C_i \tilde{\mathbf{P}}_i. \tag{5.5}$$

The $\tilde{\mathbf{P}}_i$ are matrices of dimension $3L \times 3L$ with a structure of nine blocks (corresponding to the different combinations for the spectra of T,E,B) and constituted mostly by zeroes except for some selected elements of value 1 at the adequate positions. In particular, for an i index corresponding to a case of auto-spectra and to a multipole ℓ, we have $2\ell + 1$ non-null elements in the diagonal of the corresponding block, which are related to the m-elements of the considered multipole. For the case of cross-spectra, there are again $2\ell+1$ non-null elements but in two of the blocks corresponding to the considered cross-spectrum.

To illustrate better the structure of these matrices, let us consider as a toy model a map with signal only at multipoles $\ell = 0$ and $\ell = 1$.[4] In this case, the index i of eq. (5.5) runs from 1 to 12, corresponding to $\ell = 0, 1$ for each of the 6 possible power spectra (TT, EE, BB, TE, TB and EB). The ordering of the indices is such that i=(1, 2, 3, ..., 12) $\rightarrow$ ($\ell = 0$, TT; $\ell = 1$, TT; $\ell = 0$, EE; ... ; $\ell = 1$, EB).[5] As an example the $\tilde{\mathbf{P}}_i$ matrix for $i = 8$, corresponding to $\ell = 1$ and the TE cross-spectrum is given by

---

[4] Although in a realistic case $\ell_{min} = 2$ is usually assumed (in fact, the polarization signal is not even defined for $\ell < 2$), for the sake of simplicity, we will present some examples assuming that those multipoles actually exist in order to illustrate the calculations in a simple case. Whenever necessary, we will also indicate general results for the case $\ell_{min} = 2$.

[5] To avoid confusion, we remark that, just by chance, in our simple example, the $\tilde{\mathbf{P}}_i$ matrices have dimension 12×12 (since this is fixed by the quantity $3L$ and we have L=4) and there are also 12 values of the i index, which is given by the number of considered multipoles times the number of different power spectra (i.e., $2 \times 6 = 12$). However, in a general case, these two numbers will be different.

$$\tilde{\mathbf{P}}_8 = \left(\begin{array}{cccc|cccc|cccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right). \tag{5.6}$$

Equation (5.5) is the analogous in the harmonic space to $\mathbf{S} = \sum_i C_i \mathbf{P}_i$ in the pixel space. Combining eqs. (5.5) and (1.33), we have

$$\mathbf{S} = \mathbf{Y}\tilde{\mathbf{S}}\mathbf{Y}^\dagger = \mathbf{Y}\left(\sum_i C_i \tilde{\mathbf{P}}_i\right)\mathbf{Y}^\dagger = \sum_i C_i \mathbf{Y}\tilde{\mathbf{P}}_i\mathbf{Y}^\dagger. \tag{5.7}$$

Where we can identify

$$\mathbf{P}_i = \mathbf{Y}\tilde{\mathbf{P}}_i\mathbf{Y}^\dagger. \tag{5.8}$$

Thus

$$\mathbf{x}^t\mathbf{C}^{-1}\mathbf{P}_i\mathbf{C}^{-1}\mathbf{x} = \mathbf{x}^t\mathbf{C}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_i\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{x}. \tag{5.9}$$

To get the value of the last expression, we compute first the vector $\mathbf{Y}^\dagger(\mathbf{C}^{-1}\mathbf{x})$. The last operation transforms information in the pixel space to information in harmonic space. The next step is to collect it adequately taking into account the effect of multiplying by the $\tilde{\mathbf{P}}_i$ matrices. When $\tilde{\mathbf{P}}_i$ is one of the matrices associated to the cases TT, EE and BB, the vector product $\tilde{\mathbf{P}}_i\mathbf{Y}^\dagger(\mathbf{C}^{-1}\mathbf{x})$ is a vector of zeroes, except the values of $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{x}$ at the positions of the 1's in the diagonal of $\tilde{\mathbf{P}}_i$. Therefore the product of eq. (5.9) is directly the sum of the square of the real part plus the square of the imaginary part of the values of $\mathbf{Y}^\dagger(\mathbf{C}^{-1}\mathbf{x})$ at the adequate positions, what allows for a fast computation of these elements.

When $\tilde{\mathbf{P}}_i$ is one of the matrices associated to the cases TE, TB and EB, the vector product $\tilde{\mathbf{P}}_i\mathbf{Y}^\dagger(\mathbf{C}^{-1}\mathbf{x})$ is also a vector of zeroes, except for some values of $\mathbf{Y}^\dagger(\mathbf{C}^{-1}\mathbf{x})$ moved to other positions (since the non-null values of $\tilde{\mathbf{P}}_i$ are not in the diagonal in these cases). To illustrate this, let us consider again the toy model of a map with signal only at multipoles $\ell = 0$ and $\ell = 1$. In this case the vector has 12 elements grouped in three blocks of four elements each block; one of T, one of E and another of B. Written as a row vector

$$\mathbf{Y}^{\dagger}(\mathbf{C}^{-1}\mathbf{x}) = (T_0, T_1, T_2, T_3, E_0, E_1, E_2, E_3, B_0, B_1, B_2, B_3), \tag{5.10}$$

where the sub-indices code the pairs (ℓ, m): $(\ell = 0, m = 0) \to 0, \ldots$ and $(\ell = 1, m = 1) \to 3$. If we wanted to obtain the value of eq. (5.9) for the cross-power TE and $\ell = 1$, the product $\tilde{\mathbf{P}}_8\mathbf{Y}^{\dagger}(\mathbf{C}^{-1}\mathbf{x})$ would be

$$\tilde{\mathbf{P}}_8\mathbf{Y}^{\dagger}(\mathbf{C}^{-1}\mathbf{x}) = (0, E_1, E_2, E_3, 0, T_1, T_2, T_3, 0, 0, 0, 0) \tag{5.11}$$

and, finally, to obtain the product of eq. (5.9) for the value of the index $i = 8$ all we have to do is

$$\mathbf{x}^t\mathbf{C}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_8\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{x} = \sum_{k=1}^{3}[T_k E_k^* + T_k^* E_k]. \tag{5.12}$$

This establishes a rule also for a fast computation of all the elements associated to the cross-power correlation and therefore the expression (5.9) can be efficiently computed.

The previous paragraphs show that by including the matrix $\mathbf{Y}$, we go from working with the matrices $\mathbf{P}_i$, which implied carry out very demanding computations, to operate with some kind of matrices of selection $\tilde{\mathbf{P}}_i$. What makes this approach more efficient is that, in practice, we do not need to calculate these matrices, and their action on vectors (on matrices in the next sections) is just to select (or select and move the position of) certain elements of the vector. Therefore, with our approach, we significantly simplify this part of the QML calculation.

Note that it is also possible to work with several maps at the same time. In this case, one just need to replace in $\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{x}$ the vector $\mathbf{x}$ by a matrix whose columns are the maps, transform the matrix-vector multiplications into matrix-matrix multiplications and adequately collect and combine the values of that matrix product.

Computing $\mathbf{b}_i$

The next step is to compute the noise contribution in harmonic space. To do this, we need to compute just one matrix multiplication, $\mathbf{C}^{-1}\mathbf{Y}$, that will also be used to compute the Fisher matrix. Introducing $\tilde{\mathbf{P}}_i$ in eq. (2.11), we get

$$\mathbf{b}_i = \frac{1}{2}\,\mathrm{tr}\left[\mathbf{N}\mathbf{C}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_i\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\right]. \tag{5.13}$$

Let us show how to compute this quantity step by step. The matrix product $\mathbf{C}^{-1}\mathbf{Y}$ has as many rows as the vector map (i.e., $3N_{\mathrm{pix}}$) and as many columns as the matrix $\mathbf{Y}$ (i.e., 3L). For instance, for our toy model with $\ell \in \{0,1\}$ and only two pixels in the map, this is a matrix of six rows and twelve columns of structure

$$\mathbf{C}^{-1}\mathbf{Y} = \begin{pmatrix} TT_{10} & TT_{11} & \dots & TB_{13} \\ TT_{20} & TT_{21} & \dots & TB_{23} \\ QT_{10} & QT_{11} & \dots & QB_{13} \\ QT_{20} & QT_{21} & \dots & QB_{23} \\ UT_{10} & UT_{11} & \dots & UB_{13} \\ UT_{20} & UT_{21} & \dots & UB_{23} \end{pmatrix}, \tag{5.14}$$

where, for example, QB_{23} means that this element corresponds to a row associated to the second pixel of the Q component and to the $a_{X,\ell m}$ with $X = E$, $\ell = 1$ and $m = 1$.

In our model, the noise is assumed to be isotropic and uncorrelated, thus the noise matrix $\mathbf{N}$ is diagonal with values σ_T^2 and $\sigma_Q^2 = \sigma_U^2$. In this case, the matrix obtained by the operation $\mathbf{N}\mathbf{C}^{-1}\mathbf{Y}$ is simply the matrix given in eq. (5.14) with each j row multiplied by the diagonal j element of the noise matrix. Note that this is also valid if the noise is anisotropic and uncorrelated, since in this case the noise matrix is also diagonal.

In order to calculate the trace of eq. (5.13), let us recall that the trace of the product of two matrices $\mathbf{A}$ and $\mathbf{B}$ is

$$\operatorname{tr}\mathbf{AB} = \sum_{ij} \mathbf{A}_{ij}\mathbf{B}_{ji}, \tag{5.15}$$

i.e., we can obtain the trace of the product of a matrix without calculating the actual matrix multiplication. In this way, the trace can be written as

$$\mathbf{b}_i = \frac{1}{2}\sum_{\alpha\beta} (\mathbf{N}\mathbf{C}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_i)_{\alpha\beta} (\mathbf{C}^{-1}\mathbf{Y})^*_{\alpha\beta}. \tag{5.16}$$

Given that the matrices $\tilde{\mathbf{P}}_i$ are sparse, the first matrix product in the previous equation has a large number of null elements what, as we will see, simplifies the calculation of $\mathbf{b}_i$.

Let us show one example, again for our toy model with $\ell = \{0,1\}$ and two pixels, calculated for $i = 2$ (what corresponds to $\ell = 1$ and the component TT). In this case, the product $\mathbf{N}\mathbf{C}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_2$ has dimensions 6×12, and has the structure given by

$$\mathbf{NC}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_2 = \begin{pmatrix} 0 & \sigma_T^2 TT_{11} & \sigma_T^2 TT_{12} & \sigma_T^2 TT_{13} & 0 & \ldots & 0 \\ 0 & \sigma_T^2 TT_{21} & \sigma_T^2 TT_{22} & \sigma_T^2 TT_{23} & 0 & \ldots & 0 \\ 0 & \sigma_T^2 QT_{11} & \sigma_Q^2 QT_{12} & \sigma_Q^2 QT_{13} & 0 & \ldots & 0 \\ 0 & \sigma_T^2 QT_{21} & \sigma_Q^2 QT_{22} & \sigma_Q^2 QT_{23} & 0 & \ldots & 0 \\ 0 & \sigma_T^2 UT_{11} & \sigma_U^2 UT_{12} & \sigma_U^2 UT_{13} & 0 & \ldots & 0 \\ 0 & \sigma_T^2 UT_{21} & \sigma_U^2 UT_{22} & \sigma_U^2 UT_{23} & 0 & \ldots & 0 \end{pmatrix}, \tag{5.17}$$

where the first index indicates the pixel number and the second index runs over the different (ℓ, m) pairs. Therefore, this matrix has $2\ell+1$ columns with non-null elements. Taking this into account, to compute the required trace, we only need to multiply element by element, the second, third and fourth columns of the matrix of eq. (5.17) by the complex conjugates of the elements of the second, third and fourth columns of $\mathbf{C}^{-1}\mathbf{Y}$, and then sum the results.

From the previous calculations, it becomes apparent that the noise bias of y_ℓ^{TT} depends explicitly not only on the temperature noise but also on the noise of the polarization components. This is not surprising because the y_i quantities are a combination of the different power spectra, and not the power spectrum itself. So the noise bias is also a combination of the noise of the different components.

When computing the terms b_i associated to the cross power components, the effect of multiplying by matrices $\tilde{\mathbf{P}}_i$ is to select and reorder columns of $\mathbf{NC}^{-1}\mathbf{Y}$. For example, for $i = 7$ in our simple model, which corresponds to $\ell = 0$ and the TE component, we have

$$\mathbf{NC}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_7 = \begin{pmatrix} \sigma_T^2 TE_{10} & 0 & 0 & 0 & \sigma_T^2 TT_{10} & 0 & \ldots & 0 \\ \sigma_T^2 TE_{20} & 0 & 0 & 0 & \sigma_T^2 TT_{20} & 0 & \ldots & 0 \\ \sigma_Q^2 QE_{10} & 0 & 0 & 0 & \sigma_Q^2 QT_{10} & 0 & \ldots & 0 \\ \sigma_Q^2 QE_{20} & 0 & 0 & 0 & \sigma_Q^2 QT_{20} & 0 & \ldots & 0 \\ \sigma_U^2 UE_{10} & 0 & 0 & 0 & \sigma_U^2 UT_{10} & 0 & \ldots & 0 \\ \sigma_U^2 UE_{20} & 0 & 0 & 0 & \sigma_U^2 UT_{20} & 0 & \ldots & 0 \end{pmatrix}. \tag{5.18}$$

In this case, the number of columns with non-null elements is $2(2\ell+1)$.

Therefore, looking at eqs. (5.14)–(5.17), we can infer that to compute an element $\mathbf{b}_i$ corresponding to a given multipole ℓ and an auto spectra XX component, we have to multiply the elements of the appropriate $2\ell+1$ columns of the matrix of eq. (5.14) by their complex conjugates. The next step would be to multiply these terms by the appropriate noise variance but, since this quantity is the same for all the elements of a given row, it is more convenient to construct first a vector with one column, where each element is given by the sum of the $2\ell+1$ products corresponding to a fixed row. We then calculate the dot product of this vector by the vector constructed with the diagonal of the noise matrix $\mathbf{N}$ and, finally, the result is divided by two. A similar procedure can be inferred to calculate $\mathbf{b}_i$ for the cross-spectra components ($X \neq Y$) taking into account eq. (5.18).

Thus, in this section we have shown that the only demanding operation needed to obtain $\mathbf{b}_i$ is the matrix multiplication $\mathbf{C}^{-1}\mathbf{Y}$. Its computational cost can be reduced by taking into account that, as shown in eq. (1.32), $\mathbf{Y}$ is a matrix constituted by two diagonal blocks. Thus a good strategy to calculate this product is to split the matrix $\mathbf{C}^{-1}$ in four blocks that fit to the structure of $\mathbf{Y}$ and to compute four multiplications of matrices of smaller dimensions.

Computing the Fisher matrix

Writing eq. (2.38) in terms of $\mathbf{Y}$ and the matrices $\tilde{\mathbf{P}}$ and taking into account that the trace is invariant under cyclic permutations, we have

$$\mathbf{F}_{ii'} = \frac{1}{2}\,\mathrm{tr}\left[\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_i\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_{i'}\right]. \tag{5.19}$$

Since we have already computed $\mathbf{C}^{-1}\mathbf{Y}$, the next step is to multiply this matrix by $\mathbf{Y}^{\dagger}$, another strong consuming operation. The product $\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y}$ is an Hermitian matrix[6] structured on nine squared blocks of length L of the form

$$\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y} = \begin{pmatrix} TT & TE & TB \\ ET & EE & EB \\ BT & BE & BB \end{pmatrix}. \tag{5.20}$$

Let us define a notation to refer to the elements of the last matrix. For example, in the framework of our toy model, the element TE_{02} is located in the TE block in row number 1, corresponding to the harmonic of index 0 ($\ell = 0$ and $m = 0$) and in column number 3 of the block, corresponding to the harmonic of index 2 ($\ell = 1$ and $m = 0$). Note that this corresponds to the element in row number 1 and column number $L+3$ of the full matrix.

According to eq. (2.38), each element of the Fisher matrix involves a pair of multipoles. Since the power spectrum is composed of the six modes TT, EE, BB, TE, TB and

---

[6] Note that in practice this reduces the computational time, since only six out of the nine blocks needs to be computed.

EB, **F** is a symmetric matrix of $6^2(\ell_{\max}-1)^2$ elements.[7] For convenience, let us arrange them in a matrix conformed by 36 squared blocks of $(\ell_{\max}-1)^2$ elements each one

$$\mathbf{F} = \begin{pmatrix} TTTT & TTEE & TTBB & TTTE & TTTB & TTEB \\ EETT & EEEE & EEBB & EETE & EETB & EEEB \\ BBTT & BBEE & BBBB & BBTE & BBTB & BBEB \\ TETT & TEEE & TEBB & TETE & TETB & TEEB \\ TBTT & TBEE & TBBB & TBTE & TBTB & TBEB \\ EBTT & EBEE & EBBB & EBTE & EBTB & EBEB \end{pmatrix}, \tag{5.21}$$

where, for example, the block TTTE contains the cross terms between the multipoles C_ℓ^{TT} and $C_{\ell'}^{TE}$.

Taking into account eq. (5.15), we can obtain the Fisher matrix without actually calculating the direct matrix multiplication of eq. (5.19). Moreover, once the product $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}$ is computed, to obtain a given element of the Fisher matrix $\mathbf{F}_{ii'}$, we just have to localize the values and positions of the non null elements of $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_i$ and of the transpose of $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_{i'}$ (characterized by the positions of the 1's in $\tilde{\mathbf{P}}_i$ and $\tilde{\mathbf{P}}_{i'}$), multiply element by element those pairs formed by two non null elements and sum the result. In the following, we will show in detail how this technique leads to a fast method to compute **F**.

Let us show the procedure by showing how to calculate the elements of the first block (TTTT) of the Fisher matrix. For example, let us consider the element $\mathbf{F}_{i=1,i'=2}$ corresponding to $C_1 = C_0^{TT}$ and $C_2 = C_1^{TT}$ of eq. (2.38). In this case, the matrix $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_0^{TT}$, that we will call M0, reads

$$\mathrm{M0} \equiv \begin{pmatrix} TT_{00} & 0 & \dots & 0 \\ TT_{10} & 0 & \dots & 0 \\ TT_{20} & 0 & \dots & 0 \\ TT_{30} & 0 & \dots & 0 \\ ET_{00} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ BT_{30} & 0 & \dots & 0 \end{pmatrix}. \tag{5.22}$$

---

[7] For this number we are assuming $\ell_{min} = 2$ and, therefore, we have $(\ell_{\max}-1)$ different multipoles.

Analogously, the product $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_1^{TT}$ is given by

$$\mathrm{M1} \equiv \begin{pmatrix} 0 & TT_{01} & TT_{02} & TT_{03} & 0 & 0 & \dots & 0 \\ 0 & TT_{11} & TT_{12} & TT_{13} & 0 & 0 & \dots & 0 \\ 0 & TT_{21} & TT_{22} & TT_{23} & 0 & 0 & \dots & 0 \\ 0 & TT_{31} & TT_{32} & TT_{33} & 0 & 0 & \dots & 0 \\ 0 & ET_{01} & ET_{02} & ET_{03} & 0 & 0 & \dots & 0 \\ 0 & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & BT_{31} & BT_{32} & BT_{33} & 0 & 0 & \dots & 0 \end{pmatrix}. \tag{5.23}$$

The next step is to calculate the matrix whose elements are the multiplication of the elements of M0 and the transpose of M1

$$\mathrm{R01} \equiv \begin{pmatrix} 0 & 0 & \dots & 0 \\ TT_{10}TT_{01} & 0 & \dots & 0 \\ TT_{20}TT_{02} & 0 & \dots & 0 \\ TT_{30}TT_{03} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}. \tag{5.24}$$

The sum of the elements of R01 is twice the element $\mathbf{F}_{12}$ of the Fisher matrix.

A more complete analysis shows that, in fact, it is possible to compute, almost simultaneously, all the elements of each one of the 36 blocks of eq. (5.21) by means of adequate sums of the product of elements of certain blocks of $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}$ by elements of the transpose of certain blocks of the matrix $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}$. To illustrate this, let us show the matrices analogous to R01 of eq. (5.24) that emerge when computing with this technique another elements of the block TTTT of the Fisher matrix. For example, for $\ell = 0$ and $\ell' = 0$, R00 reads

$$\mathrm{R00} = \begin{pmatrix} TT_{00}TT_{00} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 0 \end{pmatrix}, \tag{5.25}$$

while for $\ell = 1$ and $\ell' = 0$, we have

$$\mathrm{R10} \equiv \begin{pmatrix} 0 & TT_{01}TT_{10} & TT_{02}TT_{20} & TT_{03}TT_{30} & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}, \tag{5.26}$$

and for $\ell = 1$ and $\ell' = 1$, R11 is

$$\text{R11} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & TT_{11}TT_{11} & TT_{12}TT_{21} & TT_{13}TT_{31} & 0 & \dots \\ 0 & TT_{21}TT_{12} & TT_{22}TT_{22} & TT_{23}TT_{32} & 0 & \dots \\ 0 & TT_{31}TT_{13} & TT_{32}TT_{23} & TT_{33}TT_{33} & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \tag{5.27}$$

In each case, the different elements of the first block (TTTT) of the Fisher matrix can be computed as the sum of the elements of the corresponding R matrices, divided by two. Moreover, one can infer that these elements of the Fisher matrix can also be computed by calculating the product, element by element, of the block TT of $\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y}$ times its transpose (which would give rise to a dense matrix filled by the non-null elements of eq. (5.24 – 5.27)) and then sum over the appropriate elements, selected taking into account the value of ℓ and ℓ', divided by two. That is, the element of the Fisher matrix associated to C_{ℓ}^{TT} and $C_{\ell'}^{TT}$ is given by

$$\mathbf{F}_{\ell\ell'}^{TTTT} = \frac{1}{2}\sum_{k_\ell k_{\ell'}} TT_{k_{\ell'}k_\ell} TT_{k_\ell k_{\ell'}}, \tag{5.28}$$

where TT on the right hand side refers to the first block of $\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y}$, the index k_ℓ runs over $2\ell+1$ values and $k_{\ell'}$ over $2\ell'+1$. If we denote k as the index that runs from 0 to $L-1$ (L is the size of the TT block in eq. (5.20)), for $\ell_{min} = 2$, we have that $k = 0$ corresponds to the pair $(\ell, m) = (2, -2)$. It can be shown that, for a given ℓ, k_ℓ runs from $\ell^2 - 4$ to $2\ell + \ell^2 - 4$, what defines the elements of the block that must be selected to construct the corresponding Fisher element.

For convenience, let us define the operator {} to symbolize the computation of the block TTTT of the Fisher matrix eq. (5.21) for all the pairs (ℓ, ℓ') using the expression (5.28) as

$$\mathbf{TTTT} = \{TT, TT\}, \tag{5.29}$$

where the first block TT in $\{TT, TT\}$ enters as it is and the second one is transposed.

To compute the elements of the rest of the blocks of $\mathbf{F}$, eq. (5.21), one has to apply the same technique used for the case TTTT but varying the matrices $\tilde{\mathbf{P}}_i$. Following a similar procedure as before, one can infer that there are three different cases

1. The blocks of $\mathbf{F}$ where the two considered power spectra correspond to autocorrelations (TT, EE, BB), that is, the blocks TTTT, TTEE, TTBB, EEEE, EEBB, BBBB.

2. The blocks of **F** that mix one autocorrelation and one cross-correlation, i.e., TTTE, TTTB, TTEB, EETE, EETB, EEEB, BBTE, BBTE, BBBE.

3. The blocks that involve two cross-correlations, i.e., TETE, TETB, TEEB, TBTB, TBEB, EBEB.

For instance, if one derives the equations equivalent to (5.22) - (5.28) for the block TTEE (case i), it is found that one has to multiply element by element the block ET of eq. (5.20) and the transpose of the block TE, sum the sub blocks adequately, and divide by two. That is

$$\mathbf{TTEE} = \{ET, TE\}. \tag{5.30}$$

The rest of the blocks of the case (i) are obtained as

$$\begin{aligned}
\mathbf{TTBB} &= \{BT, TB\}, \\
\mathbf{EEEE} &= \{EE, EE\}, \\
\mathbf{EEBB} &= \{BE, EB\}, \\
\mathbf{BBBB} &= \{BB, BB\}.
\end{aligned} \tag{5.31}$$

Let us show an example to find the rule to compute the blocks of the case (ii), in particular, we will consider the element of $\ell = 0$ and $\ell' = 1$ of the block TTTE of our toy model, that is, the element $\mathbf{F}_{ii'}$ of $C_i = C_0^{TT}$ and $C_{i'} = C_1^{TE}$ of eq. (2.38). In this case, one has to compute $\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_0^{TT}$ and multiply element by element by the transpose of $\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y}\tilde{\mathbf{P}}_1^{TE}$. Given that the $\tilde{\mathbf{P}}_1^{TE}$ matrix has a total of six elements with the value of 1, grouped in two sets of three, outside the main diagonal (see eq. (5.6)),[8] this gives rise to two blocks of three non-null elements in R01.

---

[8] Note that in the notation used in this subsection $\tilde{\mathbf{P}}_1^{TE} \equiv \tilde{\mathbf{P}}_8$

In particular, the result reads

$$\mathrm{R01} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ TT_{10}TE_{01} & 0 & \dots & 0 \\ TT_{20}TE_{02} & 0 & \dots & 0 \\ TT_{30}TE_{03} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ ET_{10}TT_{01} & 0 & \dots & 0 \\ ET_{20}TT_{02} & 0 & \dots & 0 \\ ET_{30}TT_{03} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 0 \end{pmatrix}. \tag{5.32}$$

Therefore, following the same reasoning as in previous cases, one can write

$$\mathbf{TTTE} \quad = \quad \{TT,TE\} + \{ET,TT\}. \tag{5.33}$$

The imaginary part of $\{TT,TE\}+\{ET,TT\}$ is zero and the real part of $\{TT,TE\}$ is equal to the real part of $\{ET,TT\}$, therefore

$$\mathbf{TTTE} \quad = \quad 2\{TT,TE\}. \tag{5.34}$$

For the sake of clarity, let us write explicitly how to calculate an element of the Fisher matrix in the block TTTE

$$\mathbf{F}_{\ell\ell'}^{TTTE} = \frac{1}{2} \times 2 \sum_{k_\ell k_{\ell'}} TT_{k_{\ell'}k_\ell} TE_{k_\ell k_{\ell'}}. \tag{5.35}$$

Applying the technique to the rest of the blocks of the Fisher matrix of case (ii), we get

$$\begin{aligned} \mathbf{TTTB} &= \{TT,TB\} + \{BT,TT\} = 2\{TT,TB\}, \\ \mathbf{TTEB} &= \{ET,TB\} + \{BT,TE\} = 2\{ET,TB\}, \\ \mathbf{EETE} &= \{TE,EE\} + \{EE,ET\} = 2\{TE,EE\}, \\ \mathbf{EETB} &= \{TE,EB\} + \{BE,ET\} = 2\{TE,EB\}, \\ \mathbf{EEEB} &= \{EE,EB\} + \{BE,EE\} = 2\{EE,EB\}, \\ \mathbf{BBTE} &= \{TB,BE\} + \{EB,BT\} = 2\{TB,BE\}, \\ \mathbf{BBTB} &= \{TB,BB\} + \{BB,BT\} = 2\{TB,BB\}, \\ \mathbf{BBEB} &= \{EB,BB\} + \{BB,BE\} = 2\{EB,BB\}. \end{aligned} \tag{5.36}$$

Finally, for the blocks corresponding to case (iii), we find

$$\begin{aligned}
\mathbf{TETE} &= \{TE,TE\} + \{TT,EE\} + \{EE,TT\} + \{ET,ET\} \\
&= 2\{TE,TE\} + 2\{TT,EE\}, \\
\mathbf{TETB} &= \{TE,TB\} + \{TT,EB\} + \{BE,TT\} + \{BT,ET\} \\
&= 2\{TE,TB\} + 2\{TT,EB\}, \\
\mathbf{TEEB} &= \{EE,TB\} + \{ET,EB\} + \{BE,TE\} + \{BT,EE\} \\
&= 2\{EE,TB\} + 2\{ET,EB\}, \\
\mathbf{TBTB} &= \{TB,TB\} + \{TT,BB\} + \{BB,TT\} + \{BT,BT\} \\
&= 2\{TB,TB\} + 2\{TT,BB\}, \\
\mathbf{TBEB} &= \{EB,TB\} + \{ET,BB\} + \{BB,TE\} + \{BT,BE\} \\
&= 2\{EB,TB\} + 2\{ET,BB\}, \\
\mathbf{EBEB} &= \{EB,EB\} + \{EE,BB\} + \{BB,EE\} + \{BE,BE\} \\
&= 2\{EB,EB\} + 2\{EE,BB\}.
\end{aligned} \tag{5.37}$$

5.3.1 Discussion on efficiency

We have shown a way of implementing the computations involved in the application of the QML method that significantly reduces the number of operations. Let us add some details regarding the extent to which it does so when computing the most computationally intensive mathematical element in the whole method, the Fisher matrix. The elements of the matrix are

$$\mathbf{F}_{ii'} = \frac{1}{2}\,\mathrm{tr}\left[\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial C_i}\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial C_{i'}}\right]. \tag{5.38}$$

As shown in this chapter, and also used in [53], it is more efficient to compute it by making use of the matrices $\tilde{\mathbf{P}}_i$

$$\mathbf{F}_{ii'} = \frac{1}{2}\,\mathrm{tr}\left[(\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y})\tilde{\mathbf{P}}_i(\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y})\tilde{\mathbf{P}}_{i'}\right], \tag{5.39}$$

where i runs over the six components of the spectra and over multipoles $\ell = 2$ to $\ell_{\max}$.

Since the matrix is symmetric, our method starts by computing 6 out of the 9 blocks of the product $\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y}$. Once this is done, a naive implementation of the product of matrices would compute one by one each of the independent elements of the Fisher matrix by performing the large matrix multiplications of eq. (5.39). As an illustration, for $\ell_{\max} = 191$, there are a total of 650370 elements to be calculated, corresponding to the elements of the upper triangle of the Fisher matrix of size 6x190 (note that since the matrix is symmetric, the lower triangle does not have to be calculated). Since the $\tilde{\mathbf{P}}_i$

are sparse matrices, these operations are less demanding than those of a direct Fisher matrix computation (eq. 5.38) but still entail a large number of operations involving very large matrices.

In contrast, our method uses a different approach. As explained in this chapter, after computing the six blocks of the product $\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y}$, instead of treating each element of the Fisher matrix as a product of large matrices, we have developed an algorithm that allows us to compute each element of the Fisher matrix by multiplying certain elements of the different blocks and then summing them appropriately. This reduces the number of operations significantly with respect to an also naive implementation of the calculation of the Fisher matrix elements based on eq. (5.39) and products of sparse matrices. Moreover, in our method these calculations do not consist on matrix operations —for example, for $\ell_{\max} = 191$ we save 650370 matrices operations— and are well suited to be parallelized, as we have done in the `ECLIPSE` implementation.

5.4 Working with variables D_i, beam and pixel window

To implement the QML in terms of the variables D_i and to take into account the effect of the beam of the experiment and the pixel window, one just needs to define adequately the matrix $\mathbf{Y}$.

Writing an element of the matrix $\mathbf{S}$ according to the expression (1.33), we have

$$\mathbf{S}_{ij} = \sum_{kk'} \mathbf{Y}_{ik} \tilde{\mathbf{S}}_{kk'} \mathbf{Y}^{\dagger}_{k'j} = \sum_{kk'} \mathbf{Y}_{ik} C_{kk'} \mathbf{Y}^{\dagger}_{k'j}, \tag{5.40}$$

where, taking into account the structure of $\tilde{\mathbf{S}}$, $C_{kk'}$ is some of the C_ℓ^{TT}, C_ℓ^{EE}, C_ℓ^{BB}, when $k = k'$, while for $k \neq k'$ corresponds to C_ℓ^{TE} in some cases or to zero otherwise.

Introducing the beam and pixel window functions through B_k and the $D_{kk'}$ variables instead of $C_{kk'}$, we have

$$\mathbf{S}_{ij} = \sum_{kk'} \mathbf{Y}_{ik} \frac{2\pi B_k B_{k'}}{\ell(\ell+1)} D_{kk'} \mathbf{Y}^{\dagger}_{k'j}. \tag{5.41}$$

We can rewrite the previous equation as

$$\mathbf{S}_{ij} = \sum_{kk'} \left(\sqrt{\frac{2\pi}{\ell(\ell+1)}} B_k \mathbf{Y}_{ik} \right) D_{kk'} \left(\sqrt{\frac{2\pi}{\ell(\ell+1)}} B_{k'} \mathbf{Y}^{\dagger}_{k'j} \right). \tag{5.42}$$

From the last expression we can define a new matrix $\check{\mathbf{Y}}$ that satisfies

$$\mathbf{S}_{ij} = \sum_{kk'} \check{\mathbf{Y}}_{ik} D_{kk'} \check{\mathbf{Y}}^{\dagger}_{k'j}. \tag{5.43}$$

That is

$$\check{\mathbf{Y}}_{ik} = \sqrt{\frac{2\pi}{\ell(\ell+1)}} B_k \mathbf{Y}_{ik}. \tag{5.44}$$

Hence the columns of $\check{\mathbf{Y}}$ are just the columns of $\mathbf{Y}$ multiplied by the appropriate factor $\sqrt{\frac{2\pi}{\ell(\ell+1)}} B_k$.

5.5 The ECLIPSE implementation

The `ECLIPSE` code, written in `Fortran` language, implements the QML method and performs the mathematical operations as described in this chapter.

Matrix operations (multiplications, Cholesky decomposition and inversion) are performed efficiently through calls to `ScaLAPACK` library routines, so the code is designed to be executed in parallel. In its design, we have tried to make it efficient when operating with matrices by means of calls to routines of proven quality. In addition, we have tried to optimize the operations that act on the blocks of the matrices distributed by the grid of processors that have been described in this chapter.

Because it is designed as parallelized code, it is not only faster than standard code, but also allows the method to be applied to large-scale, memory-intensive problems, which with today's computers can only be accessed by distributed parallel computations on a network of processors. Thus, `ECLIPSE` is designed and optimized to be used on supercomputers, although it can also run on personal computers — if provided with the appropriate libraries.

Although the computational procedure is only one, three variants of `ECLIPSE` have been written, which differ in the data with which they work and on the spectra that they calculate. `ECLIPSE_TEB` operates on intensity and polarization maps, can use the same or different masks for intensity and polarization, and calculates the six components of the power spectrum. `ECLIPSE_EB` operates on polarization-only maps maps and computes the three polarization components of the power spectrum with the advantages that will be described in the Section 6.3, leading to a reduction in time and computational resources. The third version, `ECLIPSE_T`, operates on intensity-only maps and computes the TT component of the power spectrum. Unlike the other two variants, it computes the power spectrum starting from $\ell = 1$.

The code works with standard maps in cosmology, FITS files written by `HEALPix` routines. It can compute the full or binned power spectrum. When estimating it binned, one can apply the optimal binning estimator described in Chapter 4 or a non-optimal version that does not include information on the fiducial. Noise can be isotropic or anisotropic. Besides, it can compute auto-correlation spectra of individual maps and cross-correlation spectra of pairs of maps.

The **ECLIPSE User guide**, given in Appendix G, contains information on how to build the code and run it on personal computers and supercomputers, how it works, the data it reads and stores and its configuration, the configuration of the information that the program reads before loading data and performing calculations, and how to obtain estimates of the amount of memory needed for a given calculation, among other things.

Because of the optimized implementation, parallelization and the efficiency of the `Fortran` language, we believe that `ECLIPSE` will be very useful for the community, allowing to reach higher multipoles and to carry out, in practice, analyses that before were prohibitively slow. The code, manual and examples are available at https://github.com/CosmoTool/ECLIPSE.

The sequence and structure of operations

In this section we briefly describe the sequence and the form in which the `ECLIPSE` implementation performs the matrix operations.

Initially, the program loads the data specifying the type of calculation it has to perform, the location of the files with the data, and the number of cores being used and their distribution in the processor grid. Next, the program loads, among other data, the temperature mask, the polarization mask and the fiducial power spectrum.

Once the data has been loaded, the program starts with the calculations. These are performed in six steps that correspond to the sequence of matrix operations described in the previous sections of this chapter. The steps are:

1. The computation of the covariance matrix, $\mathbf{C}$.
2. The inversion of the covariance matrix, $\mathbf{C}^{-1}$
3. The computation of the coupled power in harmonic space, $\mathbf{x}^t\mathbf{E}_i\mathbf{x}$.
4. The computation of the product, $\mathbf{C}^{-1}\mathbf{Y}$.
5. The computation of the noise bias, b_i
6. The computation of the Fisher matrix.

In all steps we operate with blocks of matrices instead of complete matrices, so that we do not perform calculations that would be redundant, such as computing parts of matrices whose value we already know because they are symmetrical. This saves time and memory. Moreover, even when working at many places with complex numbers, part of the calculations are done by separating the real and imaginary parts, and operating with them to produce real numbers. By not operating with complete complex numbers, we avoid the computation of imaginary parts of blocks of matrices that we already know are null because they are made up of real numbers. This makes it possible to halve the number of operations in certain calculations. The messages displayed by `ECLIPSE` (see Chapter 8 of the User Guide) inform in detail about the operations it is performing at each moment, the blocks with which it is operating and the way in which it is treating the data.

After having performed the previous steps, which are the ones that involve a high computational workload, the program calculates the power spectrum, either binned or unbinned. If a binned spectrum is desired, it can calculate the optimal binned spectrum making use of the information in the fiducial (as described in Chapter 4) or it can calculate a simple binned spectrum.

While performing computations, the program displays the elapsed time and, if the user wishes, it displays the memory used by the processor grid to store the matrix blocks. The sequence of operations and memory allocation and deallocation for storing blocks is organized to reduce the memory required. The `ECLIPSE` code includes a small auxiliary program that allows to know in advance the memory needed to perform the matrix operations once the dimensions of the estimation to be performed are known (N_{pix} and ℓ_{max}). This makes it easier to decide the number of processors to request to perform the calculations.

There are certain stages in the sequence of calculations when the program performs validation tests. For example, it checks whether the matrices it has to invert (covariance, Fisher or compacted Fisher) are regular or whether the diagonal of $\mathbf{C}^{-1}\mathbf{C}$ is made of ones (for more details see the **ECLIPSE User guide** given in Appendix G).

5.6 Conclusions

The QML method is very computationally demanding and, therefore, can be applied only at limited resolution. In this chapter, we have developed a new implementation of the QML that reduces considerably the required computationally resources (CPU time and memory).

In its natural mathematical formulation, QML operations are performed in pixel space. By translating them to harmonic space, a significant reduction in the number of operations is achieved. The essence of the reduction lies in the fact that the covariance matrix of the signal in harmonic space is much simpler than in pixel space, because it becomes a sum of scalars multiplying sparse matrices with very few ones. By analytically analyzing in detail the effect of those few ones when multiplying by them, rules can be found that allow predicting the result of the computations leading to the essential quantities in QML, with consequent savings in the number of matrix operations and in the computational workload involved in each matrix multiplication. In addition, the rules for calculating the result of matrix operations can be implemented in parallel, so that each processor operates independently on the values of the matrix blocks that it stores in its memory.

The key element connecting harmonic and pixel spaces is the **Y** matrix. Since it consists of two diagonal blocks, all the operations in which it participates can be organized by blocks. In addition, all the numbers that QML ultimately produces are real numbers, so operations involving blocks of complex numbers can be organized so that only the real parts are computed.

This yields to a very significant reduction in the number of operations. For instance, we can compute the intensity and polarization power spectra of 1000 simulated maps of 39322 observed pixels at resolution $N_{\text{side}} = 64$ up to $\ell_{\text{max}} = 191$ in 90 minutes using 144 cores at the Altamira supercomputer, with a reduction in the number of operations around a factor 1000 with respect to the optimal direct implementation.

Our implementation, `ECLIPSE`, is published and freely available. It has already been used by the author or third parties for several works other than this Thesis (for more information, see Section 6.7).

CHAPTER 6

TEST ON THE PERFORMANCE OF QML

6.1 Motivation

This chapter is intended to illustrate the performance of the QML estimator. We present results for a space-based B-mode mission and for a typical ground-based experiment. The QML method is well-known, but there are several practical issues that should be taken into account when applying it to real data. In particular, depending on the observed sky fraction, the Fisher matrix can become singular. To solve this problem, we have found a binned version of the QML estimator to be used when the Fisher matrix is not invertible (see Chapter 4). Another important aspect of this estimator is that the user must provide an initial guess for the power spectra. However, we may wonder how the results are affected if this initial model differs from the true power spectra (see Chapter 2). We present several tests of robustness of QML versus the assumed fiducial model and show that starting from significantly different initial guesses, through an iterative scheme map by map, QML drives statistically to the optimal estimator. We also study the possibility of using the QML method to estimate only the polarization components with a consequent reduction of computational resources. A comparison with an advanced pseudo-C_ℓ algorithm (`NaMaster`, [59]) is also presented, showing the better efficiency of QML at large scales.

6.2 Instrumental configurations

In order to test the performance of the QML method in different situations, we have mainly considered two different experimental configurations along the text. Showing results for different cases is interesting since the very existence of the optimal estimator depends on the sky coverage of the experiment, while the noise level is reflected in the error of the estimated power spectra.

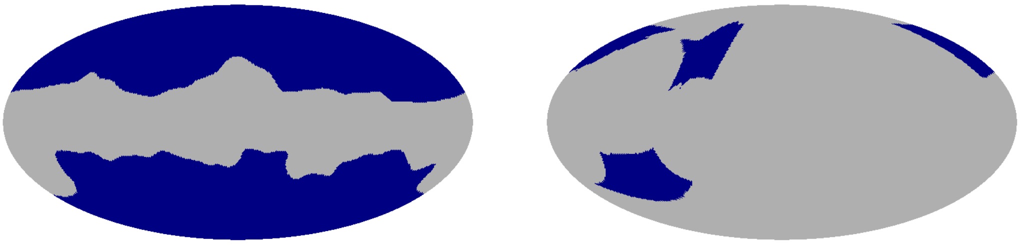


Figure 6.1: Sky coverage considered for the space (left) and ground-based (right) experiments, shown at a HEALPix resolution $N_{\text{side}} = 64$.

	f_{sky} (%)	Noise IQU ($\mu K \cdot$ arcmin)
Space	59.0	2.5
Ground	8.4	1.0

Table 6.1: Specifications of the instrumental configurations considered along this work to test the QML estimator. The sky fraction of each mask has been obtained at a HEALPix resolution of $N_{\text{side}} = 64$. A Gaussian beam of FWHM = 2.4 times the corresponding pixel size is adopted.

In particular, we have considered a spaceborne-like experiment, that provides a large sky coverage and a ground-based experiment that focuses on certain regions of the sky. As a typical example, we have considered the sky coverage given by the masks of figure 6.1 for the space (left panel) and ground-based (right panel) experiments. The first mask corresponds to the Galactic mask provided by *Planck*, that allows one to use the 60% of the sky at full resolution,[1] while the second one corresponds approximately to the three cosmological regions selected by the QUIJOTE experiment [96, 97]. Regarding noise sensitivities, for the space mission, we have considered a noise level of 2.5 μK · arcmin, similar to that expected for the JAXA LiteBIRD satellite [69], while for the ground experiment, a value of 1 μK · arcmin is assumed. This sensitivity is similar to what could be obtained by the US-led CMB-S4 experiment [68] or by a future Low Frequency Survey [98]. Table 6.1 summarizes the specifications of the two selected configurations.

For our tests, CMB simulations have been produced using the HEALPix package [39], for a standard cosmological model given by the best fit values of the cosmological

---

[1] More specifically, we have used the mask file HFI_Mask_GalPlane-apo0_2048_R2.00.fits available at the *Planck* Legacy Archive.

parameters of the baseline *Planck* 2018 ΛCDM model [21], but adding a tensor-to-scalar ratio $r = 3 \times 10^{-3}$. The corresponding power spectra have been obtained using CAMB [99]. Along the chapter, we will refer to this choice as *Planck* model and, except when otherwise stated, it will be the power spectra assumed in this work. A Gaussian beam of FWHM = 2.4 times the pixel size is also adopted.

6.3 Full and only-polarization implementations

The QML estimator has been usually implemented to compute either only the temperature power spectrum or all the six possible spectra (intensity and polarization) simultaneously. However, if we ignore the information about the correlation between temperature and polarization, it is also possible to implement the QML only for the three polarization spectra, i.e., EE, BB and EB. This is interesting since it implies an important reduction of the computational requirements, allowing one to work at higher resolution. Independently of the assumed fiducial model, the QML estimator is unbiased and, therefore, these different constructions of the QML should produce, on average, the same results. However, the estimator error is only optimal if we use the correct fiducial model in the definition of the covariance matrix.

An example in which this partial estimation of the CMB angular power spectra, focused on the polarization signal, could imply a clear benefit is related to instrumental calibrations of CMB experiments. In particular, the accurate estimation of the EB angular power spectrum can be used as a capital observable to perform the polarization angle calibration. This is recognized as one of the most important systematics to have under control for incoming high-sensitivity CMB polarization experiments. Mismatch calibrations of the polarization angle (above a few arcminutes) could induce a leakage from E-modes to B-modes that masks any possible primordial signal with $r \lesssim 10^{-3}$. This degree of accuracy can not be obtained from astrophysical sources and, up to date, nulling the observed EB angular power spectrum (as it would be expected from the standard ΛCDM model), is a clear approach to reach the required degree of accuracy on the polarization angle estimation, see, for instance, [100]. This observable is only useful for this purpose if the EB estimation is unbiased, with optimal error bars, and up to a large multipole value.

In this section, we study the performance of the QML both when recovering the complete power spectra and also when recovering only the polarization terms, checking that it provides unbiased results and quantifying the increase in the error of the estimator when neglecting the correlation between temperature and polarization.

From eq. (1.40), the structure of the covariance matrix in the temperature and polarization case is given by

$$\mathbf{C} = \begin{pmatrix} \text{Block}\begin{bmatrix} TT \end{bmatrix} & \text{Block}\begin{bmatrix} TQ & TU \end{bmatrix} \\ \\ \text{Block}\begin{bmatrix} QT \\ UT \end{bmatrix} & \text{Block}\begin{bmatrix} QQ & QU \\ UQ & UU \end{bmatrix} \end{pmatrix}. \tag{6.1}$$

The matrix is composed of a diagonal block that accounts exclusively for temperature correlations, another diagonal block that encodes polarization correlations and two off-diagonal blocks that mix temperature and polarization. These off-diagonal blocks are related to C_ℓ^{TB}, which is expected to vanish in the standard cosmological model, and to C_ℓ^{TE}. Therefore, if we impose $C_\ell^{TE} = 0$ in the fiducial model, the estimations for temperature and polarization are decoupled. Taking this into account, we have considered three different QML implementations:

1. $\mathcal{C}_{IP}$: estimation of intensity and polarization spectra using a complete covariance matrix of order $3N_{\text{pix}}$ with $C_\ell^{TE} \neq 0$ in the fiducial model.
2. $\mathcal{C}'_{IP}$: estimation of intensity and polarization spectra using a complete covariance matrix of order $3N_{\text{pix}}$ but with $C_\ell^{TE} = 0$ in the fiducial model, i.e., containing zero off-diagonal blocks.
3. $\mathcal{C}_P$: a reduced version operating only on Q and U. The covariance matrix is the second diagonal block of eq. (6.1), of order $2N_{\text{pix}}$. This version only estimates EE, BB and EB spectra.

Let us remark that, with regard to the TT, EE, BB and EB spectra, the implementation $\mathcal{C}'_{IP}$ is equivalent to run two independent QMLs, one for intensity and one for polarization since, when $C_\ell^{TE} = 0$ in the fiducial model, the calculations for the estimation of intensity and polarization spectra are decoupled. Of course, the $\mathcal{C}'_{IP}$ configuration allows also the estimation of the TE and TB spectra, that would not be obtained with two independent QML estimators. Therefore, the $\mathcal{C}'_{IP}$ and $\mathcal{C}_P$ configurations should provide exactly the same results for the EE, BB and EB spectra map by map, but different from those obtained with $\mathcal{C}_{IP}$. Also, $\mathcal{C}_{IP}$ and $\mathcal{C}'_{IP}$ provide, map by map, different estimations for TT, TE and TB.

It is clear that the lower dimensionality of the problem in the only polarization case implies a reduction of the required computational resources. Not only the covariance matrix has a smaller size, but also the number of elements in the Fisher matrix goes down from $6^2(\ell_{\max}-1)^2$ in the $\mathcal{C}_{IP}$ and $\mathcal{C}'_{IP}$ cases to $3^2(\ell_{\max}-1)^2$ for the $\mathcal{C}_P$ configuration. In particular, we can obtain a rough estimation of the reduction of the CPU time, by referring again to the computation of the matrix $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}$. In the $\mathcal{C}_P$ case, the product $\mathbf{C}^{-1}\mathbf{Y}$ requires $8N_{\text{pix}}^2 L$ operations while the computation of the blocks needed from $\mathbf{Y}^\dagger(\mathbf{C}^{-1}\mathbf{Y})$, only three blocks in this case, takes $6NL^2$; thus we have a total of $8N_{\text{pix}}^2 L + 6N_{\text{pix}}L^2$ operations. This leads to a reduction of approximately a factor 2.3 in the number of operations with respect to the full implementation of (T,E,B) for the case considered in table 5.1. Note that this factor is only mildly dependent on the values of $\ell_{\max}$ and N_{pix} and will range approximately between 2 and 3.

6.3.1 Comparison of full and only-polarization implementations

We have also compared the performance of the three QML estimators on a practical example, considering 5000 CMB simulated maps for the space configuration given in table 6.1, at resolution $N_{\text{side}} = 64$ and $\ell_{\max} = 128$. All the calculations were done in the Altamira supercomputer with 100 processors, with our efficient harmonic implementation, taking 32 and 18 minutes for the $\mathcal{C}_{IP}$ and $\mathcal{C}_P$ cases, respectively, which means a factor of improvement of around two. Although, given the high optimization of our code, this is not as large as one would obtain in the case of a direct implementation of the method, the $\mathcal{C}_P$ implementation still provides a significant reduction of computational time that, together with the smaller memory requirements, can be interesting if one wants to go to the highest possible resolution.

Figure 6.2 shows the TE power spectrum (right panel) as derived with the $\mathcal{C}_{IP}$ and $\mathcal{C}'_{IP}$ QML estimators, and the BB power spectrum (left panel) obtained with the three different configurations. For all the different power spectra, we find that all the estimations are unbiased and that the corresponding errors are very similar for the considered configurations. We also checked that, as expected, the polarization spectra is identical for the $\mathcal{C}'_{IP}$ and $\mathcal{C}_P$ approaches.

In order to quantify the different performance of the three estimators, we have calculated the increase in the error of the $\mathcal{C}'_{IP}$ and $\mathcal{C}_P$ configurations with respect to the $\mathcal{C}_{IP}$ reference case, which gives the best estimation since it uses the complete information in the fiducial model. In particular, for the $\mathcal{C}'_{IP}$ approach, we have calculated this increment in the error as

$$\sigma_{\text{rel}}\left(\mathcal{C}'_{IP}\right) = 100\frac{\sigma\left(D_\ell^{\mathcal{C}'_{\text{IP}}}\right) - \sigma\left(D_\ell^{\mathcal{C}_{\text{IP}}}\right)}{\sigma\left(D_\ell^{\mathcal{C}_{\text{IP}}}\right)}, \tag{6.2}$$

where $\sigma\left(D_\ell\right)$ is the dispersion obtained from the 5000 simulations.

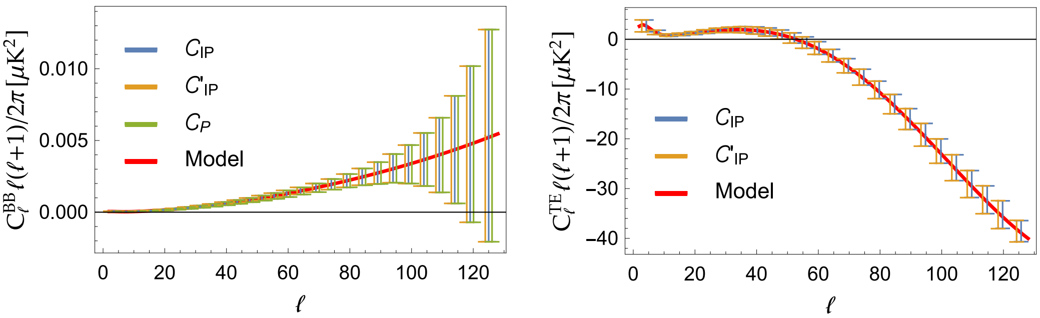


Figure 6.2: Estimated power spectrum for BB (left) and TE (right) obtained with the different QML configurations explained in the text. For comparison the fiducial model is also plotted (red line). Error bars have been computed as the dispersion from 5000 simulations (we have tested that in the cases that the fiducial model matches the power in the maps, i.e., $\mathcal{C}_{IP}$ and $\mathcal{C}_P$, the error estimated from simulations agrees very well with that obtained from the Fisher matrix). For a better visualization, the power spectra has been binned and some of the points have been plotted with a small shift in the multipole value.

Figure 6.3 shows this quantity for TT, EE and BB (left panel) and for TE, TB and EB (right panel) when estimating the power spectrum with $\mathcal{C}'_{IP}$ (note that $\sigma_{\rm rel}(\mathcal{C}_P)$ is identical to that of $\mathcal{C}'_{IP}$ for the EE, BB and EB spectra). When excluding the information about TE in the fiducial model, we are slightly increasing the error bar in the final estimation of the different spectra. However, this increase is very moderate, showing that the method is close to optimal and that, if one is only interested in polarization spectra, the method can be implemented specifically for this case with an important reduction of computational resources. Let us remark that the reduction of computational time can be important, for instance, if one needs to repeat the process over many data sets or to iterate over the fiducial model (see Section 6.5). Another important advantage is that for a given set of computational resources, this implementation can work up to higher $N_{\rm side}$ and greater $\ell_{\rm max}$ than the complete TEB configuration, since the memory required to store the matrices in the EB implementation is significantly lower.

6.4 Performance of the binned estimator

Figure 6.4 shows the results of the application of this method to an estimation of the mean power spectrum and its corresponding error using 1000 simulated maps at resolution $N_{\rm side} = 128$ for the ground configuration given in table 6.1, up to $\ell_{\rm max} = 256$. The fist bin runs from 2 to 5; the second, from 6 to 10; and the rest of the bins are of length

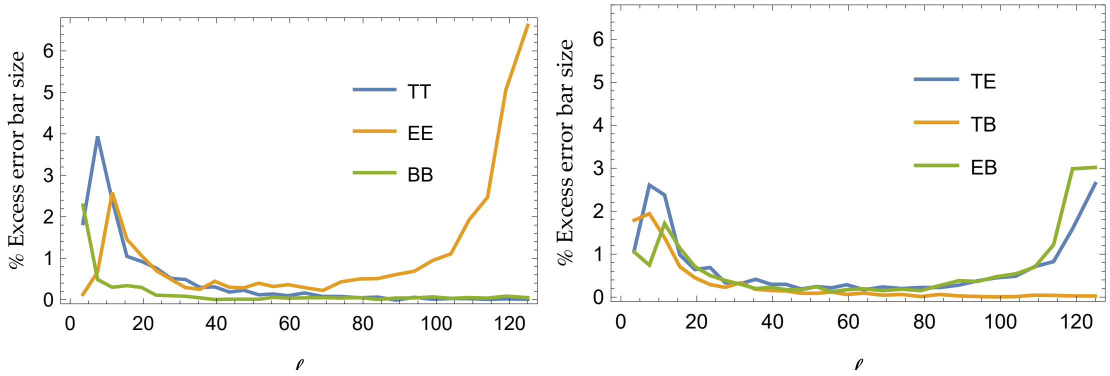


Figure 6.3: Relative difference in per cent of the estimation error of the different power spectra given by the $\mathcal{C}'_{IP}$ implementation with respect to that of $\mathcal{C}_{IP}$. These quantities have been computed with eq. (6.2).

ten. As seen, the agreement between the estimated binned spectrum and the underlying true model is very good and we also find a very good match between the errors estimated from simulations and from the Fisher matrix. Therefore, this confirms that the method is unbiased and of minimum variance. The values of B_b (and consequently the values of f_ℓ^b) and ℓ_b^* were calculated from the fiducial model as the mean of the values in the bins weighted by their theoretical error (according to eqs. (4.1) and 4.2)).

6.5 Performance when the fiducial differs from the model in the maps

As shown in Chapter 2, the QML method is, under the considered assumptions, unbiased and of minimum variance. However, this requires the use of the correct fiducial model, what is in general unknown. In Section 2.6.3 we have also shown that the method is unbiased when the fiducial does not reflect the true underlying model on the map, but not of minimum variance. In this case, one may use an iterative scheme, such that the initial fiducial model is updated taking into account the output of the QML. Therefore, it is interesting to test this approach and to check how the estimator and its error depend on the choice of the fiducial model.

So far, we have shown results for the QML method using a fiducial model that perfectly matches that of the simulated maps, with the exception of the calculation of C_ℓ^{TE} in the $\mathcal{C}'_{IP}$ implementation, in which $C_\ell^{TE} = 0$ is assumed for the fiducial model. In this section, we will study the robustness of the results when the fiducial model differs

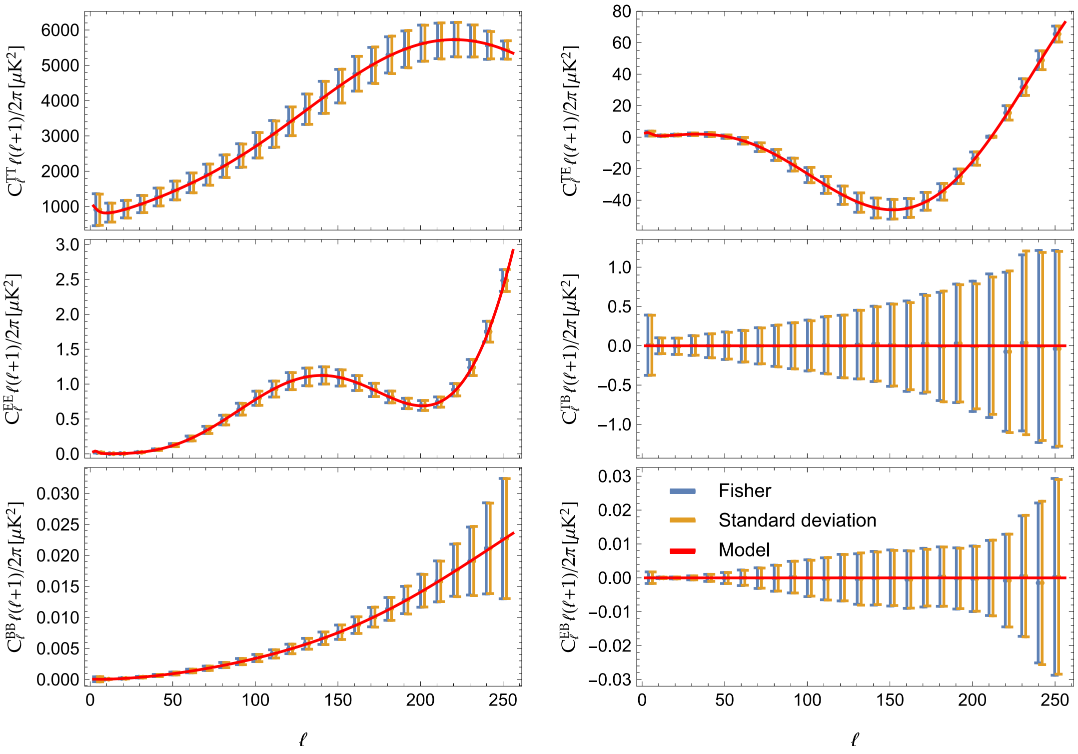


Figure 6.4: Unbiased binned power spectrum estimation in a case when the sky coverage is such that the Fisher matrix becomes singular. Ground experiment at resolution $N_{\text{side}} = 128$. The red line shows the model in the simulated maps used as fiducial; the orange error bars, the dispersion on the estimated power spectra obtained from the 1000 simulations; the blue error bars, the error estimated from eq. (4.17).

from that assumed for the simulations in different cases, also checking the convergence of an iterative approach. We will also show the performance of the binned method and a comparison of the results for the full and only-polarization approaches described in Section 6.3.

6.5.1 Robustness of QML with respect to the assumed fiducial model

In Section 2.6.3 we have shown that the method is unbiased even when the fiducial differs from the model in the maps, but not of minimum variance. In this section we will check the performance of the method under these circumstances.

In order to test the robustness of the power spectra estimated by QML versus the initial assumed model, we have generated $N_{\text{sim}} = 10000$ simulations in the space configuration using our *Planck* model (i.e., the *Planck* best-fit ΛCMD model but adding

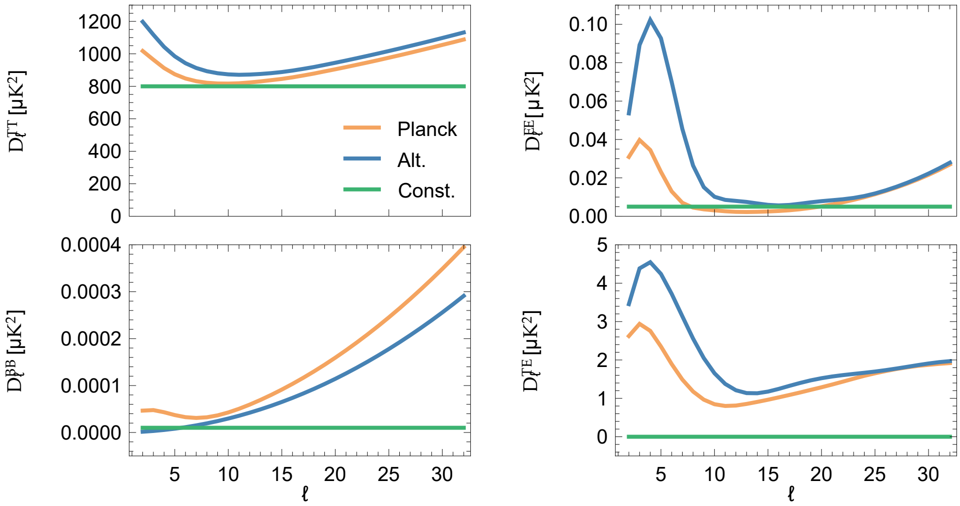


Figure 6.5: Power spectra considered to test the robustness of the QML method versus the assumed fiducial model.

r=0.003) for a resolution of $N_{\rm side} = 16$ and $\ell_{\rm max} = 32$. We have estimated the power spectra of each simulation assuming three different power spectra: the *Planck* model (i.e., the correct underlying model), an alternative ΛCDM model (with a larger scalar amplitude than the previous *Planck* model and $r = 0$) and a constant value for each of the six components. The three models are given in figure 6.5. Note that we have assumed a null TB and EB spectra for the three cases.

We find that the estimated power spectra averaged over the simulations follow the *Planck* model closely, for the three considered cases, i.e, as expected, the method is unbiased even if the power spectrum assumed to calculate the QML estimates differs from the true underlying model.

To quantify this result, we have calculated the relative bias β_ℓ between the true and estimated spectra (averaged over simulations) with respect to the estimated error on the mean average of the power spectrum, i.e.,

$$\beta_\ell^{\rm Altern.} = \frac{\langle D_\ell^{\rm Altern.} \rangle - D_\ell^{\rm Planck}}{\sigma_\ell^{\rm Altern.}/\sqrt{N_{\rm sim}}} \qquad (6.3)$$

when assuming the alternative ΛCDM model and analogously for the *Planck* and constant spectra. Figure 6.6 gives the relative bias for the three considered fiducial models, showing that there are not significant outliers for any multipole or spectra component, confirming that the method is unbiased.

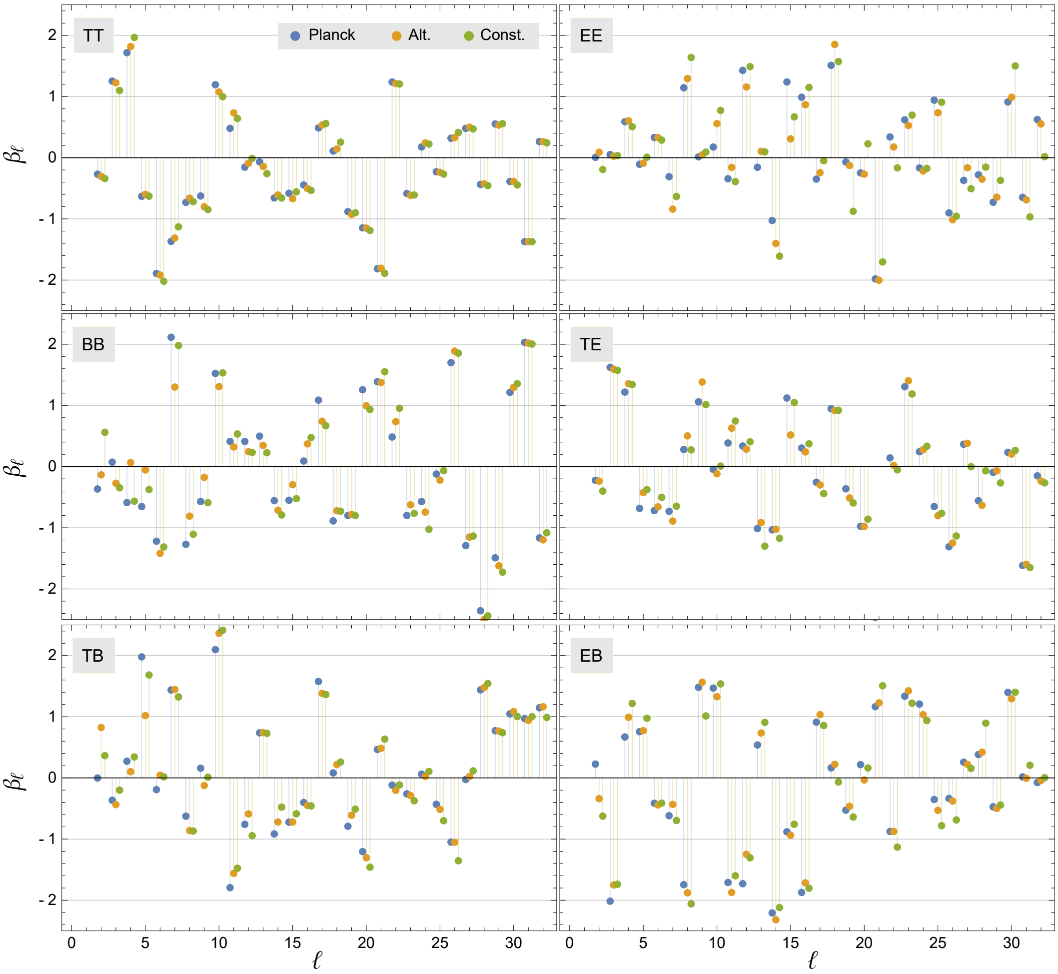


Figure 6.6: Relative bias on the power spectra estimated with QML when starting with the *Planck* (blue), alternative ΛCDM (orange) and constant models (green) obtained using 10000 simulations generated with the *Planck* model.

Regarding the errors of the estimated power spectra, as expected, we find that they are generally sub-optimal when starting with a wrong fiducial model. This has been quantified by looking at the ratio between the errors σ_ℓ at each multipole obtained from simulations with a wrong fiducial (alternative or constant) versus those of the *Planck* fiducial, i.e.,

$$\eta_\ell^{\text{Altern.}} = \frac{\sigma_\ell^{\text{Altern.}}}{\sigma_\ell^{Planck}}, \tag{6.4}$$

when assuming the alternative ΛCDM model and analogously for a constant spectrum.

Fiducial	TT	EE	BB	TE	TB	EB
Maximum η_ℓ						
Alternative	1.01	1.11	1.79	1.04	1.37	1.39
Constant	1.02	1.11	1.30	1.05	1.14	1.14
$\langle\eta_\ell\rangle$						
Alternative	1.00	1.02	1.08	1.01	1.04	1.05
Constant	1.01	1.07	1.04	1.03	1.02	1.06

Table 6.2: Top: maximum value of η_ℓ, the ratio between the errors obtained when the assumed fiducial is the alternative or the constant model versus those obtained when the fiducial is *Planck*. Bottom: average value of η_ℓ.

In particular, table 6.2 shows the maximum value of η_ℓ for the alternative and constant cases (top) and its average value over multipoles (bottom) for the six components of the power spectra. The maximum difference is found for the alternative case and the BB spectra corresponding to a multipole $\ell = 2$, with a ratio of 1.79. The TB and EB spectra are also affected, while for TT, the errors increase only slightly. Regarding the mean ratio, again the alternative case for the BB spectra gives the largest errors (1.08) when compared to the case when the correct fiducial model is used. In summary, these results confirm that the QML estimate is unbiased versus the choice of a fiducial model but sub-optimal with regard to its errors. This leads naturally to the possibility of using an iterative scheme, where the assumed fiducial model is updated taking into account the output of the QML estimator. In the next subsections we will check the validity of this approach.

We may wonder if the binned QML will also be robust versus the choice of the fiducial model. As shown in Section 4, we recall that the binned QML makes used of the information of the fiducial model not only to evaluate the matrix $\mathbf{C}$ (as in the unbinned case) but also to construct the $\{f_\ell^b\}$ set which is needed to reduce the dimensionality of the Fisher matrix in order to make it regular.

To test the performance of the binned estimator, we have applied it to the same 10000 simulations considering the three different fiducial models again and using the bins limits

$$\ell^b_{\text{high}} = \{4, 8, 12, 16, 20, 24, 28, 32\}. \tag{6.5}$$

Figure 6.7 shows the relative bias β_ℓ for the different assumed fiducial models when estimating the power spectra with the binned QML. In this case, we find that the method is unbiased only when starting with the correct fiducial model (*Planck*), while

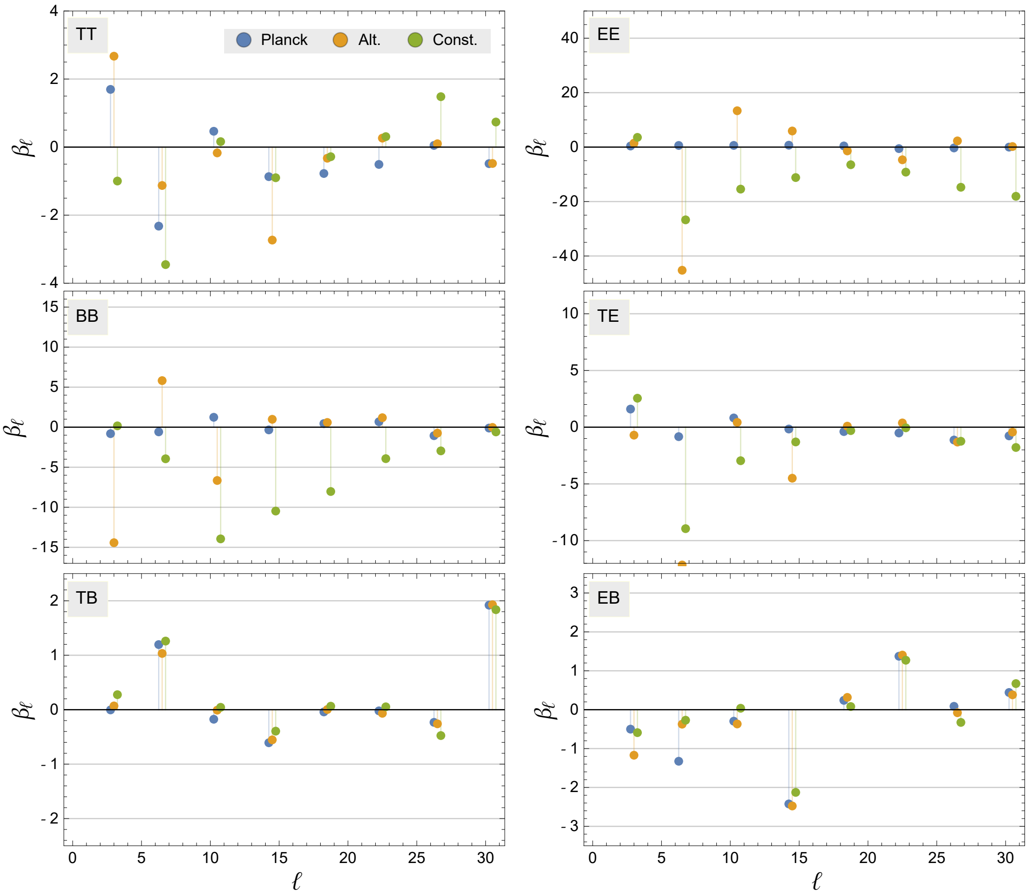


Figure 6.7: Relative bias on the power spectra estimated with the binned QML when assuming the *Planck* (blue), alternative (orange) and constant (green) models obtained using 10000 simulations generated with the *Planck* model.

significant biases are found when assuming the constant or alternative models. This reflects the fact that the binned QML is more sensitive to the choice of the initial power spectra. This could be understood since this quantity appears in a different way in the binned estimator. In particular, for the standard QML the estimated spectrum is given by eq. (2.50), therefore the fiducial model enters twice through the inverse of the covariance matrix in the matrices $\mathbf{E}_i$ (eq. (2.44)) and $\mathbf{F}$ (eq. (2.38)). The estimation itself is given by the multiplication of the vector $\mathbf{y}$ and the inverse of the Fisher matrix, being its effect somehow partially compensated. However, in the binned QML estimation the fiducial enters also through the matrix $\mathbf{R}$ (eq. (4.18)) through the factors $\{f_\ell^b\}$. This matrix is used twice to reduce the size of the Fisher matrix and only once for the same process for the vector $\mathbf{y}$ and, therefore, the effect of the fiducial is more unbalanced.

Fiducial	TT	EE	BB	TE	TB	EB
Maximum η_ℓ						
Alternative	1.01	1.11	1.25	1.05	1.18	1.25
Constant	1.01	1.12	1.08	1.07	1.07	1.30
$\langle\eta_\ell\rangle$						
Alternative	1.00	1.02	1.06	1.01	1.04	1.07
Constant	1.00	1.06	1.00	1.04	1.02	1.08

Table 6.3: Top: maximum value of η_ℓ, the ratio between the errors obtained when the assumed fiducial is alternative or constant versus those obtained when the fiducial is *Planck*. Bottom: average value of η_ℓ. Results are obtained for the binned QML.

We may wonder if the situation improves if we do not include the information about the fiducial model in the binning (i.e., using $f_\ell^b = 1$). However, we note that this is actually equivalent to assume a constant fiducial model for the binning step. Therefore, when including constant weights in the binning, we are actually considering a extreme case for the fiducial and, as one would expect, this leads in general to larger biases. Therefore, once a (reasonable) fiducial model is assumed, it is convenient to include this information in all the steps.

We should also note that the biases found are well within the error of the estimated power spectra for a single realization and, therefore, in practice, they are relatively small. This can be better appreciated in figure 6.8, which, as an illustration, shows that the three estimations for the EE (left) and BB (right) spectra are actually quite similar independently of the initial guess. However, the small error in the average power spectra allows to detect the presence of these biases. One can also appreciate that, when assuming a wrong fiducial spectrum, the output binned QML moves from it towards the correct model. This is a clear indication that the initial fiducial model does not reflect the true underlying spectra and, therefore, some kind of iterating scheme is recommended. This is discussed in more detail for the binned estimator in Section 6.5.3.

Regarding the increase of the estimation error, table 6.3 shows the maximum and mean value of η_ℓ for the binned QML. We find a similar behavior to that of the standard QML, although with lower ratios, especially for BB whose maximum ratio is found to be 1.25 (when starting with the alternative fiducial model).

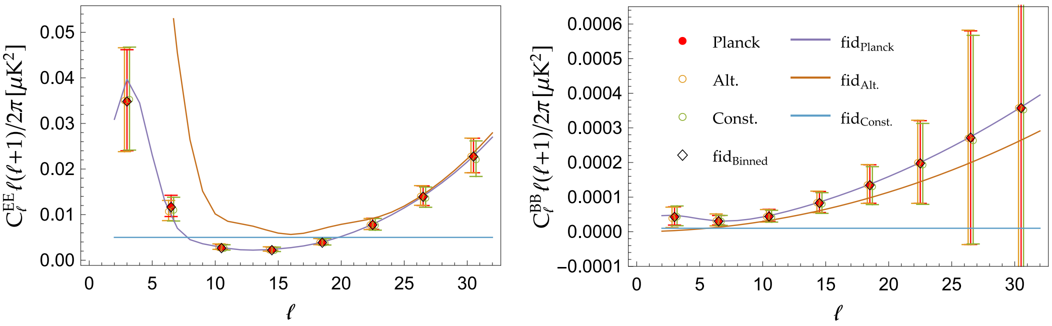


Figure 6.8: The mean and dispersion of the power spectra for EE (left) and BB (right) estimated with the binned QML, averaging over 10000 simulations and starting with three different fiducial models are shown. The used fiducial models (labeled as $\mathrm{fid}_{\mathrm{Planck}}$, $\mathrm{fid}_{\mathrm{Alt.}}$ y $\mathrm{fid}_{\mathrm{Const.}}$) are also given as solid lines. For comparison, the binned spectra obtained from the *Planck* model (labeled as $\mathrm{fid}_{\mathrm{Binned}}$), i.e., the model used to generate the simulations, is also given.

6.5.2 Iterative QML

To test if an iterative process would lead to an unbiased and minimum variance QML estimator, independently of the initially assumed fiducial model, we have carried out a further test using 500 simulations with the same characteristics as those of the previous subsection. In particular, we have estimated the power spectra with three different initial power spectra (as before, *Planck*, alternative ΛCDM and constant spectra) for each of the simulations. After this first estimation, we have iteratively estimated the spectra another four times modifying the assumed fiducial model taking into account the output of QML for the previous step. Therefore, in this test, we have applied QML a total of $3 \times 500 \times 5$ times. In this subsection, we have considered only the standard (unbinned) QML, while the performance of the binned QML will be explored for a specific case in the next subsection.

When iterating, one could simply use as the updated fiducial model, the values of the power spectra directly estimated with QML. However, we have tested that this may lead to failures in the method, due for instance to the fact that some multipoles are estimated as negative (especially for BB),[2] leading to singular covariance matrices. Therefore, it is convenient to use a smoothed version of the output spectra as a guess

---

[2] Note that, although by definition auto-correlation multipoles are positive, estimators do not necessarily produce these values. An estimator is just a mathematical operation that assigns values to the parameters. And, in principle, one should not expect the function to assign values that satisfy the theoretical mathematical properties of the parameters it estimates. On the other hand, the values given by an

for the next iterative step. Details about how this smoothing has been implemented are given in Appendix E. Even with this approach, some instabilities can be present and it has not been possible to complete the full iterative process for all the simulations and the three different initial fiducial models. Note that one could complete the full process for all the simulations by tuning the parameters used in the smoothing. However, we have seen that this does not affect our results and, for simplicity, we have just discarded those simulations that have failed at some step. Therefore, we present results for a total of 453 simulations, for which the full process has been completed without further tuning.

To test the performance of the iterative QML, we have studied the convergence of the method independently of the initial fiducial model, the consistency of the results for the three considered initial spectra and the evolution of the estimation errors with iterations.

Regarding the convergence of the method, we find that after around five steps (i.e., the initial QML plus four iterations), the results are already quite stable. In particular, we have studied the evolution of the convergence by looking at the relative difference between two consecutive steps with respect to the spectra estimated with the correct fiducial model (in this case without iterating), i.e.,

$$\delta_\ell = \frac{\langle D_\ell^{\mathrm{j+1,Altern.}} - D_\ell^{\mathrm{j,Altern.}} \rangle}{\langle D_\ell^{\mathrm{Planck}} \rangle} \times 100, \tag{6.6}$$

when starting with the alternative ΛCDM model and analogously for the constant model. Note that j corresponds to the step in the iterative process and that averages are obtained over simulations. As an example, figure 6.9 shows this quantity for two cases: TT spectrum starting with the constant model (left) and BB spectrum starting with the alternative model (right). As one would expect if the method converges, these differences decrease when advancing in the number of iterations. A similar behavior is found for the other considered cases.

We have also tested if the results obtained with the QML by the three different starting fiducial models converge to the same values as the iterations progress. To quantify this point, we have calculated for each simulation and for each iteration the dispersion between the three estimations obtained with the three different starting models (*Planck*, alternative and constant) at each multipole. Therefore we have the function

$$\sigma_{\ell,i}^{\mathrm{j}} = \mathrm{Dispersion}\{D_{\ell,\mathrm{i}}^{\mathrm{j,Planck}}, D_{\ell,\mathrm{i}}^{\mathrm{j,Alt.}}, D_{\ell,\mathrm{i}}^{\mathrm{j,Const.}}\} \tag{6.7}$$

---

estimator are accompanied by some uncertainty, so there is a certain probability that a parameter that in principle is positive will end up having a negative value.

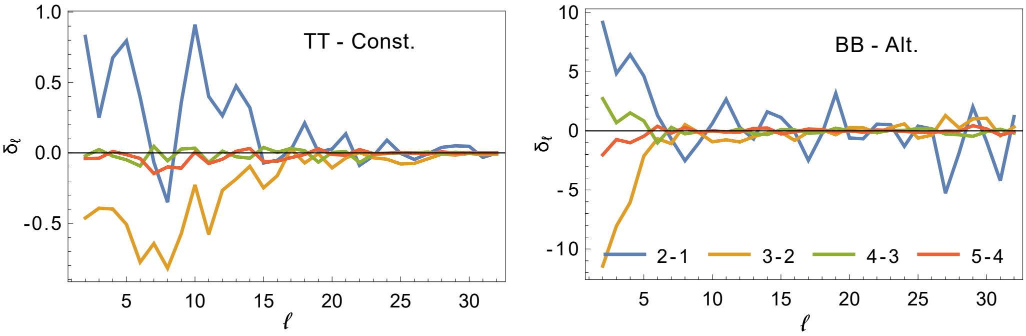


Figure 6.9: Convergence of the iterations for TT when starting with the constant spectra (left) and BB when starting with the alternative model (right). The different colors indicate the relative difference (in percentage) between two consecutive steps averaged over simulations with respect to the value estimated with the correct fiducial model.

for each multipole ℓ, simulation i and step j. Figure 6.10 shows the ratio (in percentage) of this quantity averaged over simulations relative to the estimated spectra obtained when starting with the *Planck* model (without iterating), also averaged over simulations, for BB (left) and TE (right). As seen, the dispersion between estimates is significantly reduced when increasing the number of iterations, showing that the iterative QML leads to very similar results, not only on average but also simulation by simulation, independently of the chosen initial spectra. Similar conclusions are reached for the other spectra.

We have also checked that, independently of the starting fiducial model, the errors in the estimated power spectra at the end of the iterative process are very similar to those obtained for the optimal case (i.e., using the correct fiducial model and not iterating). This is expected since we have seen that the iterative QML converges basically to the same result for the three considered cases. Note that these conclusions also hold for the case in which we start with the correct fiducial model, showing that the process is stable and that there is not danger in iterating even when one is already in the right initial point.

6.5.3 Robustness of QML with respect to the assumed tensor-to-scalar ratio

In the previous subsections, we have considered the robustness of the QML versus the initial choice of the fiducial model as well as the performance of an iterative approach for different initial generic spectra. However, future experiments will focus on the estimation of the tensor-to-scalar ratio and, therefore, we think it is interesting to study

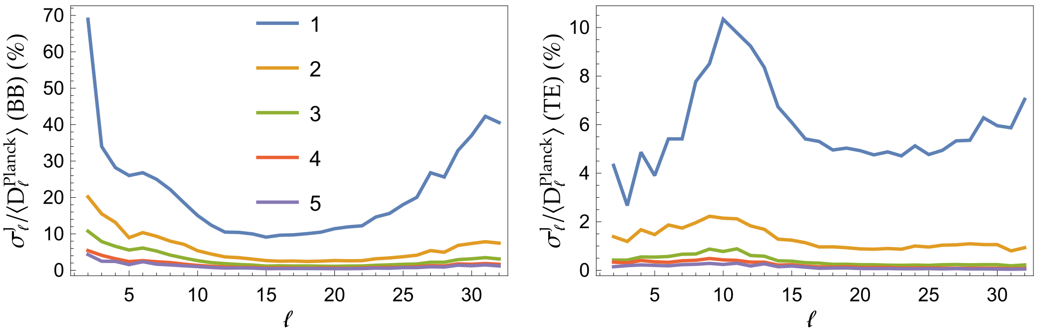


Figure 6.10: Ratio (in percentage) of the average dispersion obtained over the three different estimates of the spectra over the average power spectra obtained with the *Planck* fiducial (without iterating) for each iterative step $j = 1,5$ (indicated by different colors) for BB (left) and TE (right).

specifically the sensitivity of the method to a wrong initial value of r. Given that many of these experiments observe only a small fraction of the sky, we will also consider the binned version of the QML. This also allows us to check the consistency between both approaches and whether information could be missed when using the binned estimator.

In this section, the iterative approach will also be considered, although we will use a different method to provide an initial guess for the fiducial model. We will assume that all cosmological parameters are known, except for the tensor-to-scalar ratio r, so we will consider three different initial fiducial models for QML that differ in the value of r. Then, rather than smoothing the output spectra of the previous step, we will estimate r from the QML spectra and use it to construct the guess spectra for the next iteration. The estimator used for r is described in detail in Appendix F.

In particular, we have carried out the following procedure:

1. We simulate one map (including CMB and noise) with the specifications of the space case (given in table 6.1) and with a tensor-to-scalar ratio $r_{true} = 0.003$. The map is generated at $N_{\rm side} = 16$, with $\ell_{\rm max} = 32$ and smoothed with a Gaussian beam of 8.79 degrees of full-width half maximum.

2. We apply the unbinned and the binned QML to estimate the power spectrum, starting with a wrong fiducial model (i.e., $r_0 \neq r_{true}$). More specifically, we consider two cases: $r_0 = 0$ and $r_0 = 0.03$. For comparison, the case $r_0 = r_{true}$ is also considered. For the binned estimator, we have used 8 bins with $\ell^b_{\rm high}$ given by {4, 8, 12, 16, 20, 24, 28, 32}.

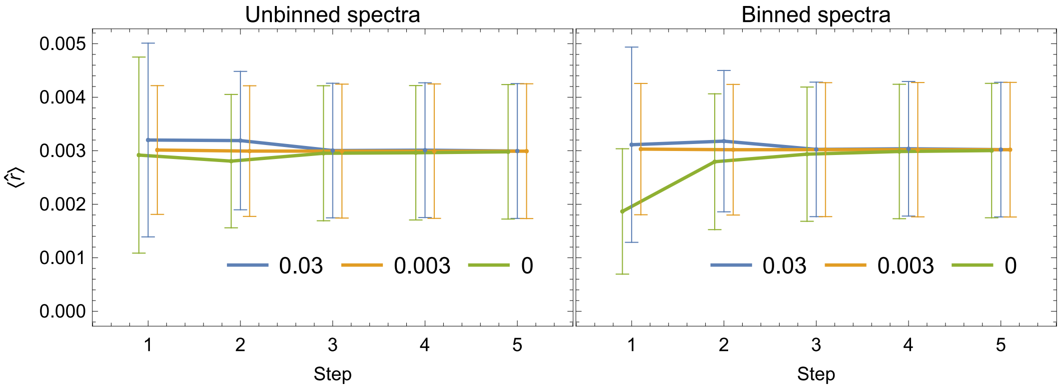


Figure 6.11: Left: evolution through the steps on each one of the five iterations with starting points $r_0 = 0.03$, 0.003 and 0, of the mean and the standard deviation over 200 simulated maps of $\hat{r}$ estimated from the QML power spectrum. Right: $\hat{r}$ obtained from the same maps in the same conditions than in the left part, but from a binned spectra.

3. An estimator $\hat{r}$ of the tensor-to-scalar ratio is obtained as explained in Appendix F.
4. We update the fiducial model using the estimated value $\hat{r}$ and apply again the unbinned and binned QML. A total of five iterative steps are performed.

The full process is repeated for 200 simulations.

Figure 6.11 shows the progression with the number of iterations of the mean value and dispersion of $\hat{r}$ obtained over the simulations, for the three values of r_0 and for the unbinned (left) and the binned (right) QML estimator. For the standard (unbinned) estimator, the results indicate that when starting with a wrong fiducial, even without any iteration, the estimator is close to unbiased although not of minimum variance. For the binned estimator, this is also the case when starting with a fiducial model with $r_0 = 0.03$. However, for $r_0 = 0$ without iterating, we find that the mean of the estimated values of r is around 1σ below the true value. Since that for the standard QML case we do not find that deviation (left panel), this indicates again that the binned QML is more sensitive to the choice of the initial fiducial model. However, we see that $\langle \hat{r} \rangle$ converges rapidly with the iterations to the true value, independently of the starting point, for both the binned and unbinned QML. Also, the error of the estimation of r decreases, becoming quite stable after around four steps.

The good convergence of the iterative QML can also be confirmed by looking at the top and bottom-left panels of figure 6.12 that show, for each simulation, the estimated value of r for one starting point ($r_0 = 0$) versus the one estimated for the other initial value ($r_0 = 0.03$) for different number of iterations. For both, the unbinned (top panel)

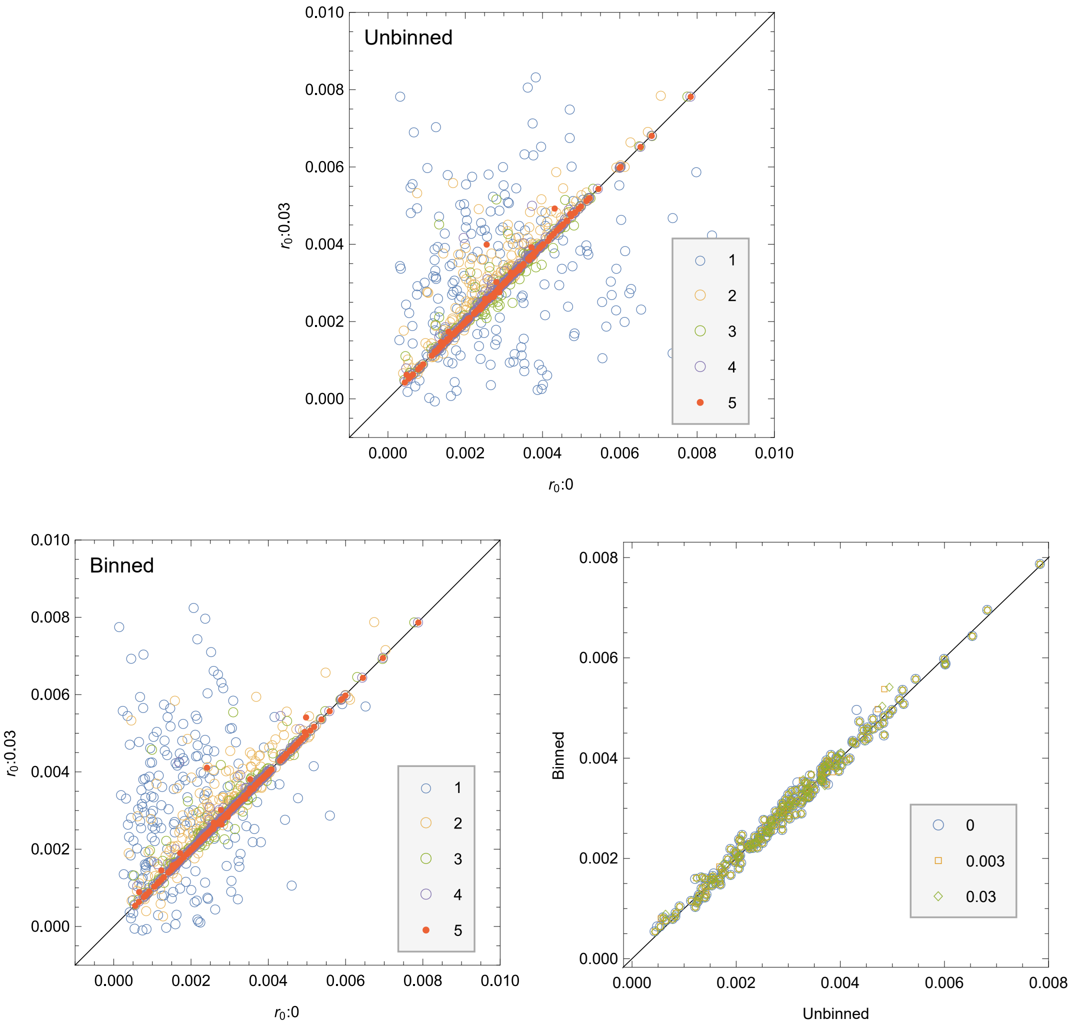


Figure 6.12: Top: Tensor-to-scalar ratio estimated from the QML spectra starting with a fiducial model with $r = 0.03$ (ordinate) plotted against the tensor-to-scalar ratio estimated starting with a fiducial model with $r_0 = 0$ (abscissa) for different number of iterations. Bottom-left: same plot using the binned spectra. Bottom-right: $\hat{r}$ obtained at the last step of the iteration from the unbinned spectra versus that obtained from binned spectra, for the three different starting points of r_0 considered. In all cases, points from 200 simulations are shown.

and binned (bottom-left panel) QML, we see that individual values of $\hat{r}$ tend to migrate to the diagonal of the plot through the steps of the iteration, showing the good

	r_0	$\langle\hat{r}\rangle$	$\sigma_{\hat{r}}$
Standard QML	0	3.00×10^{-3}	1.26×10^{-3}
	0.003	3.02×10^{-3}	1.26×10^{-3}
	0.03	3.02×10^{-3}	1.26×10^{-3}
Binned QML	0	2.98×10^{-3}	1.26×10^{-3}
	0.003	2.99×10^{-3}	1.26×10^{-3}
	0.03	2.99×10^{-3}	1.26×10^{-3}

Table 6.4: Mean and standard deviation of $\hat{r}$ from 200 simulated maps (space configuration) at resolution $N_{\text{side}} = 16$ generated with $r_{\text{true}} = 0.003$ at the last of the five steps of an iterative scheme for the standard and binned QML. In both cases, values obtained after an iterative process with different starting points are shown.

performance of the iterative approach.[3] For the binned QML, it also becomes apparent that when no iterating, the estimation of r tends to be lower when starting with $r_0 = 0$ versus the values obtained with $r_0 = 0.03$ as reflected in the asymmetric distribution of blue circles around the diagonal. The bottom-right panel of figure 6.12 shows the estimated values of r, obtained at the last step of the iterations, from the binned spectra versus those obtained from the unbinned one, for the three different starting points considered. It is apparent that the two estimations of r (obtained from the binned and unbinned QML) are very similar, clustering around the diagonal. In addition, we see again that the position of the points is independent of the value of r_0.

Finally, table 6.4 gives the mean value and standard deviation of $\hat{r}$ in the last iterative step for the three starting points for the unbinned (top) and binned (bottom) spectra, showing an excellent agreement between the different cases. This shows that, at least in the considered case, no information is lost when using the binned version of the QML with respect to the standard implementation and that the iterative approach is robust versus the choice of the initial tensor-to-scalar ratio for the binned and unbinned QML. For comparison, we note that the theoretical errors (Δr given by eq. (F.5)) for a fiducial model with $r = 0.003$, are 1.16×10^{-3} and 1.18×10^{-3} for the unbinned and binned spectra, respectively, which are somewhat below those found for $\sigma_{\hat{r}}$. This difference is due to the relatively small number of simulations, which only allows estimating the error of $\hat{r}$ with limited precision.

---

[3] Note that this test is stronger than that of figure 6.11, since we impose convergence at each individual simulation, finding that a few values of $\hat{r}$ still deviate from the diagonal at step 5. Indeed, we tested that with more iterations (with around ten steps in total), these values also move to the diagonal

Specification	$\langle\hat{r}\rangle(\times 10^{-3})$	$\sigma_r(\times 10^{-4})$	$\Delta r(\times 10^{-4})$
$\mathcal{C}_{IP}$	3.00	6.51	6.50
$\mathcal{C}'_{IP}$	3.00	6.52	6.51
$\mathcal{C}_P$	3.00	6.52	6.51

Table 6.5: Results on the estimation of r in the configurations $\mathcal{C}_{IP}$, $\mathcal{C}'_{IP}$ and $\mathcal{C}_P$ described in Section 6.3.

6.5.4 Only-polarization QML

Finally, we may also wonder whether the estimation of $\hat{r}$ is degraded when using the only-polarization QML configuration ($\mathcal{C}_P$) versus the full estimator ($\mathcal{C}_{IP}$) described in Section 6.3. This is important because of the reduction in CPU time achieved when working only with polarization. Therefore, we have applied the configurations $\mathcal{C}_{IP}$ and $\mathcal{C}_P$ (and also $\mathcal{C}'_{IP}$ for comparison) to 5000 simulations (space configuration, $N_{\text{side}} = 64$, $\ell_{\text{max}} = 128$) starting from the correct fiducial $r = 0.003$ and have estimated the mean value and error for $\hat{r}$.

Table 6.5 shows the mean value, standard deviation and theoretical error bar of $\hat{r}$ for the three considered configurations. As one would expect, since $\mathcal{C}_{IP}$ includes the full information from the power spectra, it provides a slightly smaller error on $\hat{r}$ than the other two configurations which, at the considered precision, are indistinguishable. However, the differences are very small and, therefore, in practice, it is perfectly acceptable to use the $\mathcal{C}_P$ configuration in order to save computational resources.

6.6 Comparison between QML and `NaMaster`

The so-called pseudo-spectrum methods (e.g. [44]), have become widely used to estimate the CMB power spectra since they require significantly lower computational resources than QML, allowing their computation up to very high multipoles. These methods calculate the spherical harmonic transform in a masked sky and try to deconvolve the effect of this mask through the inverse of the kernel that encodes the coupling in the harmonic space produced by the loss of orthogonality due to the incomplete sky. To reduce the effect of the coupling, masks are usually apodized (e.g. [47]). In the polarization case, due again to the loss of orthogonality, a leakage between E and B modes is also present, which adds an additional complexity to the sought of the very weak primordial B-mode of polarization. The most advanced pseudo-spectrum methods incorporate purification techniques of the E- and B- modes of polarization with the aim to reduce this transfer of power between them [101]. Although pseudo-spectrum methods are, in practice, of minimum variance for intermediate and high multipoles, they are sub-optimal at large scales, which are particularly relevant for a future determination of the scalar-to-tensor ratio. Therefore, it is interesting to compare the performance of the QML and pseudo-spectrum estimators, to understand the advantages and limitations of each of them (for previous discussions, see e.g. [102, 94, 54]).

In particular, in this section we compare the results from `ECLIPSE`, our QML implementation, to those obtained with `NaMaster` [59], an advanced public implementation of the pseudo-spectrum method,[4] that incorporates different types of apodization as well as the purification technique. For our test, we have applied QML and `NaMaster` to 10000 simulations in the space configuration at resolution $N_{\rm side} = 64$, with the usual mask (left panel of figure 6.1) and considering $\ell_{\rm max} = 128$. In the case of `NaMaster`, it is possible to tune several parameters (such as type of apodization, apodization scale, to include or not purification) in order to improve the estimated spectra. Although a detailed study of the optimal choice of these parameters is outside the scope of this chapter, we have explored several possibilities, prioritizing the recovery of the lowest multipoles for BB. In particular, we find that the *C2*[5] option for apodization with a scale of 22° and the use of B-mode purification is well suited for our purpose when the previous simple Galactic mask is used. However, note that, as it will also be shown, different configurations may produce better results for other components or scales of the spectra as well as for different masks.

As expected, we find that both `ECLIPSE` and `NaMaster` provide unbiased estimations of the different components of the spectra (in the case of NaMaster after subtracting the noise bias). Figure 6.13 (top panel) shows the ratio of the estimation errors (obtained from the simulations) achieved with `NaMaster` over those from QML for the considered case. It becomes apparent that important differences are found up to ℓ around 20, with maximum values for the ratio of around 5. At the largest multipoles, the ratio is close to 1, but we see that QML still provides better results. This is actually due to our choice

---

[4] `NaMaster` is available at https://github.com/LSSTDESC/NaMASTER

[5] In this case, pixels are multiplied by a factor f given by

$$f = \begin{cases} 0.5\left[1-\cos\left(\pi x\right)\right] & \text{if} \quad x < 1 \\ 1 & \text{otherwise} \end{cases} , \tag{6.8}$$

where $x = \sqrt{(1-\cos\theta)/(1-\cos\theta_*)}$, θ_* is the apodization scale and θ is the angular separation between a pixel and the nearest masked pixel. Note that all pixels separated from any masked pixel by more than the apodization scale are left untouched.

of a very large scale of apodization, which in practice reduces the effective available information, degrading the error of the spectra at higher multipoles. By choosing a smaller apodization scale, this ratio tends to unity at these multipoles although at the price of degrading the recovery of BB at large scales very significantly. This is shown in the left-bottom panel of figure 6.13 where an apodization scale of 4° is instead used. We also note an increase in the estimated error for `NaMaster` at the highest considered multipoles. We found that, at the limit of the resolution of the map, QML also performs better than pseudo-spectrum methods. However, in practice, this is not a real limitation of this technique in comparison to QML, since the pseudo-spectrum method can recover this range of multipoles from maps with higher resolution (where this effect will move to the highest considered resolution, that in any case will not usually be achieved by QML due to computational limitations).

The mask that we have considered in this test is well suited for apodization, since it only presents one boundary between the included and excluded regions. However, masks can also exclude regions outside the Galactic plane that, when apodizing, will introduce a further loss of information for pseudo-spectrum methods. This is not the case for QML, where only the pixels discarded by the mask are removed from the analysis. To test this situation, we have repeated the previous exercise considering the mask given in figure 6.14, which is constructed by excluding additional regions outside the Galactic plane, that are present in the *Planck* common confidence mask for polarization [15]. Note that this extended mask allows the use of 58.6 per cent of the sky versus 59.0 allowed by the original mask (left panel of figure 6.1). For this case, we found that the use of the B purification technique and the *Smooth* apodization option[6] with a scale of 4.5° were giving better results for the lowest multipoles of BB (again we remark that different configurations could be better suited for other purposes). The bottom panel of figure 6.13 shows the ratio between the errors obtained with `NaMaster` versus those of QML. We see that the behavior is qualitatively similar to that found for the original mask (top panel) but the differences between both methods are amplified, confirming that a loss of a small fraction of the sky can degrade the performance of the pseudo-spectrum methods with respect to QML very significantly if the mask is not compact. Indeed, we find that for QML the errors of the estimated spectra increase only slightly (at the subpercent level for all multipoles) with respect to the original mask. However, for `NaMaster`, the estimated errors for the case of the extended mask are between 1.4 to

---

[6] In this case all pixels closer than 2.5 times the apodization scale to a masked pixel are initially set to zero. The resulting mask is then smoothed with a Gaussian kernel with standard deviation given by the apodization scale. Finally, all pixels originally masked are put back to zero.

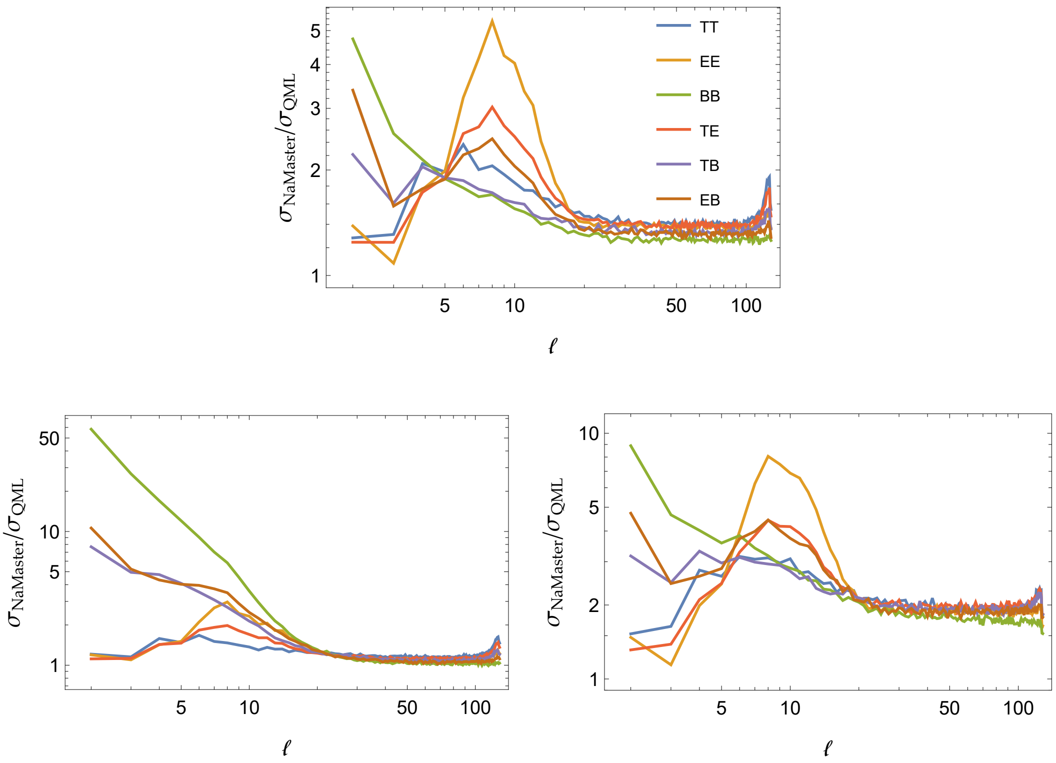


Figure 6.13: Ratio of the errors of the power spectra obtained with `NaMaster` over those estimated with QML. For `NaMaster`, B-mode purification is always used. Top panel and left-bottom panel, a simple Galactic mask has been considered, whereas for the right-bottom panel, the extended mask from figure 6.14 was used. The following apodization options were considered in each case for `NaMaster`: *C2* option with a scale of 22° (top), *C2* with a scale of 4° (left-bottom), *Smooth* with a scale of 4.5° (right-bottom).

2.0 times larger than those of the original mask (being the largest scales more affected). A practical way to improve these results for the pseudo-spectrum methods would be to perform some kind of inpainting in the data that allows the use of a more compact mask, although this incorporates an additional complication to the procedure, whose effect should be carefully quantified.

6.7 Some applications of `ECLIPSE`

`ECLIPSE` has already been used or is being used in several applications, both to data and simulations.

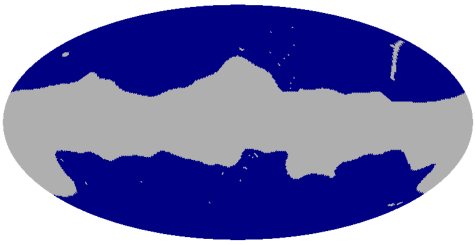


Figure 6.14: Extended mask that covers the Galactic mask from the top panel of Fig 6.1 plus some additional regions outside the Galactic plane constructed at $N_{\text{side}} = 64$. The number of valid pixels is 28824.

A general map-making method for ground-based microwave experiments, implemented specifically for QUIJOTE multifrequency instrument (MFI) data, is presented in [103]. TOD's containing sky signal have been simulated following the equations of the MFI instrumental response. The signal contains CMB plus foregrounds, dipoles (solar and orbital) and noise. The PICASSO map making method has been used to construct the observed CMB map and the other components that constitute the measured signal. To determine the transfer function at large scales, QML has been used to estimate the power spectra of the input map and the recovered map, as it is especially important to have a good estimate at low multipoles where pseudo-spectra methods are not optimal.

QML has been applied iteratively, starting from three different initial points. This process is not straightforward, presenting some subtleties that need to be dealt with in the analysis. In particular, due to the characteristics of the mask, the Fisher matrix is singular, so it is only possible to estimate the binned power spectrum. However, to iterate, it is necessary to generate a full spectrum from the binned spectrum, in order to calculate, again, the covariance matrix. Although in all tests involving iteration we have found that QML converges to a final estimate, which is independent of the starting point, the final result may be sensitive to the binning scheme and the way the full spectrum is constructed for the next iteration. These points were studied in detail in order to produce a robust estimation of the transfer function.

In [104], a methodology to determine the miscalibrated polarization angles is developed. In particular, the miscalibration angles are estimated by nulling the EB component of the power spectrum, as expected in the standard model for CMB. With this method, the miscalibration angles are fitted and corrected (de-rotated) CMB maps are obtained. To test the method, CMB maps have been simulated from a prefixed model, and foregrounds and instrumental noise have been added. In addition, the effect of the miscalibration polarization angles on the measurement of Q and U for each detector has also been considered. After applying the techniques developed to remove the systematics and recover the CMB maps, the quality of the results has been checked. Among other tests, our code has been used to study the performance of the proposed methods by comparing the power spectrum of the recovered and simulated CMB polarization B-mode as well as the level of residuals left in the output map.

In [105], QUIJOTE intensity and polarization maps in four frequency bands centered around 11, 13, 17 and 19 GHz, and covering approximately 29 000 deg^2 are presented. Given the multipole range considered in the different analyses, and the low computational resources required for pseudo-C_ℓ, this type of estimators have been used to analyze the results of the data pipeline processing and to determine the characteristics of the final maps, in particular, the codes `Xpol`[7] and `NaMaster`. The tests carried out throughout the work have shown that both methods produce consistent results. In addition, taking a reference map and mask, the `ECLIPSE` code —that implements an optimal estimator— has been used in order to compare the results obtained with both methods with the ones given by QML for that reference case. The conclusion is that all methods provide consistent results for the multipole range considered, so it is justified to use the pseudo-C_ℓ approach for the computations.

6.8 Conclusions and discussion

The QML provides an unbiased and minimum variance estimator of the intensity and polarization of the CMB power spectra, provided a correct fiducial model is assumed. If one is only interested in polarization power spectra, we show that it is possible to use an only-polarization QML implementation reducing further the required computational resources (roughly a factor of 2 in CPU and memory requirements) while obtaining basically the same results as those of the full QML implementation (where intensity and polarization are simultaneously considered).

The method is always unbiased,[8] but it is only of minimum variance if the assumed fiducial model corresponds to the underlying true model of the data, which is in generally unknown. To overcome this shortcoming, the use of the QML within an iterative scheme that updates the fiducial model at each step has been tested for different scenarios. Our results show that, even when starting with a wrong fiducial model, the

---

[7] https://gitlab.in2p3.fr/tristram/Xpol

[8] Of course, assuming that the noise model matches the one in the maps. Otherwise, the correct noise bias would not be subtracted and some residual would remain in the estimation, whatever the fiducial.

estimated power spectrum is, as expected, unbiased without iterating, although its error is somewhat larger than when starting with the correct value. Moreover, if we iterate, independently of the starting point, the errors are also consistent with those obtained when starting with the correct fiducial model. The same behavior is found for the estimation of the tensor-to-scalar ratio r from the output power spectra (in a simplified case where all other cosmological parameters are assumed to be known), i.e., its estimation is close to unbiased independently of the assumed initial power spectra but its error can be reduced by iterating. In addition, we have also repeated similar tests for the binned version of the QML, finding that, in certain cases, this may be more sensitive to the assumed initial fiducial model than the standard QML. However, when iterating, both the binned and standard QML estimators provide very similar results. Therefore, when applying the QML estimator to future CMB data is advisable to iterate at least a few steps in order to check the consistency between the assumed fiducial model and the one estimated with QML, in order to obtain optimal results. Note that the high efficiency of our code (see Chapter 5) allows the use of this iterative approach that could not be easily carried out with previous algorithms due to the high required computational resources.

This leads us to a very important point to keep in mind: the QML method guarantees that the error bars derived from the Fisher matrix obtained by applying the method match the correct covariances when the fiducial reflects the true underlying model in the maps. Only in that case we can be sure that the Fisher error bars are accurate. When we apply QML and get a spectrum that does not match the fiducial, we cannot rely on the Fisher error bars, and iteration will be necessary.

We have also compared the performance of the QML method with that provided by a pseudo-spectrum estimator (using the `NaMaster` implementation). For the configuration of a future satellite experiment, we find that the errors of the estimated power spectra at low multipoles ($\ell \lesssim 20$), which are critical for the detection of the tensor-to-scalar ratio, are significantly higher (up to a factor of 5 for a typical Galactic mask) for the pseudo-spectrum method. This method is also found to be much more sensitive than QML to the mask's geometry, degrading its performance when the mask is not compact, even if only a small fraction of the sky is removed. Also, the use of large scales of apodization, which are useful to improve the recovery of the BB spectra at low multipoles, increases the errors at the smallest scales considered, due to the loss of information. This also illustrates the fact that different tunings of `NaMaster` are needed to obtain the best possible spectra for different components or scale ranges.

Chapter 7

Conclusions

The Cosmic Microwave Background (CMB) is a radiation field incident from all directions in the sky. The CMB photons we observe nowadays connect us to the state of the universe when it was around 380 000 years old. In turn, knowledge of the state of the universe at that epoch connects with the initial state by means of well-known physical laws. So the CMB helps to establish constraints on the state of the cosmos and on its behavior at very early times.

The universe is immensely large and has existed for billion years. Therefore, our ability to observe and experiment is reduced to a very limited region of space and a portion of the time it has existed as it is. How can we be sure how far in space and how far forward and backwards in time everything we have learned from observing and experimenting over such a small time span compared to the life of the universe and in such a limited region of space is valid? To answer this question, the Cosmological Principle comes to our aid, which states that the universe is as a whole determined by the same physics —it all began with Newton and his Law of Universal Gravitation—. This means that the place we occupy and what we observe have nothing particular and, ultimately, leads to the conclusion that the universe is isotropic and homogeneous at sufficiently large scales. The Cosmological Principle is not only an explicit manifestation of our desire to have the possibility of knowing and understanding beyond what is directly accessible to our instruments: it also provides a solid methodological framework to formulate and test scientific hypotheses according to Occam's simplicity rule. When we combine the knowledge of the laws of physics that we have acquired by experimenting on our planet and its surroundings with all the information that reaches us from the immediate vicinity and even the most remote places of the universe, we find a consistent picture. If this were not true, there would have to be a large number of coincidences for everything to fit together as it seems to do. Implicitly, the Cosmological Principle implies that the laws of physics are constant over time. Otherwise, we would not be able to explain the past with what we have discovered in the present. Furthermore, otherwise, again, we would not understand how everything fits together as well as it does.

When we observe the CMB, we find some patterns of temperature and polarization anisotropies at different angular scales. According to the Cosmological Principle, these anisotropies do not represent different physical laws in certain areas of the universe, but are the manifestation of the intrinsic statistical nature of the realization of physical processes at different points in the universe.

The ultimate characteristics of our universe are determined at their origin by laws of probability. In this context, our universe is just one realization of the infinite possible ones. From its observation, we can infer the laws and the values of the parameters that govern it. Thus, from what is observed in the present and thanks to the knowledge of the laws of physics, we can reconstruct its past and predict its future.

Because of their statistical nature, the particular values of the anisotropies in each area of the sky are of no special interest. The relevant information is encoded in the power of anisotropies at different angular scales, the angular power spectrum. Models and physical theory make it possible to connect the laws, which we assume to exist and whose form and expression we seek, with the power of the spectrum at different scales. Therefore, the angular power spectrum is a valuable observable of cosmological theory.

The inflationary theory explains the origin of anisotropies in the CMB. According to it, they are the result of the amplification to macroscopic scales of quantum fluctuations.

On the other hand, when we talk about the CMB, we refer to electromagnetic radiation and, as such, it can be polarized. Inflationary models predict patterns in the polarization of the CMB signal at different angular scales. The polarization power spectrum thus becomes a crucial observable for the selection of inflationary models and for determining essential aspects in the first instants of our universe. In addition, it contains valuable information to determine aspects related to the parity of the universe.

Unfortunately, our ability to measure anisotropy values in different regions of the sky is limited. Satellite-based missions can read data from the full-sky and ground-based experiments can only observe a small fraction of the sky, the one accessible from their location. But in both cases, the data recorded by our detectors from the signal they receive from certain regions of the sky are affected by a high contamination of emissions in the frequencies occupied by the CMB, as is the case of the area covered by our galaxy. Therefore, we cannot determine the values of the power spectrum in a way that is equivalent to solving an unknown or solving an equation. That is, by the nature of the data, there are no mathematical methods to solve a problem affecting the power spectrum. Therefore, we can do no more than resort to mathematical methods that allow us to estimate the power spectrum.

Whatever the method used, the capacity to estimate parameters is intrinsically limited by the nature of the statistical distribution from which the data are derived. Thus, there are constraints that set minimum bounds on the uncertainty with which parameters can be estimated.

One of the most widely used mathematical methods to estimate parameters is the maximun likelihood method. It is an unbiased and minimum variance method, and we can apply it to power spectrum estimation. But given the computational workload required to use it, it becomes unfeasible when the data sample size is large. That is, when applied to CMB, when the size of the data maps is large.

Mathematical methods equivalent to the maximum likelihood in terms of the quality of the results can be sought, but with a lower computational workload. The quadratic maximum likelihood method (QML) meets both requirements. It is equivalent and is internally equipped with a solution search mechanism that leads directly to the best estimate. With it, the computational workload is significantly reduced, but still remains large, making it necessary to search for optimal ways to implement it. In the development of the method it has been assumed that the anisotropies are Gaussian. Therefore, care must be taken when applying it to calculate power spectra of maps that do not meet this condition.

In applying the method, a number of complex mathematical operations have to be performed. One of the critical points is the inversion of the covariance matrix. But the covariance matrix is not regular per se. The necessary condition for the signal covariance matrix to be regular is that the power spectrum used to describe it introduces a number of spherical harmonics equal to or greater than the size of the maps. However, the combination of the symmetries between pairs of pixels in the different pixelizations with the characteristics of the spherical harmonics leads that the range of the harmonic matrix may be lower than expected. By adding to the signal covariance matrix a diagonal noise matrix, a regular covariance matrix is obtained. The noise regularizes the matrix because, in the end, adding noise is adding a signal that requires a collection of infinite spherical harmonics to be described. The propagation of errors when performing floating point calculations can lead to incorrect results in cases where the matrix is at the limit of being regular. Therefore, it is convenient to incorporate mechanisms that control the regularity of the matrix when applying any mathematical method that requires its inversion, as is the case of QML.

The QML method consists of two steps. In the first, the power of the anisotropies in pixel space is transformed to harmonic space. All the information of cosmological interest is already encoded in this space. In the second, operating entirely in harmonic space, one goes from an expression in which the powers of the six components of the spectrum and of the different scales appear mixed, to the expression in the form of a power spectrum. To undo the mixing, one needs to calculate the inverse of the Fisher matrix. But, again, the Fisher matrix is not regular per se. When this happens, one cannot calculate the multipole-to-multipole power spectrum. Still, one can calculate the binned power spectrum. In principle, when binning, information can be lost, the result may not be unbiased and control of the covariance matrix is lost. In this Thesis, making use of the model that describes the a priori knowledge that we have of the statistical characteristics of the maps, we have developed a binning scheme that produces an unbiased and minimum variance estimate. In short, for a given set of bins, we have the best form of applying QML to obtain a binned spectra, but that does not mean that we have the best way to define an optimal set. As far as we know, there is no way to choose a priori the best binning configuration. But once we have defined a set of bins, we do know the best way to calculate the power in them. On the other hand, the fact that we can compute the power in the bins in an unbiased and minimum covariance way does not mean that we do not loose information: obviously, if we have the power spectrum and the covariances of a map, we can compute the binned power. But we cannot go from binned power to multipole-to-multipole power.

Equal in the quality of the results to the maximum likelihood method, the QML method has less computational workload, although it is still high. Among other things, it requires calculating the covariance matrix, its inverse and the Fisher matrix. The structure of the signal covariance matrix is much simpler in harmonic space than in pixel space. Taking advantage of this fact, the method can be implemented such as the heaviest calculations (the determination of the Fisher matrix) involve the use of matrices with a large number of zeros and a few ones. By analyzing in detail the role played by the location of these ones in the context of matrix multiplications, one can analytically predict the result to be obtained after complex and time-consuming matricidal mathematical operations. So we can predict the value of essential matrix elements in the method by significantly reducing the number of operations needed to find them. In this way, an implementation of the method can be developed that significantly saves the number of matrix operations and can also be parallelized when implementing it in computer language. Thus we have developed an implementation that reduces the number of operations by several orders of magnitude with respect to previous implementations of the method. Moreover, since the organization and structure of the computations is parallelizable, we have developed an implementation in `Fortran` language that takes advantage of pre-existing libraries optimized to perform unavoidable parallel matrix operations. On the other hand, the calculation of the result of the matrix operations that we save with our method has been implemented parallelized with our own algorithms. The result has been a parallel code that implements an optimal method with optimized algorithms —although probably still with some room for improvement—. Thus, QML can be applied relatively quickly to large problems, making use of the computational power and the huge amount of memory of supercomputers. The ECLIPSE code is a public version of our implementation, freely available on the Internet. In this Thesis we have incorporated the manual of use of the code in order to illustrate its ease of use (within the difficulty of providing all the necessary information to accurately determine a situation in which QML can be applied) and versatility.

In the tests that we have carried out to study the performance of QML, we have found it to be unbiased and of minimum variance when the fiducial model and the noise information we supply to the method matches those of the simulated maps, both when computing the full power spectrum and when doing so by applying the optimal binning technique developed in this work. If we are only interested in polarization, QML can be applied to estimate only the related components, reducing the number of operations by a factor of about two, keeping the results unbiased and with an uncertainty very similar to that found when calculating the full spectrum.

When the fiducial model does not match the signal on the maps, the method is shown to be unbiased and uncertainties increase. However, the binned method may introduce some bias in this case. This is an expected phenomenon, associated with an extra use of the fiducial information when binning. When applying an iterative map-to-map scheme, the estimates converge to the same values regardless of the starting fiducial. The above happens both when applied with and without binning; this shows that the method is robust. When applying the method to a very simple model for the estimation of the cosmological parameter r —tensor-to-scalar ratio— we have found the same behavior.

The pseudo-C_ℓ methods are unbiased and computationally much faster than the maximum likelihood methods. However, we found that they produce estimates with significantly larger uncertainties, especially in the region of low ℓ, which is critical for the detection of tensor modes. Because reducing aliasing requires apodizing the maps, they are much more sensitive to the existence of sparse areas where signal is missing.

Because QML operates with the maps in pixel space, it is not affected by such problems; missing a few pixels hardly translates into increased error bars.

In short, in this work we present a detailed study of the possibilities and results of estimating the CMB power spectrum using a quadratic maximum likelihood method. For this purpose, we have developed a mathematical formulation that significantly reduces the number of operations and we have written a code that implements it. The code is freely available and we hope that it will become a useful and easy-to-use tool for the scientific community.

APPENDIX A

STUDY OF THE DETERMINANT OF C

According to eq. (3.11), we can expand the determinant of **C** as a sum of determinants of matrices whose columns are the spherical harmonics calculated at the different pixels of the image. If one determinant in the sum is not null, there will be other $n! - 1$ terms that are not zero, corresponding to the permutations of its columns. However, in principle, it could happen that the sum of all these elements is zero. In this appendix we will calculate this sum and show that the result is a real, positive number.

Let us suppose that we have a particular collection of values $\{\mu_1, \mu_2, \ldots, \mu_n\}$ such that the corresponding determinant in eq. (3.11) is not null; of course, the indices must be different and correspond to a collection of linearly independent spherical harmonics. For the sake of simplicity, let us reassign labels to the spherical harmonics in such a way that now Y_{μ_i} is labeled as Y_i, so we have a matrix whose n columns are the n spherical harmonics $Y_i, i = 1, \ldots n$ evaluated in the n pixels of the map. Let us define D and **M**

$$D \equiv \det(\mathbf{M}) \equiv \det \begin{pmatrix} Y_{11} & Y_{21} & \cdots & Y_{n1} \\ Y_{12} & Y_{22} & \cdots & Y_{n2} \\ \vdots & \vdots & & \vdots \\ Y_{1n} & Y_{2n} & \cdots & Y_{nn} \end{pmatrix}. \tag{A.1}$$

In the concatenated summations in eq. (3.11) there are another $n!-1$ no null terms given by permutations of the columns of the matrix in eq. (A.1), whose determinants take values $\pm D$ depending on the signature of the permutation. Let S be the contribution to the determinant of **C** of the terms coming from those permutations, let P_n be the collection of permutations of n elements, and let us also relabel C_{μ_i} in eq. (3.11) as C_i.

The summation extends over the permutations P_n, and

$$\begin{aligned} S &= \sum_{\sigma\in P_n}\left[\prod_{i=1}^{n} C_i\right]\left[\prod_{i=1}^{n} Y^*_{\sigma_i i}\right]\mathrm{sign}\,(\sigma)\,D \\ &= D\left[\prod_{i=1}^{n} C_i\right]\det\left(\mathbf{M}^*\right) \\ &= \left[\prod_{i=1}^{n} C_i\right]|D|^2. \end{aligned} \tag{A.2}$$

Note that in order to get to the second line of the equation, we have taken out of the sum the constant D as well as the factors coming from the power spectrum that multiplies each permutation in eq. (3.11), since these factors only depend on the spherical harmonics in the permutation and not on its order. In this way we have identified the remaining sum as Leibniz's formula for the determinant of $\mathbf{M}^*$. Taking into account that determinant and conjugate commute, we get to the final expression of the previous equation.

Therefore, assuming that there are enough non-zero C_ℓ's, if we are able to find a collection of spherical harmonics such that $D \neq 0$, we have

$$\det(\mathbf{C}) > 0. \tag{A.3}$$

Finally, let us calculate the full value of the determinant of $\mathbf{C}$. If we consider n points on the sphere and N spherical harmonics, we can find as many different sets of n spherical harmonics as the number of n-combinations of an N-set. Since the expression (A.2) gives the generic contribution to the determinant of $\mathbf{C}$ of a set of n harmonics, if we call σ to one of the combinations, and S_σ to the sum in eq. (A.2) for this combination, then

$$\det(\mathbf{C}) = \sum_{\sigma} S_\sigma = \sum_{\sigma}\left(\prod_{i=1}^{n} C_{\sigma_i}\right)|D_\sigma|^2. \tag{A.4}$$

APPENDIX B

SYMMETRIES ON THE SPHERICAL HARMONICS MATRIX

B.1 S_I symmetry: $\hat{\boldsymbol{r}} \rightarrow -\hat{\boldsymbol{r}}$

The spherical harmonics satisfy

$$Y_{\ell m}(-z, \phi + \pi) = (-1)^{\ell} Y_{\ell m}(z, \phi). \tag{B.1}$$

If for each pixel with coordinates (z, ϕ) there exists a pixel with coordinates $(-z, \phi + \pi)$, the matrix in eq. (3.13) simplifies notably. This can be easily seen considering the simple case with $\ell = 1$ and four pixels

$$\mathbf{Y} = \begin{pmatrix} Y_1^{-1}(\hat{r}_1) & Y_1^{0}(\hat{r}_1) & Y_1^{1}(\hat{r}_1) \\ Y_1^{-1}(\hat{r}_2) & Y_1^{0}(\hat{r}_2) & Y_1^{1}(\hat{r}_2) \\ Y_1^{-1}(\hat{r}_3) & Y_1^{0}(\hat{r}_3) & Y_1^{1}(\hat{r}_3) \\ Y_1^{-1}(\hat{r}_4) & Y_1^{0}(\hat{r}_4) & Y_1^{1}(\hat{r}_4) \end{pmatrix}. \tag{B.2}$$

If the pixels are opposed pairwise, $\hat{r}_3 = -\hat{r}_1$ y $\hat{r}_4 = -\hat{r}_2$, then the matrix of eq. (B.2) is a rank 2 matrix, since we have only two linearly independent rows

$$\mathbf{Y} = \begin{pmatrix} Y_1^{-1}(\hat{r}_1) & Y_1^{0}(\hat{r}_1) & Y_1^{1}(\hat{r}_1) \\ Y_1^{-1}(\hat{r}_2) & Y_1^{0}(\hat{r}_2) & Y_1^{1}(\hat{r}_2) \\ -Y_1^{-1}(\hat{r}_1) & -Y_1^{0}(\hat{r}_1) & -Y_1^{1}(\hat{r}_1) \\ -Y_1^{-1}(\hat{r}_2) & -Y_1^{0}(\hat{r}_2) & -Y_1^{1}(\hat{r}_2) \end{pmatrix}. \tag{B.3}$$

With the help of this property we can notably simplify the matrix **Y**. But let us first define an operator that will simplify the notation in the coming expression. Suppose that we have selected a certain pixel collection, indexed from 1 to k, and a set of harmonics, represented by Y_{LM}, that contains U elements indexed from 1 to U. The operator $\otimes$ constructs the block obtained by applying all the harmonics of the set on the selected pixels

$$Y_{LM} \otimes \begin{pmatrix} \hat{r}_1 \\ \vdots \\ \hat{r}_k \end{pmatrix}_{k\times U} \equiv \begin{pmatrix} Y_{\ell_1 m_1}(\hat{r}_1) & \cdots & Y_{\ell_U m_U}(\hat{r}_1) \\ \vdots & & \vdots \\ Y_{\ell_1 m_1}(\hat{r}_k) & \cdots & Y_{\ell_U m_U}(\hat{r}_k) \end{pmatrix}. \tag{B.4}$$

Going back to $\mathbf{Y}$ and considering the case of n pixels in the pixelization that fully accomplish the S_I symmetry, $\hat{r}_{i+n/2} = -\hat{r}_i$, $i = 1 \ldots n/2$, if we reorder the spherical harmonics in such a way that the ones with even ℓ occupy the first columns and those with odd ℓ occupy the last columns, $\mathbf{Y}$ becomes

$$\mathbf{Y} = \begin{pmatrix} Y_{\ell_e} \otimes \begin{pmatrix} \hat{r}_1 \\ \vdots \\ \hat{r}_{n/2} \end{pmatrix}_{n/2\times N_e} & Y_{\ell_o} \otimes \begin{pmatrix} \hat{r}_1 \\ \vdots \\ \hat{r}_{n/2} \end{pmatrix}_{n/2\times N_o} \\ Y_{\ell_e} \otimes \begin{pmatrix} \hat{r}_{n/2+1} \\ \vdots \\ \hat{r}_n \end{pmatrix}_{n/2\times N_e} & Y_{\ell_o} \otimes \begin{pmatrix} \hat{r}_{n/2+1} \\ \vdots \\ \hat{r}_n \end{pmatrix}_{n/2\times N_o} \end{pmatrix}, \tag{B.5}$$

where ℓ_e and ℓ_o represent the set of spherical harmonics with even and odd ℓ respectively. The corresponding numbers of harmonics in each block are given by N_e and N_o (see eq. (3.17)).

Taking into account the symmetry of pixels and spherical harmonics, we have

$$\mathbf{Y} = \begin{pmatrix} Y_{\ell_e} \otimes \begin{pmatrix} \hat{r}_1 \\ \vdots \\ \hat{r}_{n/2} \end{pmatrix}_{n/2\times N_e} & Y_{\ell_o} \otimes \begin{pmatrix} \hat{r}_1 \\ \vdots \\ \hat{r}_{n/2} \end{pmatrix}_{n/2\times N_o} \\ Y_{\ell_e} \otimes \begin{pmatrix} \hat{r}_1 \\ \vdots \\ \hat{r}_{n/2} \end{pmatrix}_{n/2\times N_e} & -Y_{\ell_o} \otimes \begin{pmatrix} \hat{r}_1 \\ \vdots \\ \hat{r}_{n/2} \end{pmatrix}_{n/2\times N_o} \end{pmatrix}. \tag{B.6}$$

It is possible to substitute the first $n/2$ rows by the linear combination $\mathbf{r}_i \longleftarrow \frac{1}{2}(\mathbf{r}_i + \mathbf{r}_{i+n/2})$, $i = 1,\ldots,n/2$. Similarly, the second half of rows can be replaced by $\mathbf{r}_i \longleftarrow$

$\frac{1}{2}(\mathbf{r}_{i-n/2}-\mathbf{r}_i)$, $i=n/2+1,\ldots,n$. With this transformation, one finds

$$\mathbf{Y} \to \begin{pmatrix} Y_{\ell_e} \otimes \begin{pmatrix} \hat{r}_1 \\ \vdots \\ \hat{r}_{n/2} \end{pmatrix}_{n/2\times N_e} & \mathbf{0}_{n/2\times N_o} \\ \mathbf{0}_{n/2\times N_e} & Y_{\ell_o} \otimes \begin{pmatrix} \hat{r}_1 \\ \vdots \\ \hat{r}_{n/2} \end{pmatrix}_{n/2\times N_o} \end{pmatrix}. \tag{B.7}$$

Therefore, the rank of the matrix is equal to the sum of the ranks of the two diagonal blocks above. Each of these ranks is less or equal than the minimum between the number of rows ($\leq n/2$) and the number of columns (N_e or N_o), i.e.,

$$\text{rank}(\mathbf{Y}) \leq \min(N_e, n/2) + \min(N_o, n/2), \tag{B.8}$$

which corresponds to the constraint R_1 presented in Section 3.3.1.

B.2 S_{II} symmetry: $\phi \to \phi + \pi$

Another interesting property of the spherical harmonics is

$$Y_{\ell m}(z, \phi+\pi) = (-1)^m Y_{\ell m}(z, \phi). \tag{B.9}$$

Let us assume that we have a pixelization that satisfy the previous symmetry S_I. If for each point in the pixelization with coordinates (z, ϕ) there is another point with coordinates $(z, \phi+\pi)$, then it is possible to further simplify the matrix of eq. (B.7). In order to do this, one should first note that the non-zero blocks are evaluated only on the first half of the pixels. All pixels with $z \neq 0$ that are included in these blocks have their symmetric pair $\phi \to \phi + \pi$ also in the first half of pixels, which allows one to use symmetry S_{II} to reduce the size of the non-zero blocks. However, the case with $z = 0$ needs to be addressed separately. For these points, the corresponding $\phi \to \phi + \pi$ pixel satisfies also $\hat{r} \to -\hat{r}$, and therefore it has been removed out of the matrix during the transformations that led to the matrix of eq. (B.7). Therefore, for those positions with $z = 0$, the pixel symmetric with respect to the S_{II} symmetry is not present in the non-zero blocks and can not be used to further simplify the matrix of eq. (B.7).

Fortunately, there is another interesting property of the spherical harmonics that can be used when $z = 0$. If ℓ is even and m is odd, or if ℓ is odd and m is even, then

$Y_{\ell m}(z=0,\phi)=0$. Let us assume that when applying the symmetry S_I, we have kept the first $n/2$ pixels (those with $z>0$, and those with $z=0$ and $0\le\phi<\pi$). Let us now reorder the spherical harmonics (i.e., the columns of matrix $\mathbf{Y}$) in a sequence such that the values (ℓ,m) are arranged in the following combinations: (even, odd), (even, even), (odd, odd), and finally (odd, even). After this rearrangement, the even ℓ block of matrix of eq. (B.7) becomes

$$\mathbf{Y}_e=\begin{pmatrix} Y_{\ell_e m_o}\otimes\begin{pmatrix}\hat{r}_1\\ \vdots\\ \hat{r}_k\end{pmatrix}_{k\times N_{eo}} & Y_{\ell_e m_e}\otimes\begin{pmatrix}\hat{r}_1\\ \vdots\\ \hat{r}_k\end{pmatrix}_{k\times N_{ee}} \\ \mathbf{0}_{u\times N_{eo}} & Y_{\ell_e m_e}\otimes\begin{pmatrix}\hat{r}_{k+1}\\ \vdots\\ \hat{r}_{k+u}\end{pmatrix}_{u\times N_{ee}}\end{pmatrix}, \tag{B.10}$$

where we have also separated the rows corresponding to k pixels with $z>0$ from the u positions with $z=0$, thus $k+u=n/2$.

N_{eo} is the number of spherical harmonics with even ℓ and odd m, and N_{ee} is the number of spherical harmonics with even ℓ and even m. Using eq. (B.9) and dividing the k pixels with $z>0$ in two halves, one with $0\le\phi<\pi$ and another one with $\pi\le\phi<2\pi$, the block of eq. (B.10) can be transformed

$$\mathbf{Y}_e\to\begin{pmatrix} Y_{\ell_e m_o}\otimes\begin{pmatrix}\hat{r}_1\\ \vdots\\ \hat{r}_{k/2}\end{pmatrix}_{k/2\times N_{eo}} & \mathbf{0}_{k/2\times N_{ee}} \\ \mathbf{0}_{k/2\times N_{eo}} & Y_{\ell_e m_e}\otimes\begin{pmatrix}\hat{r}_1\\ \vdots\\ \hat{r}_{k/2}\end{pmatrix}_{k/2\times N_{ee}} \\ \mathbf{0}_{u\times N_{eo}} & Y_{\ell_e m_e}\otimes\begin{pmatrix}\hat{r}_{k+1}\\ \vdots\\ \hat{r}_{k+u}\end{pmatrix}_{u\times N_{ee}}\end{pmatrix}, \tag{B.11}$$

where we have operated in an analogous way to that used in the transformation that led to the matrix of eq. (B.7). The odd ℓ block of this matrix can also be simplified in the same manner, until **Y** is finally reduced to a four diagonal block matrix. The total rank is the sum of the ranks of the four blocks

$$\begin{aligned}\operatorname{rank}(\mathbf{Y}) \leq \min(N_{eo}, k/2) + \min(N_{ee}, k/2+u) \\ + \min(N_{oo}, k/2+u) + \min(N_{oe}, k/2),\end{aligned} \tag{B.12}$$

where N_{oo} y N_{oe} are the number of spherical harmonics with odd ℓ and m and with odd ℓ and even m, respectively.

B.3 S_{III} symmetry: $\phi \to \phi + \pi/2$

A third interesting symmetry relation of the spherical harmonics is

$$Y_{\ell m}(\theta, \phi + \pi/2) = i^m Y_{\ell m}(\theta, \phi). \tag{B.13}$$

Again, let us assume that we have a pixelization that satisfy the symmetries S_I and S_{II} and, therefore, we can transform the matrix **Y** into a matrix with four diagonal blocks as the one described in eq. (B.11). If, in addition, for each point in the pixelization with coordinates (z, ϕ) there is another point with coordinates $(z, \phi + \pi/2)$, it is then possible to subdivide each block into two diagonal sub-blocks. To do that, we need to rearrange the pixels and spherical harmonics properly, and take linear combinations similar to the ones described in Appendix B.2. We show as an example the transformation of the block of the spherical harmonics of even ℓ and odd m of eq. (B.11)

$$\mathbf{Y}_{eo} = Y_{\ell_e m_o} \otimes \begin{pmatrix} \hat{r}_1 \\ \vdots \\ \hat{r}_{k/2} \end{pmatrix}_{k/2 \times N_{eo}}, \tag{B.14}$$

into

$$\mathbf{Y}_{eo} \to \begin{pmatrix} Y_{\ell_e m_1} \otimes \begin{pmatrix} \hat{r}_1 \\ \vdots \\ \hat{r}_{k/4} \end{pmatrix}_{k/4 \times N_{e1}} & \mathbf{0}_{k/4 \times N_{e3}} \\ \mathbf{0}_{k/4 \times N_{e1}} & Y_{\ell_e m_3} \otimes \begin{pmatrix} \hat{r}_1 \\ \vdots \\ \hat{r}_{k/4} \end{pmatrix}_{k/4 \times N_{e3}} \end{pmatrix}. \tag{B.15}$$

N_{e1} and N_{e3} are the number of spherical harmonics with $i^m = i$ and $i^m = -i$, respectively, and m_1 and m_3 indicate the corresponding sets of indices. Note that, in this case, to obtain zero blocks the imaginary unit i has to be introduced in the linear combinations of rows. The block corresponding to the spherical harmonics with even ℓ and m in eq. (B.11), and those with odd ℓ can be similarly transformed. Finally, the matrix is divided into eight diagonal blocks whose total rank is constrained by eq. (3.24).

APPENDIX C

RANK EXPRESSION UNDER THE PRESENCE OF A MASK

In order to prove the constraint given by eq. (3.38), we start from the full matrix $\mathbf{Y}$

$$\mathbf{Y} = \begin{pmatrix} Y_{1,1} & \cdots & Y_{N,1} \\ \vdots & & \vdots \\ Y_{1,n} & \cdots & Y_{N,n} \end{pmatrix}. \tag{C.1}$$

Let us consider that we have n valid pixels allowed by the mask that correspond to n_p paired pixels (i.e., that fulfill the S_I symmetry) and n_u unpaired pixels, $n = n_p + n_u$. Let us also assume that there are N_e and N_o spherical harmonics of even and odd ℓ respectively. By properly ordering the spherical harmonics and the pixels, and after taking linear combinations of the n_p rows as we did in Appendix B.1, the matrix of eq. (C.1) can be converted into

$$\mathbf{Y} \rightarrow \begin{pmatrix} \mathbf{Y}^{even}_{n_p/2\times N_e} & \mathbf{0}_{n_p/2\times N_o} \\ \\ \mathbf{0}_{n_p/2\times N_e} & \mathbf{Y}^{odd}_{n_p/2\times N_o} \\ \\ \mathbf{Y}^{even}_{n_u\times N_e} & \mathbf{Y}^{odd}_{n_u\times N_o} \end{pmatrix} \equiv \begin{pmatrix} \mathbf{Y}^e_p & \mathbf{0} \\ \\ \mathbf{0} & \mathbf{Y}^o_p \\ \\ \mathbf{Y}^e_u & \mathbf{Y}^o_u \end{pmatrix}. \tag{C.2}$$

If the rank of the block $\mathbf{Y}^e_p$ is R_e and the one of $\mathbf{Y}^o_p$ is R_o, we have

$$R_e \leq \min(n_p/2, N_e), \qquad R_o \leq \min(n_p/2, N_o), \tag{C.3}$$

since the rank of each block has to be less or equal than the minimum between the number of rows and the number of columns.

Taking linear combinations, the blocks $\mathbf{Y}^e_p$ and $\mathbf{Y}^o_p$ of the previous matrix can be transformed into diagonal blocks with as many no null elements as their respective ranks, R_e and R_o. Showing it explicitly for $\mathbf{Y}^e_p$, and substituting $\mathbf{Y}^o_p$ by its diagonal form

$\mathbf{D}^o_p$, we have

$$\mathbf{Y} \rightarrow \begin{pmatrix} \begin{matrix} d^e_{11} & 0 & \cdots & 0 \\ 0 & d^e_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d^e_{R_eR_e} \end{matrix} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \\ \mathbf{0} & & \begin{matrix} \mathbf{D}^o_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{matrix} \\ \mathbf{Y}^e_u & & \mathbf{Y}^o_u \end{pmatrix}. \tag{C.4}$$

Note that some of the zero blocks around $\mathbf{D}^e_p$ and $\mathbf{D}^o_p$ will not appear if the equality is fulfilled in eq. (C.3).

Furthermore, using the diagonal blocks, we can obtain R_e zeros in the $\mathbf{Y}^e_u$ block and R_o zeros in the $\mathbf{Y}^o_u$ block. Therefore, the matrix can be further simplified

$$\mathbf{Y} \rightarrow \begin{pmatrix} \begin{matrix} \mathbf{D}^e_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{matrix} & \mathbf{0} \\ \mathbf{0} & \begin{matrix} \mathbf{D}^o_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{matrix} \\ \begin{matrix} \mathbf{0} & \mathbf{Y}'^e_u \end{matrix} & \begin{matrix} \mathbf{0} & \mathbf{Y}'^o_u \end{matrix} \end{pmatrix}, \tag{C.5}$$

where the $\mathbf{Y}'^e_u$ and $\mathbf{Y}'^o_u$ blocks have $N_e - R_e$ columns and $N_o - R_o$ columns, respectively. Finally, reordering the rows and columns, we obtain

$$\mathbf{Y} \rightarrow \begin{pmatrix} \mathbf{D}^e_p & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^o_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Y}'^e_u & \mathbf{Y}'^o_u \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{C.6}$$

Inspecting this matrix, we can easily derive the constraint given in eq. (3.36). For the rank($\mathbf{C}$) to be maximum, i.e., rank($\mathbf{C}$) = rank($\mathbf{Y}$) = n, $\mathbf{Y}$ needs to have n independent rows. Therefore, in this case, we would get a matrix like the one of eq. (C.6) but without the last row of zero blocks. Moreover, given that the block $\mathbf{D}^e_p$ has $n_p/2$ rows, it must also have at least $n_p/2$ linear independent columns in order to achieve the maximum rank. Since all of its columns are constructed in terms of spherical harmonics of even ℓ, then $N_e \geq n_p/2$ must be satisfied. Similarly, the same reasoning applies to block $\mathbf{D}_p^o$, finding $N_o \geq n_p/2$.

Let us now study the rank of the matrix in eq. (C.6) for any $\ell_{\max}$. Its rank is the sum of the ranks of the blocks $\mathbf{D}_p^e$, $\mathbf{D}_p^o$ and $(\mathbf{Y'}_u^e\,\mathbf{Y'}_u^o)$. Taking into account eq. (C.3) and since the block $(\mathbf{Y'}_u^e\,\mathbf{Y'}_u^o)$ has n_u rows and $L^* = N_e - R_e + N_o - R_o$ columns, we get the limit

$$\begin{aligned}\operatorname{rank}(\mathbf{Y}) &\leq R_e + R_o + \min(n_u, L^*)\\ &\leq \min(n_p/2, N_e) + \min(n_p/2, N_o) + \min(n_u, L^*).\end{aligned} \tag{C.7}$$

However this expression depends through L^* on the ranks R_e and R_o which are, in principle, unknown. It would be more convenient to have a more general constraint which did not depend on these ranks. With this aim, let us consider a new parameter, L, defined as

$$L = N_e - \min(n_p/2, N_e) + N_o - \min(n_p/2, N_o). \tag{C.8}$$

Taking into account eq. (C.3), it becomes apparent that $L^* \geq L$ and $\min(n_u, L^*) \geq \min(n_u, L)$. Therefore, it is not obvious whether the expression (C.7) is still valid when replacing L^* by L (except in the trivial case $L^* = L$).

Let us study in more detail the case $L^* > L$. We suppose $L^* = L + t$, with $t \geq 1$. In this case, R_e or R_o (or both) are lower than their possible maximum value given by the geometrical dimensions of the corresponding blocks. For the sake of simplicity, and without loss of generality, let us assume that $R_e = \min(n_p/2, N_e) - t$ (note that if both R_e and R_o are lower than their maximum geometrical value, the same reasoning would be done twice, with t_e and t_o, $t = t_e + t_o$). In this situation, we have

$$\begin{aligned}\operatorname{rank}(\mathbf{Y}) &\leq R_e + R_o + \min(n_u, L^*)\\ &\leq \min(n_p/2, N_e) - t + \min(n_p/2, N_o) + \min(n_u, L^*).\end{aligned} \tag{C.9}$$

Let us now focus on the value of the last term, $\min(n_u, L^*)$, and consider the two possible complementary cases $n_u \geq L^*$ and $n_u < L^*$.

1. If $n_u \geq L^*$, then $\min(n_u, L^*) = L^* = L + t$, and $\min(n_u, L) = L$. Expression (C.9) becomes

$$\begin{aligned}\operatorname{rank}(\mathbf{Y}) &\leq \min(n_p/2, N_e) - t + \min(n_p/2, N_o) + L + t\\ &= \min(n_p/2, N_e) + \min(n_p/2, N_o) + \min(n_u, L).\end{aligned} \tag{C.10}$$

2. Conversely, if $n_u < L^*$, expression (C.9) becomes

$$\operatorname{rank}(\mathbf{Y}) \leq \min(n_p/2, N_e) + \min(n_p/2, N_o) + n_u - t. \tag{C.11}$$

   Since $L < L^*$, if $n_u < L^*$ we have again two complementary possibilities: $n_u \leq L$ and $n_u > L$.

   (a) Let us first suppose that $n_u \leq L$, which implies $\min(n_u, L) = n_u$. In this case, from eq. (C.11) we have

$$\begin{aligned}\operatorname{rank}(\mathbf{Y}) &< \min(n_p/2, N_e) + \min(n_p/2, N_o) + n_u \\ &= \min(n_p/2, N_e) + \min(n_p/2, N_o) + \min(n_u, L).\end{aligned} \tag{C.12}$$

   (b) Finally let us consider the case $n_u < L^*$ and $n_u > L$. Since $L < n_u < L^* = L + t$, then $L - t < n_u - t < L$ and $\min(n_u, L) = L$. Therefore we can replace $n_u - t$ by L in expression (C.11) if we substitute the sign $\leq$ by $<$

$$\begin{aligned}\operatorname{rank}(\mathbf{Y}) &< \min(n_p/2, N_e) + \min(n_p/2, N_o) + L \\ &= \min(n_p/2, N_e) + \min(n_p/2, N_o) + \min(n_u, L).\end{aligned} \tag{C.13}$$

Since we have covered all the possible cases, this shows that we can write an upper limit on the rank of the masked case which does not depend on R_e and R_0 as

$$\operatorname{rank}(\mathbf{Y}) \leq \min(n_p/2, N_e) + \min(n_p/2, N_o) + \min(n_u, L). \tag{C.14}$$

which proves the validity of the constraint (3.38) given in Section 3.7.

APPENDIX D

EFFECT OF SAMPLING VARIANCE ON THE POWER SPECTRUM ERROR

It is well known that, due to sampling variance, the error of the estimated power spectrum increases as the observed sky fraction ($f_{\rm sky}$) is reduced. Its effect can be roughly approximated by an increment in the variance by a factor $1/f_{\rm sky}$ [106]. However, it is also well known that, in practice, the specific geometry of the considered mask will also affect this error. Although for our purpose (constructing the optimal binning) this approximation is sufficient, it is still interesting to perform some simple tests in order to understand the limitations of this assumption.

As an illustration, we have studied how the errors degrade with the considered sky fraction in different scenarios, in particular for the four masks shown in figure D.1. The errors have been obtained from the Fisher matrix. A resolution of $N_{\rm side} = 32$ and a binning of $\Delta_\ell = 3$ have been considered in all the cases. Noise has also been included according to the ground-based or space configurations levels, as it corresponds. Three of the considered masks correspond to three different geometries but with the same sky coverage as that of the ground-based scenario, that allows for an observed sky fraction of around 8.4 per cent (1028 pixels at $N_{\rm side} = 32$ resolution). Specifically, the first mask (top left panel of figure D.1) corresponds to that of the ground-based experiment used along the Chapter 6. A second toy-model mask contains the observed pixels distributed uniformly over the sky (top right panel of figure D.1), and we will refer to it as the uniform mask. A third mask with pixels distributed in the poles (bottom left panel of figure D.1) has also been added, referred to as poles mask; although this mask is not expected for a ground-based experiment, it allows us to check the geometrical effect of an (extreme) Galactic mask and to compare it with the other configurations. Finally, we have considered the case of the satellite-based experiment (bottom right panel of figure D.1) at a resolution of $N_{\rm side} = 32$ ($f_{\rm sky}$=0.58).

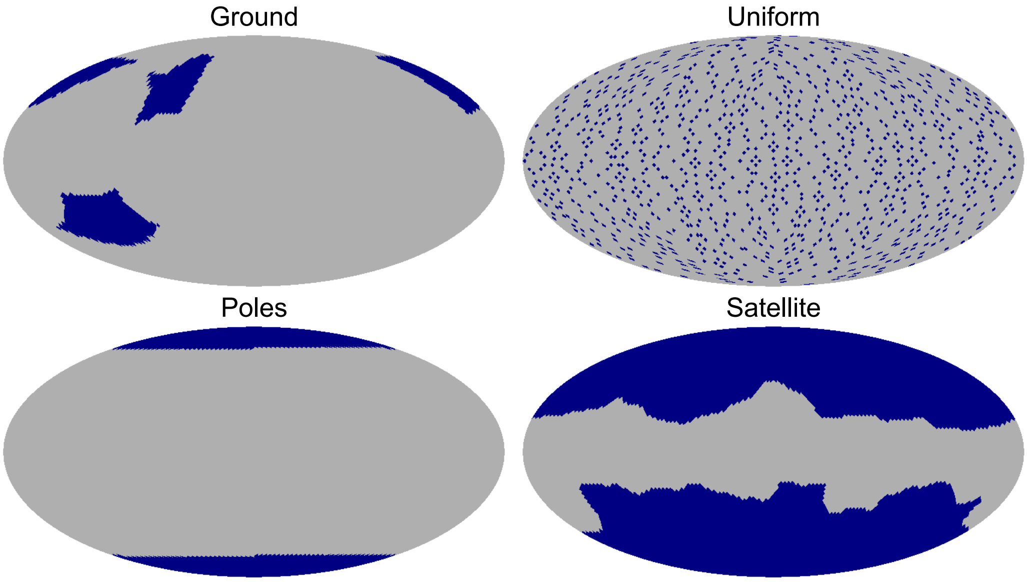


Figure D.1: Masks used to test how the error of the estimated power spectra depends on the geometry and fraction of the observed sky. All masks are shown at $N_{\rm side} = 32$. Top-left: ground-based mask. Top-right: toy-model mask with pixels distributed uniformly over the sky. Bottom-left: toy-model mask with the observed pixels distributed on the poles. Bottom-right: mask for a satellite-based experiment. The first three masks have a total of 1028 valid pixels, corresponding to a sky fraction of around 8.4 per cent, while $f_{\rm Sky}$=0.58 for the last mask. Note that the ground and satellite masks correspond to those of figure 6.1, but at a resolution of $N_{\rm side} = 32$, and are repeated here for easiness of reading.

Figure D.2 shows the ratio of the error on the estimation of the power spectrum for the four masks using the Fisher matrix over the theoretical approximation (according to eqs. (4-11) from [93]). Therefore values close to one indicate that the theoretical expression is a good approximation, whereas larger (smaller) values indicate an under- (over-) estimation of the error. As seen, for the uniform case (blue line), the estimated errors are smaller than those given by the theoretical expectation at large scales, while they explode at larger multipoles. This is expected since to estimate the power spectra at the largest scales, it is only necessary to provide data on a limited number of pixels but conveniently distributed, while this configuration can not provide the required information at small scales. It is also worth noting that the range of scales where this mask provides better results than the theoretical ones also varies significantly with the considered component of the power spectrum, working worse especially for BB. The ground (orange line) and poles (green line) masks have a qualitatively similar behav-

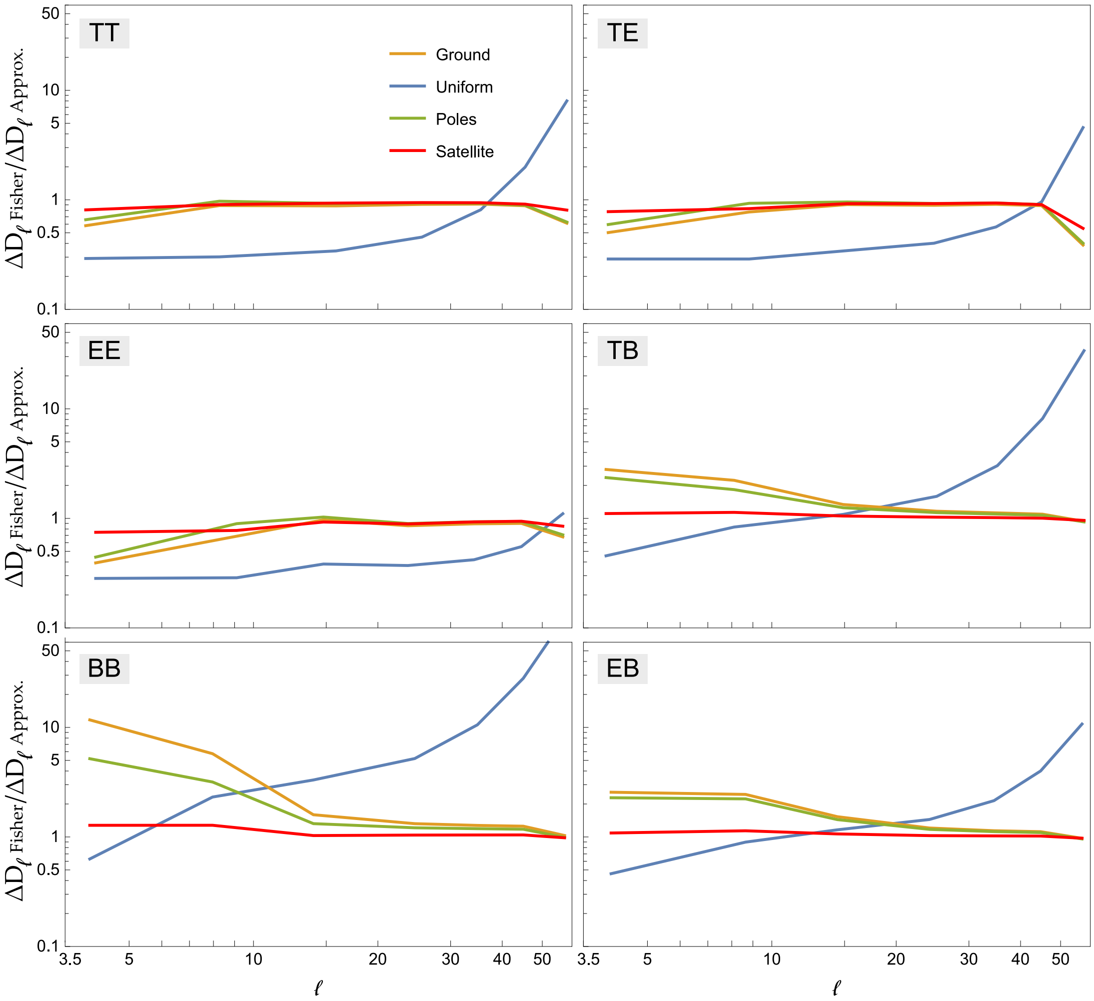


Figure D.2: Ratio between the error of the estimated power spectrum obtained with QML from the Fisher matrix, $\Delta D_{\ell\,\mathrm{Fisher}}$, over that of the theoretical approximation [93], $\Delta D_{\ell\,\mathrm{Approx.}}$, for the different masks shown in figure D.1.

ior. For TT, EE and TE, the estimated error is better than the theoretical expectation at low multipoles, while at small scales the error is underestimated by the naive f_{sky} scaling. The minimum observed at TE at the smallest scales is due to the fact that the TE spectrum becomes zero at that multipole. For BB, TB and EB, the error estimated by the Fisher matrix is larger than the approximation at all scales, with a particular large deviation for the large scales of BB. The theoretical approximation is working better for the Galactic mask in the space configuration (red line), finding only relatively small deviations between both estimations.

Therefore, if a precise estimation of the error introduced by the sky fraction is needed, it is advisable to carry out a full analysis that takes into account the effect of the geometry of the mask, especially when a small fraction of the sky is considered, as is the case in most ground-based experiments.

APPENDIX E

SMOOTHING FUNCTION FOR THE ITERATIVE QML

Although the power spectra given by a physical model will follow a smooth curve, a particular realization of the spectra and its corresponding estimation will in general be noisy. In fact, if we take directly the power spectra estimated with QML as the fiducial model for a subsequent iteration, this process may be unstable and does not necessarily lead to convergence. Therefore, it becomes necessary to apply some kind of smoothing to the estimated power spectra in order to provide a suitable fiducial model to the next iteration. An obvious choice would be to fit the estimated spectra to a cosmological model, and use that as our initial guess for the next step. This has been the approach used in Section 6.5.3, where we have considered a simplified case where all cosmological parameters are assumed to be known except the tensor-to-scalar ratio r. However, in a more general case, it could be computationally very costly to fit all the cosmological parameters and a simpler approach would be more convenient. With this aim we have constructed a smoothed version q_ℓ of each component of the estimated power spectra D_ℓ by minimizing the following function

$$\delta = \sum_{\ell=2}^{\ell_{\max}-1} L_\ell M_\ell \,, \tag{E.1}$$

where L_ℓ and M_ℓ are given by

$$L_\ell = \frac{(q_{\ell+1} - q_\ell)^2}{\Delta D_{\ell+1}^2 + \Delta D_\ell^2} + 1\,, \tag{E.2}$$

$$M_\ell = w_\ell + \frac{1}{2}\left[\frac{(q_{\ell+1} - D_{\ell+1})^2}{\Delta D_{\ell+1}^2} + \frac{(q_\ell - D_\ell)^2}{\Delta D_\ell^2}\right], \tag{E.3}$$

and ΔD_ℓ is the error associated to the D_ℓ estimation. The aim behind this approach is to minimize the sum of the length of the segments that join a pair of consecutive points (controlled by the L_ℓ term) but penalizing large differences between the smoothed and estimated values of the power spectra (encoded in the M_ℓ factor). Note that the +1 term in the first factor avoids that the minimization defaults to a constant straight line (i.e. $q_\ell = \text{constant}$). w_ℓ are weights that control the relative importance between both effects, such that larger values of the weights would tend to produce smoother curves and vice versa, allowing us to modify the level of required smoothing. In particular $w_\ell = 0$ would lead to the solution $q_\ell = D_\ell$, i.e., no smoothing. Note that a particular choice of the weights could still leave to an unstable iterative process for a given map in which case the smoothing process to provide a fiducial model for the next step would need to be repeated with different weights, such that the required number of iterations could be achieved. For simplicity, in our case we have simply discarded from the analysis those simulations that do not converge.

For our results in Section 6.5.2, we have chosen weights in the range 0.5 to 7, depending on the considered component of the power spectrum and on the multipole (larger weights are assigned to higher multipoles). Although the choice of the weights is somewhat arbitrary, we do not expect that their specific values affect to our conclusions, since they are used only to provide an initial guess at each step of the iterative process, facilitating convergence. As an illustration, figure E.1 shows a collection of weights and the TE power spectrum estimated for one simulation (blue points) and its smoothed version using two different sets of weights $w_{1\ell}$ (orange line) and $w_{2\ell} = 2w_{1\ell}$ (black line). As expected, greater weights lead to a smoother curve. Although the results of Section 6.5.2 are presented with the $w_{1\ell}$ set of weights, we have tested that very similar results are achieved when using $w_{2\ell}$ instead.

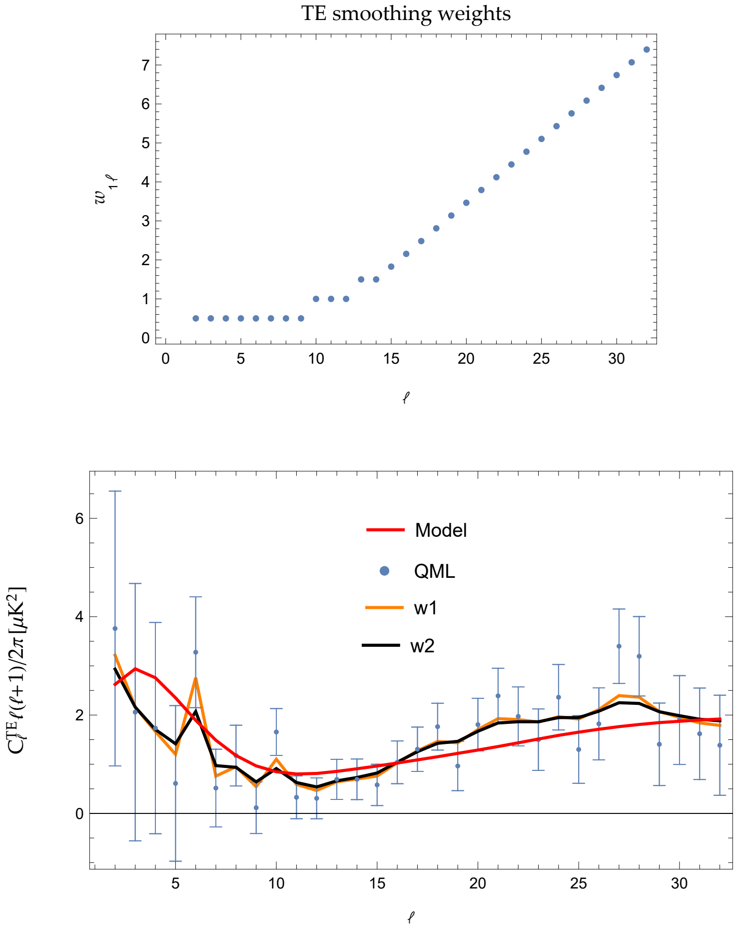


Figure E.1: TE smoothing weights (top) and power spectra (bottom) estimated from a simulation (blue points) with its corresponding error. Two smoothed versions of the spectrum are plotted obtained with weights $w_{1\ell}$ (orange line) and $w_{2\ell} = 2w_{1\ell}$ (black line). For comparison, the fiducial model used to generate the simulation is also given (red line).

APPENDIX F

ESTIMATOR OF THE TENSOR-TO-SCALAR RATIO

In order to evaluate the performance of the QML technique in some of the considered cases, it is useful to estimate the value of the tensor-to-scalar ratio from the recovered power spectra. This will be done in a simplified case, where we assume that all cosmological parameters are known but r. In this way, we can write the fiducial power spectra as a function of r

$$\mathbf{c}(r) = \mathbf{c}_S + r\mathbf{c}_T, \tag{F.1}$$

where $\mathbf{c}$ is the vector containing the six different components (temperature and polarization) of the power spectra. $\mathbf{c}_S$ and $\mathbf{c}_T$ are the scalar and tensor contributions, including lensing effects, which are fixed for a given fiducial model except for the value of r.

For a given map, let us assume that we have obtained with the QML an estimation of the power spectra ($\hat{\mathbf{c}}$) and of the corresponding Fisher matrix ($\mathbf{F}$). If we assume that the variables $\mathbf{c}(r)$ follow a Gaussian distribution,[1] the probability density function of $\hat{\mathbf{c}}$ can be written as a multivariate normal distribution of mean $\mathbf{c}(r)$ and covariance matrix $\mathbf{F}^{-1}$. Therefore, we can obtain $\hat{r}$ by maximizing the likelihood

$$\log L = -\frac{1}{2}(\hat{\mathbf{c}} - \mathbf{c}(r))^t \mathbf{F}(\hat{\mathbf{c}} - \mathbf{c}(r)) + \log((2\pi)^{6(1-\ell_{\max})}|\mathbf{F}|^{1/2}). \tag{F.2}$$

Since we are fitting r to the estimated power spectrum, our goal is to minimize the quantity

$$\chi^2 = (\hat{\mathbf{c}} - \mathbf{c}(r))^t \mathbf{F}(\hat{\mathbf{c}} - \mathbf{c}(r)). \tag{F.3}$$

This χ^2 variable is a polynomial of degree two, thus it can be maximized analytically to obtain

$$\hat{r} = \frac{\mathbf{c}_T^t \mathbf{F}(\hat{\mathbf{c}} - \mathbf{c}_S)}{\mathbf{c}_T^t \mathbf{F}\mathbf{c}_T}. \tag{F.4}$$

---

[1] It is well known that the C_ℓ's actually follow a χ^2 distribution with $2\ell + 1$ degrees of freedom and, therefore, the Gaussian approximation improves as ℓ increases. However, in our case, this approximation is sufficient to estimate r and, therefore, enough for our purpose.

The analytical expression of the error on the estimation (see eq. (1.92)) is

$$(\Delta r)^2 = -\left(\frac{d^2 \ln L}{dr^2}\right)^{-1} = (\mathbf{c}_T^t \mathbf{F} \mathbf{c}_T)^{-1}. \tag{F.5}$$

It is interesting to note that since $\hat{\mathbf{c}} = \mathbf{F}^{-1}\mathbf{y}$, r can be expressed directly in terms of $\mathbf{y}$, that is

$$\hat{r} = \frac{\mathbf{c}_T^t \mathbf{y} - \mathbf{c}_T^t \mathbf{F} \mathbf{c}_S}{\mathbf{c}_T^t \mathbf{F} \mathbf{c}_T}. \tag{F.6}$$

This is useful when the fraction of the sky covered is small and the Fisher matrix becomes singular, since it would allow the estimation of r without inverting the Fisher matrix. This also shows that all the cosmological information is actually encoded into the coupled power spectrum vector $\mathbf{y}$.

As shown in Section 6.5.3, if the value of the estimated tensor-to-scalar ratio that minimizes χ^2 differs significantly from the value used for the fiducial model (to compute $\mathbf{C}$ and $\mathbf{F}$, and to estimate $\hat{\mathbf{c}}$), one can obtain an updated power spectrum from the estimated value $\hat{r}$, that can be used as input for a new QML step in an iterative scheme until convergence is achieved.

This method can also be used with the binned estimator described in Section 4. In particular, one would need to replace the quantities $\mathbf{F}$ and $\hat{\mathbf{c}}$ in eq. (F.4) by their analogous binned versions, described in eq. (4.16) and (4.18), respectively. The fiducial power spectra $\mathbf{c}_S$ and $\mathbf{c}_T$ should also be binned in the same way.

APPENDIX G

ECLIPSE USER GUIDE V1.02

The current ECLIPSE User Guide is shown, integrated into this document, in the following pages.

ECLIPSE User Guide

A fast Quadratic Maximum Likelihood Estimator for CMB intensity and polarization power spectra.

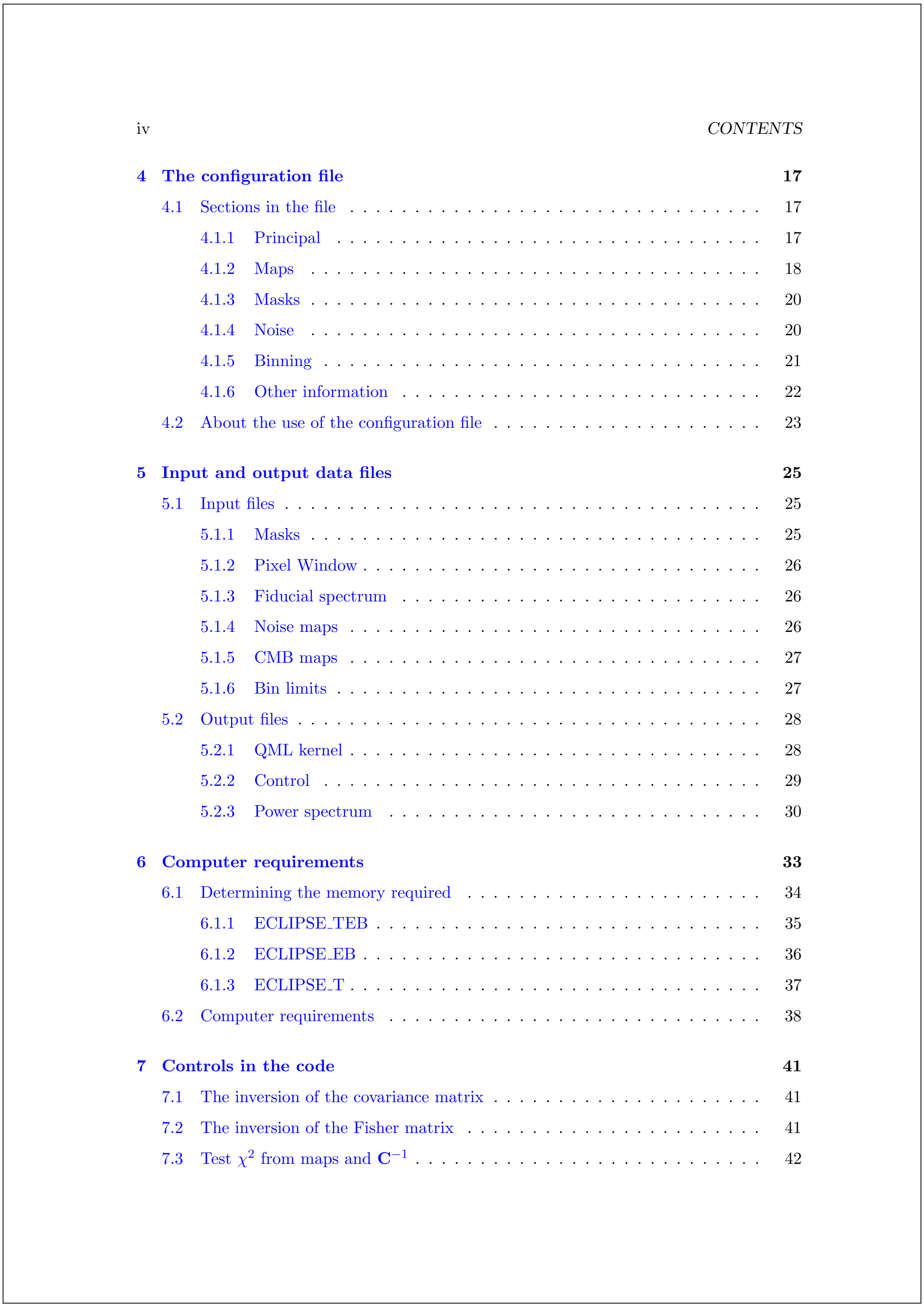
Version 1.02

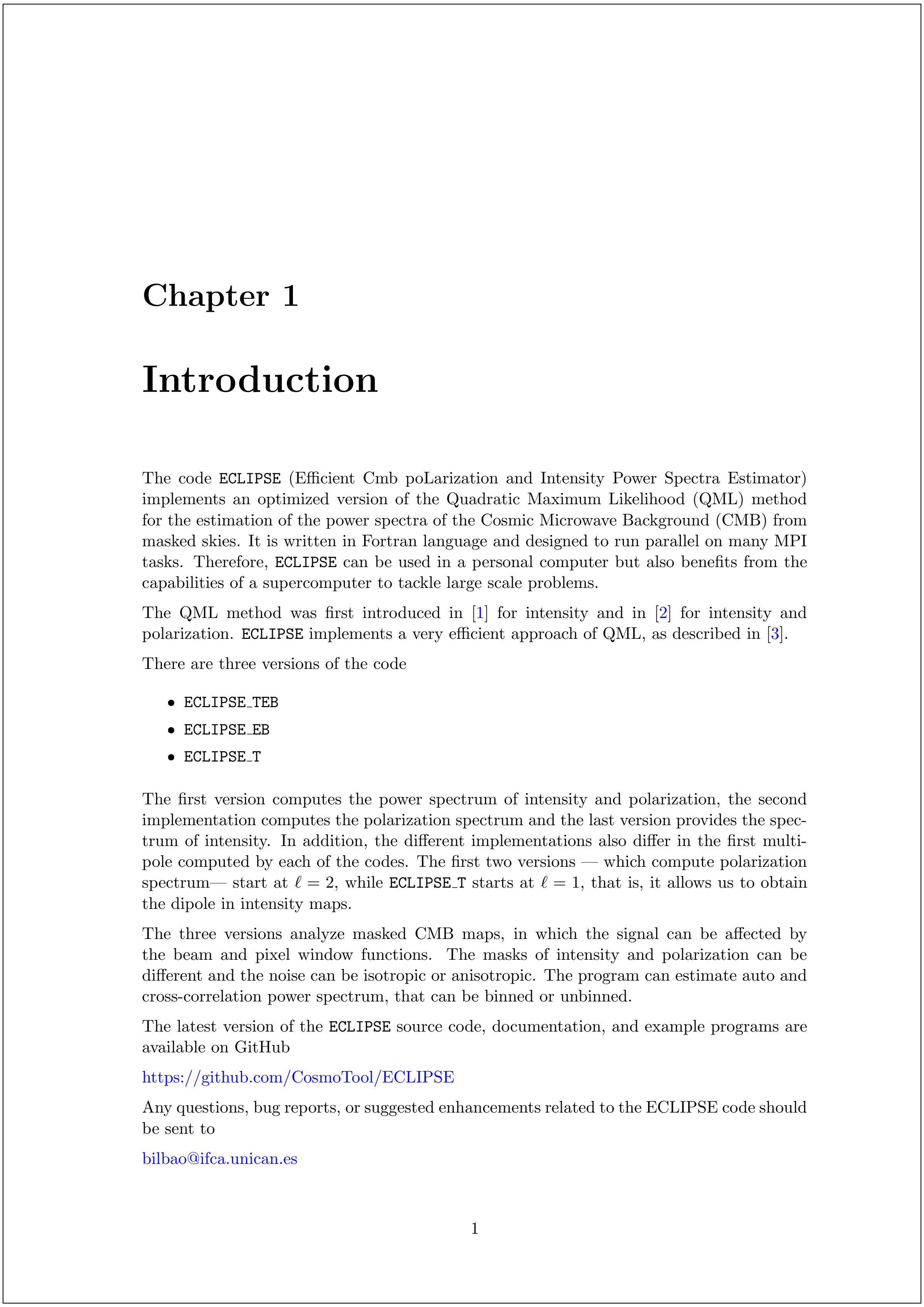
Juan Daniel Bilbao Ahedo

Instituto de Física de Cantabria
CSIC - Univ. de Cantabria
Edificio Juan Jorda
Avenida de los Castros
39005 Santander
Spain

May 2022

Contents





Chapter 1

Introduction

The code `ECLIPSE` (Efficient Cmb poLarization and Intensity Power Spectra Estimator) implements an optimized version of the Quadratic Maximum Likelihood (QML) method for the estimation of the power spectra of the Cosmic Microwave Background (CMB) from masked skies. It is written in Fortran language and designed to run parallel on many MPI tasks. Therefore, `ECLIPSE` can be used in a personal computer but also benefits from the capabilities of a supercomputer to tackle large scale problems.

The QML method was first introduced in [1] for intensity and in [2] for intensity and polarization. `ECLIPSE` implements a very efficient approach of QML, as described in [3].

There are three versions of the code

- `ECLIPSE_TEB`
- `ECLIPSE_EB`
- `ECLIPSE_T`

The first version computes the power spectrum of intensity and polarization, the second implementation computes the polarization spectrum and the last version provides the spectrum of intensity. In addition, the different implementations also differ in the first multipole computed by each of the codes. The first two versions — which compute polarization spectrum— start at $\ell = 2$, while `ECLIPSE_T` starts at $\ell = 1$, that is, it allows us to obtain the dipole in intensity maps.

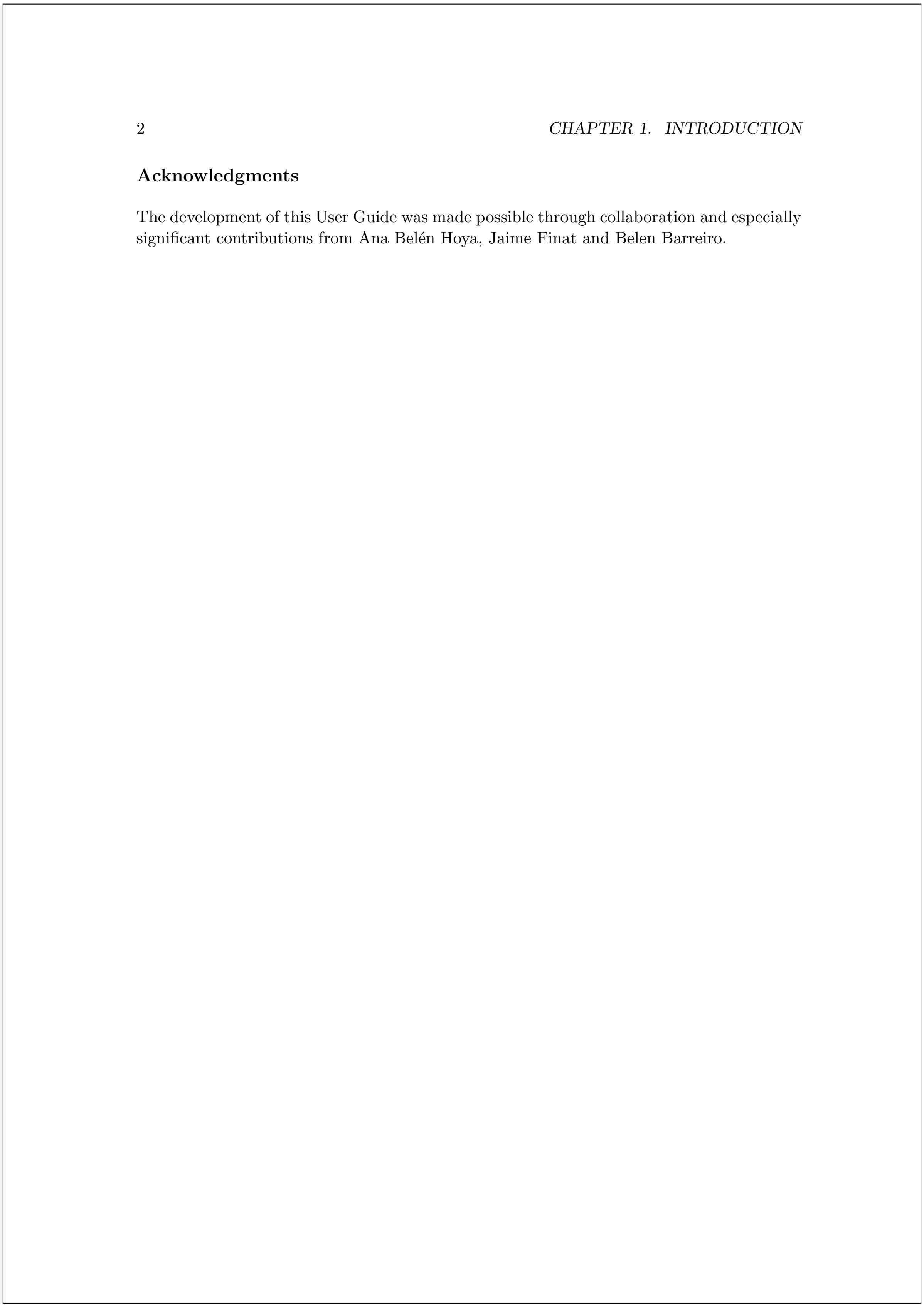
The three versions analyze masked CMB maps, in which the signal can be affected by the beam and pixel window functions. The masks of intensity and polarization can be different and the noise can be isotropic or anisotropic. The program can estimate auto and cross-correlation power spectrum, that can be binned or unbinned.

The latest version of the `ECLIPSE` source code, documentation, and example programs are available on GitHub

https://github.com/CosmoTool/ECLIPSE

Any questions, bug reports, or suggested enhancements related to the ECLIPSE code should be sent to

bilbao@ifca.unican.es



Acknowledgments

The development of this User Guide was made possible through collaboration and especially significant contributions from Ana Belén Hoya, Jaime Finat and Belen Barreiro.

Chapter 2

Building and running ECLIPSE

As an illustration, we will show in detail how to compile and run the code in a Linux personal computer as well as in the NERSC[1] and Altamira[2] supercomputers. Future versions of this document will incorporate additional systems.[3] Note that we do not intend to be exhaustive or to explore all possible options, but rather show some examples in which the code can be built and run. The essential point is that it meets the Fortran, ScaLAPACK, BLACS and FITSIO standards, so it can run in any machine and configuration compatible with them.

2.1 Building the code

The program is written in Fortran language and uses functions of the ScaLAPACK and BLACS libraries. In addition, it works with maps in FITS format. Thus the FITSIO[4] library is also needed.

Regarding ScaLAPACK, BLACS and MPI libraries, these are usually installed in supercomputers, as is the case of NERSC at Lawrence Berkeley National Laboratory and Altamira at IFCA. We offer detailed information about using `ECLIPSE` in both machines in this document. In order to build and run the code in a personal computer equipped with Linux operating system, the required set of libraries, compilers and instructions can be downloaded and installed through Intel® oneAPI Base Toolkit and the HPC Toolkit, among many other options.

2.1.1 Building ECLIPSE on a personal computer with Linux Operating System

We will show how to build `ECLIPSE` assuming that the GCC compiler is installed. To build the code the user can follow these steps

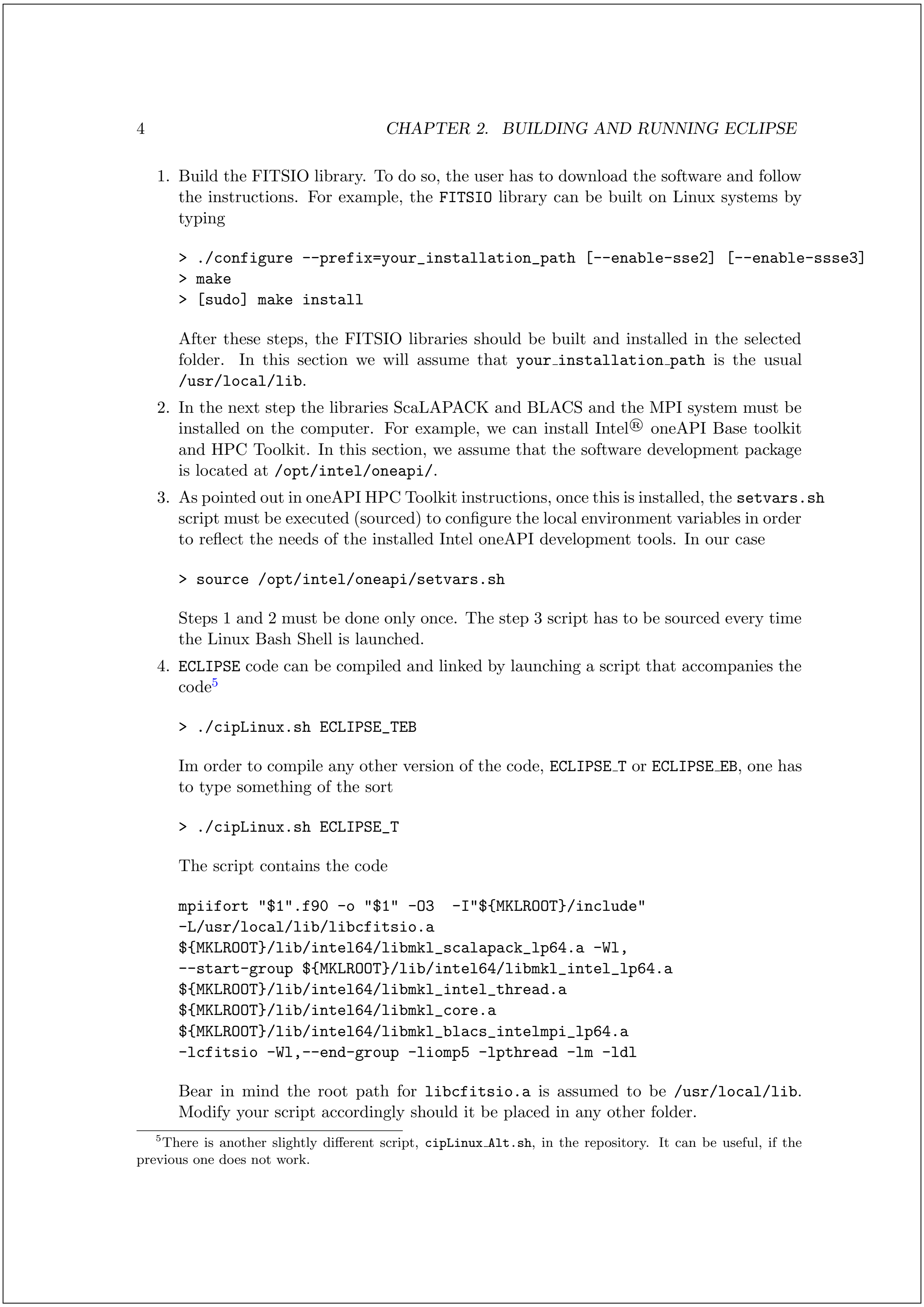
---

[1] NERSC

[2] Altamira

[3] To run it in a personal computer equipped with Windows operating system, one can install the Linux Subsystem for Windows and operate as if it were a Linux machine.

[4] The latest version of the CFITSIO source code, documentation, and example programs are available on: `http://heasarc.gsfc.nasa.gov/fitsio`



1. Build the FITSIO library. To do so, the user has to download the software and follow the instructions. For example, the `FITSIO` library can be built on Linux systems by typing

   ```
   > ./configure --prefix=your_installation_path [--enable-sse2] [--enable-ssse3]
   > make
   > [sudo] make install
   ```

   After these steps, the FITSIO libraries should be built and installed in the selected folder. In this section we will assume that `your_installation_path` is the usual `/usr/local/lib`.
2. In the next step the libraries ScaLAPACK and BLACS and the MPI system must be installed on the computer. For example, we can install Intel® oneAPI Base toolkit and HPC Toolkit. In this section, we assume that the software development package is located at `/opt/intel/oneapi/`.
3. As pointed out in oneAPI HPC Toolkit instructions, once this is installed, the `setvars.sh` script must be executed (sourced) to configure the local environment variables in order to reflect the needs of the installed Intel oneAPI development tools. In our case

   ```
   > source /opt/intel/oneapi/setvars.sh
   ```

   Steps 1 and 2 must be done only once. The step 3 script has to be sourced every time the Linux Bash Shell is launched.
4. `ECLIPSE` code can be compiled and linked by launching a script that accompanies the code[5]

   ```
   > ./cipLinux.sh ECLIPSE_TEB
   ```

   Im order to compile any other version of the code, `ECLIPSE_T` or `ECLIPSE_EB`, one has to type something of the sort

   ```
   > ./cipLinux.sh ECLIPSE_T
   ```

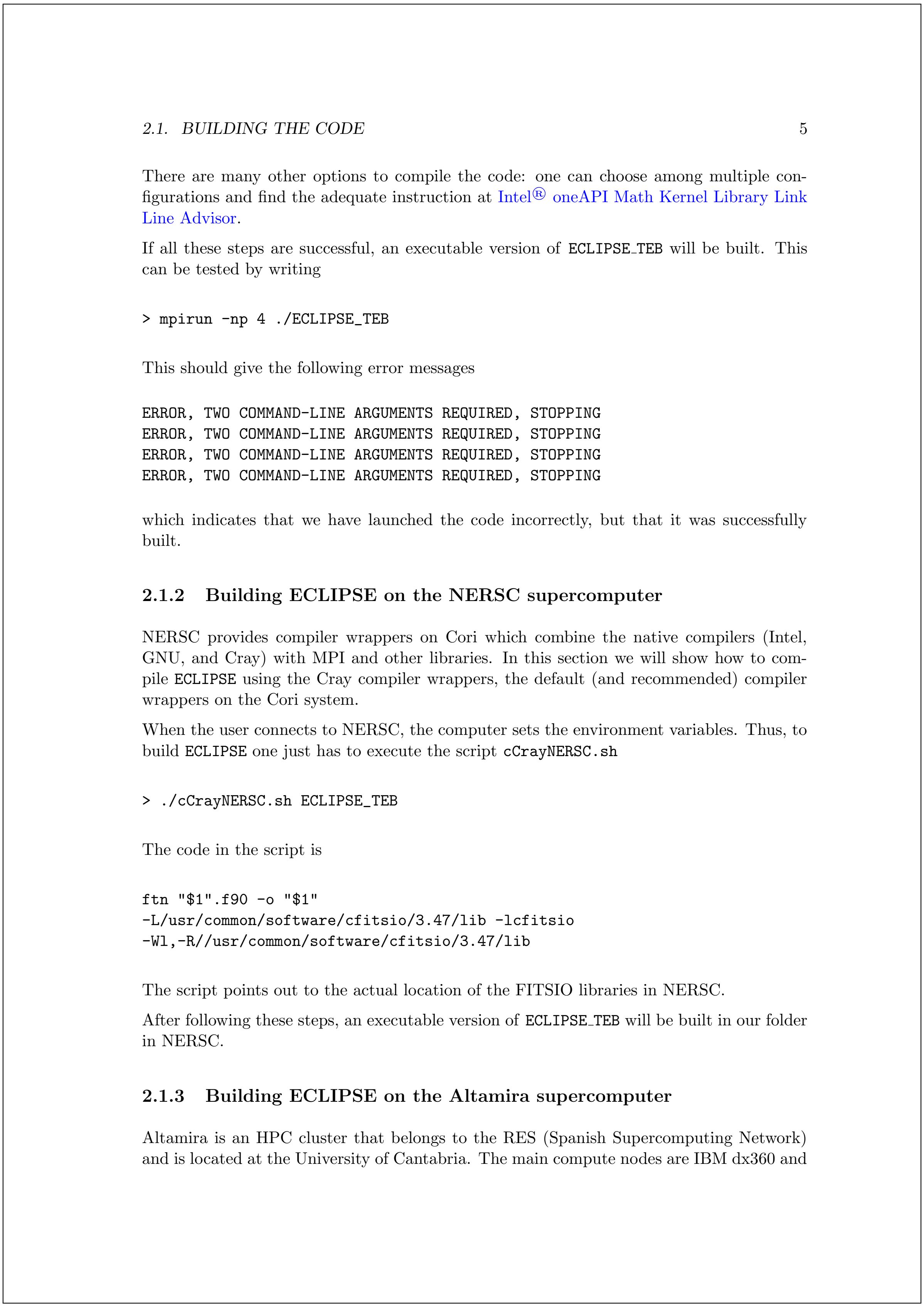
   The script contains the code

   ```
   mpiifort "$1".f90 -o "$1" -O3  -I"${MKLROOT}/include"
   -L/usr/local/lib/libcfitsio.a
   ${MKLROOT}/lib/intel64/libmkl_scalapack_lp64.a -Wl,
   --start-group ${MKLROOT}/lib/intel64/libmkl_intel_lp64.a
   ${MKLROOT}/lib/intel64/libmkl_intel_thread.a
   ${MKLROOT}/lib/intel64/libmkl_core.a
   ${MKLROOT}/lib/intel64/libmkl_blacs_intelmpi_lp64.a
   -lcfitsio -Wl,--end-group -liomp5 -lpthread -lm -ldl
   ```

   Bear in mind the root path for `libcfitsio.a` is assumed to be `/usr/local/lib`. Modify your script accordingly should it be placed in any other folder.

---

[5] There is another slightly different script, `cipLinux_Alt.sh`, in the repository. It can be useful, if the previous one does not work.



There are many other options to compile the code: one can choose among multiple configurations and find the adequate instruction at Intel® oneAPI Math Kernel Library Link Line Advisor.

If all these steps are successful, an executable version of `ECLIPSE_TEB` will be built. This can be tested by writing

```
> mpirun -np 4 ./ECLIPSE_TEB
```

This should give the following error messages

```
ERROR, TWO COMMAND-LINE ARGUMENTS REQUIRED, STOPPING
ERROR, TWO COMMAND-LINE ARGUMENTS REQUIRED, STOPPING
ERROR, TWO COMMAND-LINE ARGUMENTS REQUIRED, STOPPING
ERROR, TWO COMMAND-LINE ARGUMENTS REQUIRED, STOPPING
```

which indicates that we have launched the code incorrectly, but that it was successfully built.

2.1.2 Building ECLIPSE on the NERSC supercomputer

NERSC provides compiler wrappers on Cori which combine the native compilers (Intel, GNU, and Cray) with MPI and other libraries. In this section we will show how to compile `ECLIPSE` using the Cray compiler wrappers, the default (and recommended) compiler wrappers on the Cori system.

When the user connects to NERSC, the computer sets the environment variables. Thus, to build `ECLIPSE` one just has to execute the script `cCrayNERSC.sh`

```
> ./cCrayNERSC.sh ECLIPSE_TEB
```

The code in the script is

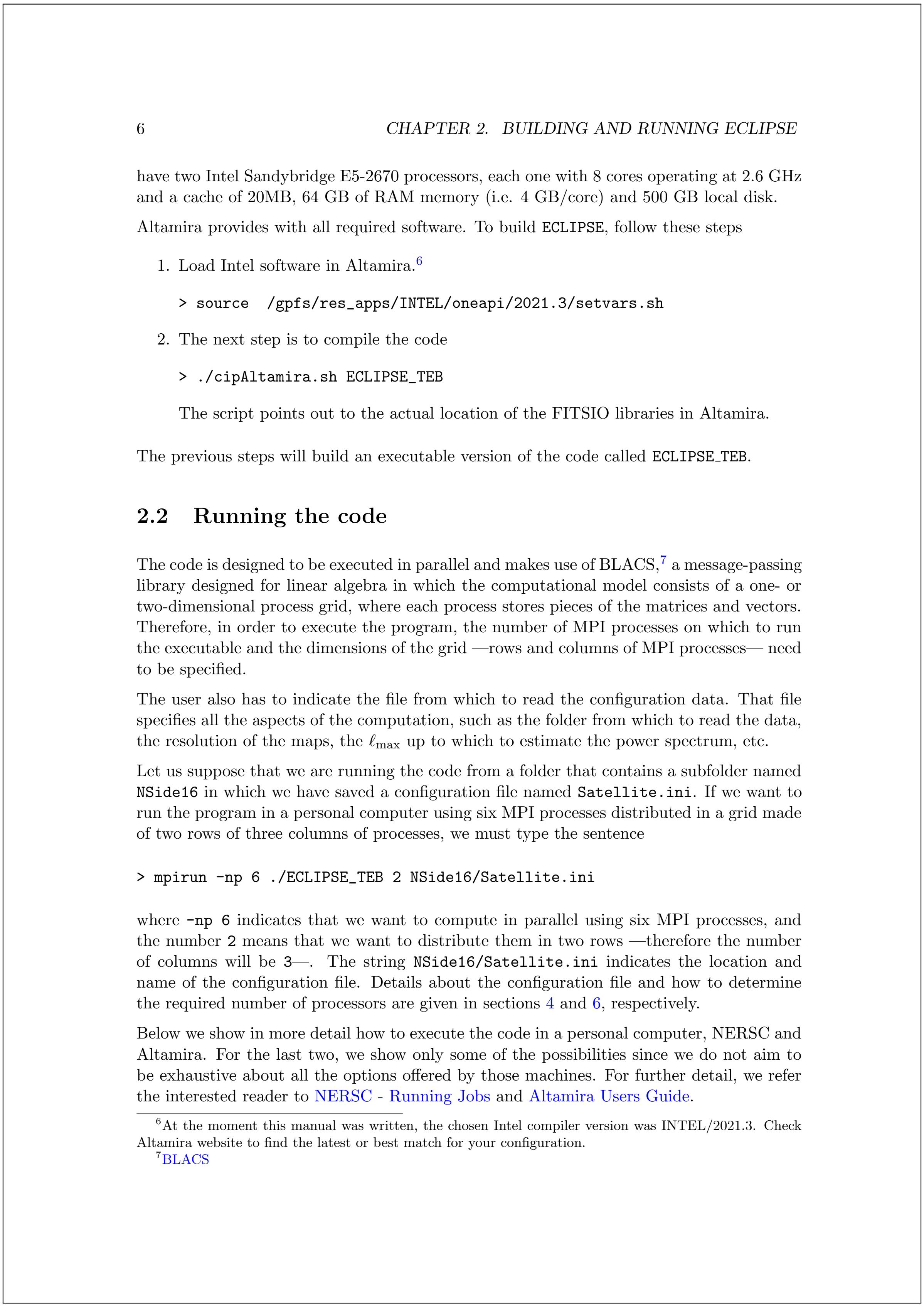
```
ftn "$1".f90 -o "$1"
-L/usr/common/software/cfitsio/3.47/lib -lcfitsio
-Wl,-R//usr/common/software/cfitsio/3.47/lib
```

The script points out to the actual location of the FITSIO libraries in NERSC.

After following these steps, an executable version of `ECLIPSE_TEB` will be built in our folder in NERSC.

2.1.3 Building ECLIPSE on the Altamira supercomputer

Altamira is an HPC cluster that belongs to the RES (Spanish Supercomputing Network) and is located at the University of Cantabria. The main compute nodes are IBM dx360 and



have two Intel Sandybridge E5-2670 processors, each one with 8 cores operating at 2.6 GHz and a cache of 20MB, 64 GB of RAM memory (i.e. 4 GB/core) and 500 GB local disk.

Altamira provides with all required software. To build `ECLIPSE`, follow these steps

1. Load Intel software in Altamira.[6]

   ```
   > source  /gpfs/res_apps/INTEL/oneapi/2021.3/setvars.sh
   ```

2. The next step is to compile the code

   ```
   > ./cipAltamira.sh ECLIPSE_TEB
   ```

   The script points out to the actual location of the FITSIO libraries in Altamira.

The previous steps will build an executable version of the code called `ECLIPSE_TEB`.

2.2 Running the code

The code is designed to be executed in parallel and makes use of BLACS,[7] a message-passing library designed for linear algebra in which the computational model consists of a one- or two-dimensional process grid, where each process stores pieces of the matrices and vectors. Therefore, in order to execute the program, the number of MPI processes on which to run the executable and the dimensions of the grid —rows and columns of MPI processes— need to be specified.

The user also has to indicate the file from which to read the configuration data. That file specifies all the aspects of the computation, such as the folder from which to read the data, the resolution of the maps, the $\ell_{\max}$ up to which to estimate the power spectrum, etc.

Let us suppose that we are running the code from a folder that contains a subfolder named `NSide16` in which we have saved a configuration file named `Satellite.ini`. If we want to run the program in a personal computer using six MPI processes distributed in a grid made of two rows of three columns of processes, we must type the sentence

```
> mpirun -np 6 ./ECLIPSE_TEB 2 NSide16/Satellite.ini
```

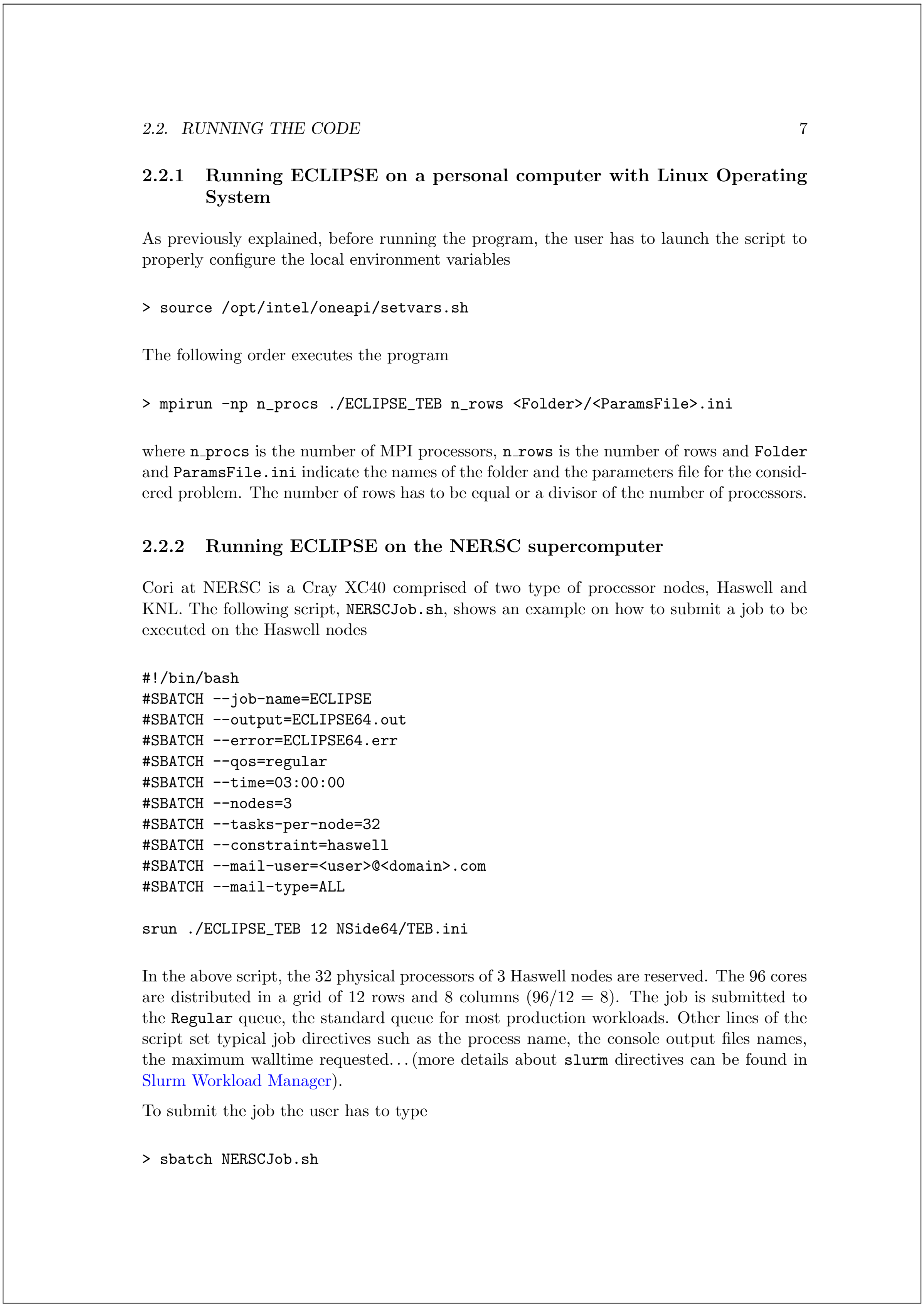
where `-np 6` indicates that we want to compute in parallel using six MPI processes, and the number `2` means that we want to distribute them in two rows —therefore the number of columns will be `3`—. The string `NSide16/Satellite.ini` indicates the location and name of the configuration file. Details about the configuration file and how to determine the required number of processors are given in sections 4 and 6, respectively.

Below we show in more detail how to execute the code in a personal computer, NERSC and Altamira. For the last two, we show only some of the possibilities since we do not aim to be exhaustive about all the options offered by those machines. For further detail, we refer the interested reader to NERSC - Running Jobs and Altamira Users Guide.

---

[6] At the moment this manual was written, the chosen Intel compiler version was INTEL/2021.3. Check Altamira website to find the latest or best match for your configuration.

[7] BLACS



2.2.1 Running ECLIPSE on a personal computer with Linux Operating System

As previously explained, before running the program, the user has to launch the script to properly configure the local environment variables

```
> source /opt/intel/oneapi/setvars.sh
```

The following order executes the program

```
> mpirun -np n_procs ./ECLIPSE_TEB n_rows <Folder>/<ParamsFile>.ini
```

where `n_procs` is the number of MPI processors, `n_rows` is the number of rows and `Folder` and `ParamsFile.ini` indicate the names of the folder and the parameters file for the considered problem. The number of rows has to be equal or a divisor of the number of processors.

2.2.2 Running ECLIPSE on the NERSC supercomputer

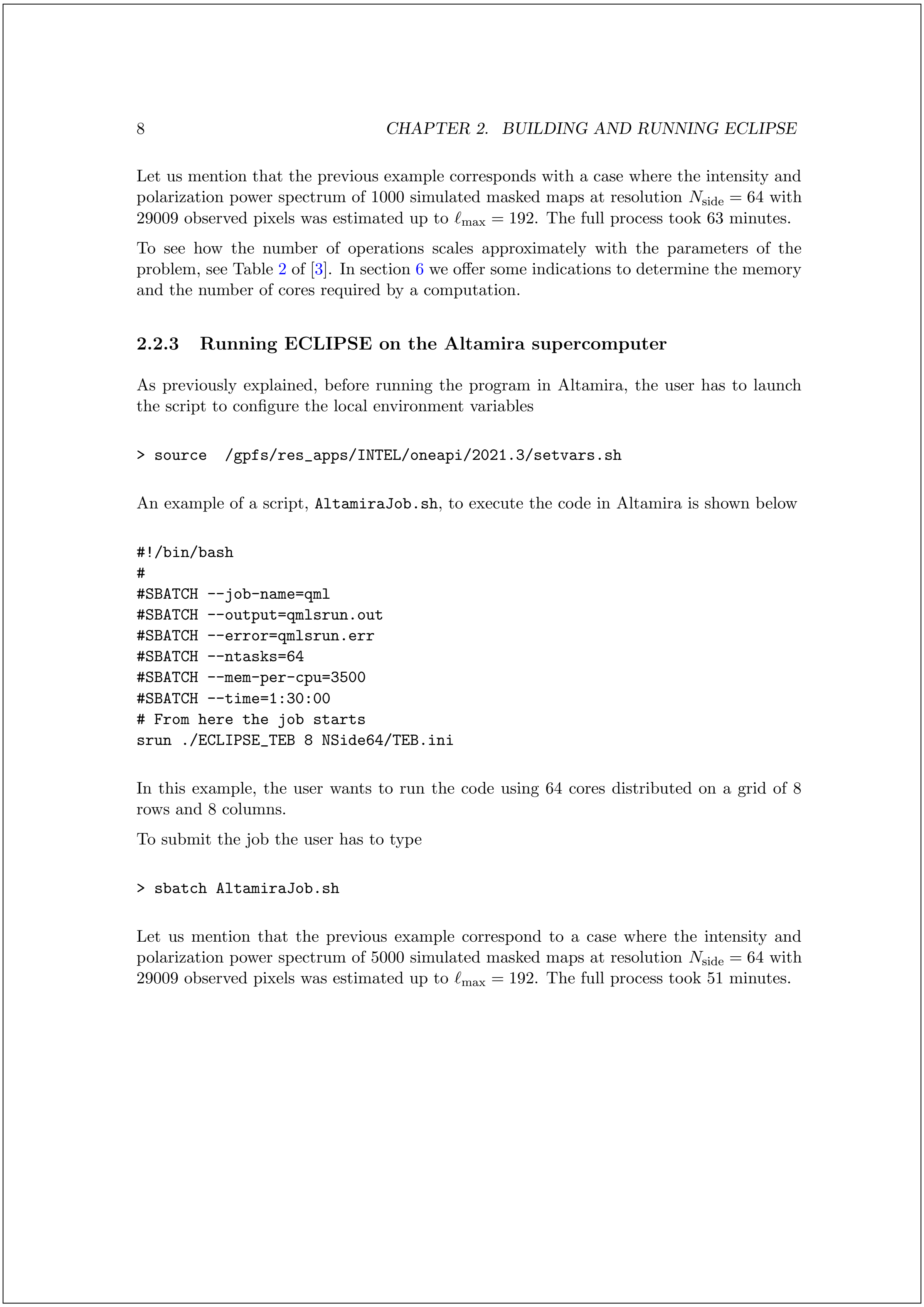
Cori at NERSC is a Cray XC40 comprised of two type of processor nodes, Haswell and KNL. The following script, `NERSCJob.sh`, shows an example on how to submit a job to be executed on the Haswell nodes

```
#!/bin/bash
#SBATCH --job-name=ECLIPSE
#SBATCH --output=ECLIPSE64.out
#SBATCH --error=ECLIPSE64.err
#SBATCH --qos=regular
#SBATCH --time=03:00:00
#SBATCH --nodes=3
#SBATCH --tasks-per-node=32
#SBATCH --constraint=haswell
#SBATCH --mail-user=<user>@<domain>.com
#SBATCH --mail-type=ALL

srun ./ECLIPSE_TEB 12 NSide64/TEB.ini
```

In the above script, the 32 physical processors of 3 Haswell nodes are reserved. The 96 cores are distributed in a grid of 12 rows and 8 columns ($96/12 = 8$). The job is submitted to the `Regular` queue, the standard queue for most production workloads. Other lines of the script set typical job directives such as the process name, the console output files names, the maximum walltime requested…(more details about `slurm` directives can be found in Slurm Workload Manager).

To submit the job the user has to type

```
> sbatch NERSCJob.sh
```



Let us mention that the previous example corresponds with a case where the intensity and polarization power spectrum of 1000 simulated masked maps at resolution $N_{\text{side}} = 64$ with 29009 observed pixels was estimated up to $\ell_{\max} = 192$. The full process took 63 minutes.

To see how the number of operations scales approximately with the parameters of the problem, see Table 2 of [3]. In section 6 we offer some indications to determine the memory and the number of cores required by a computation.

2.2.3 Running ECLIPSE on the Altamira supercomputer

As previously explained, before running the program in Altamira, the user has to launch the script to configure the local environment variables

```
> source  /gpfs/res_apps/INTEL/oneapi/2021.3/setvars.sh
```

An example of a script, `AltamiraJob.sh`, to execute the code in Altamira is shown below

```
#!/bin/bash
#
#SBATCH --job-name=qml
#SBATCH --output=qmlsrun.out
#SBATCH --error=qmlsrun.err
#SBATCH --ntasks=64
#SBATCH --mem-per-cpu=3500
#SBATCH --time=1:30:00
# From here the job starts
srun ./ECLIPSE_TEB 8 NSide64/TEB.ini
```

In this example, the user wants to run the code using 64 cores distributed on a grid of 8 rows and 8 columns.

To submit the job the user has to type

```
> sbatch AltamiraJob.sh
```

Let us mention that the previous example correspond to a case where the intensity and polarization power spectrum of 5000 simulated masked maps at resolution $N_{\text{side}} = 64$ with 29009 observed pixels was estimated up to $\ell_{\max} = 192$. The full process took 51 minutes.

Chapter 3

The QML estimator and ECLIPSE functionalities

Before describing how to use `ECLIPSE` and the files the code reads and write, let us describe the mathematics of the Quadratic Maximum Likelihood (QML) estimator.

3.1 General description of the QML

The QML is a method for obtaining an optimal estimation of the CMB power spectra and its covariance matrix from a map, which is well suited to deal with incomplete sky coverage. Assuming that the CMB fluctuations are Gaussian and isotropic, [1] and [2] show that given a CMB temperature map or CMB temperature and polarization maps $\mathbf{x}$, an estimation of the power spectra can be found in a two-step process

1. Starting from the pixels in the map, compute an angular quantity y_i that is related to the power spectrum.
2. Given this quantity, define an estimator of the power spectrum.

Before describing the method, let us first establish some basic notation (details can be found in [2]). In the only-temperature case, the map $\mathbf{x}$ is an N-dimensional vector of elements $T_i = T(r_i)$; in the full case, the map is a $3N$-dimensional vector of values T_i, Q_i, U_i.[1] The method requires a model of the signal and of the noise, that are both introduced through the signal $\mathbf{S}$ and the noise $\mathbf{N}$ covariance matrices, respectively.

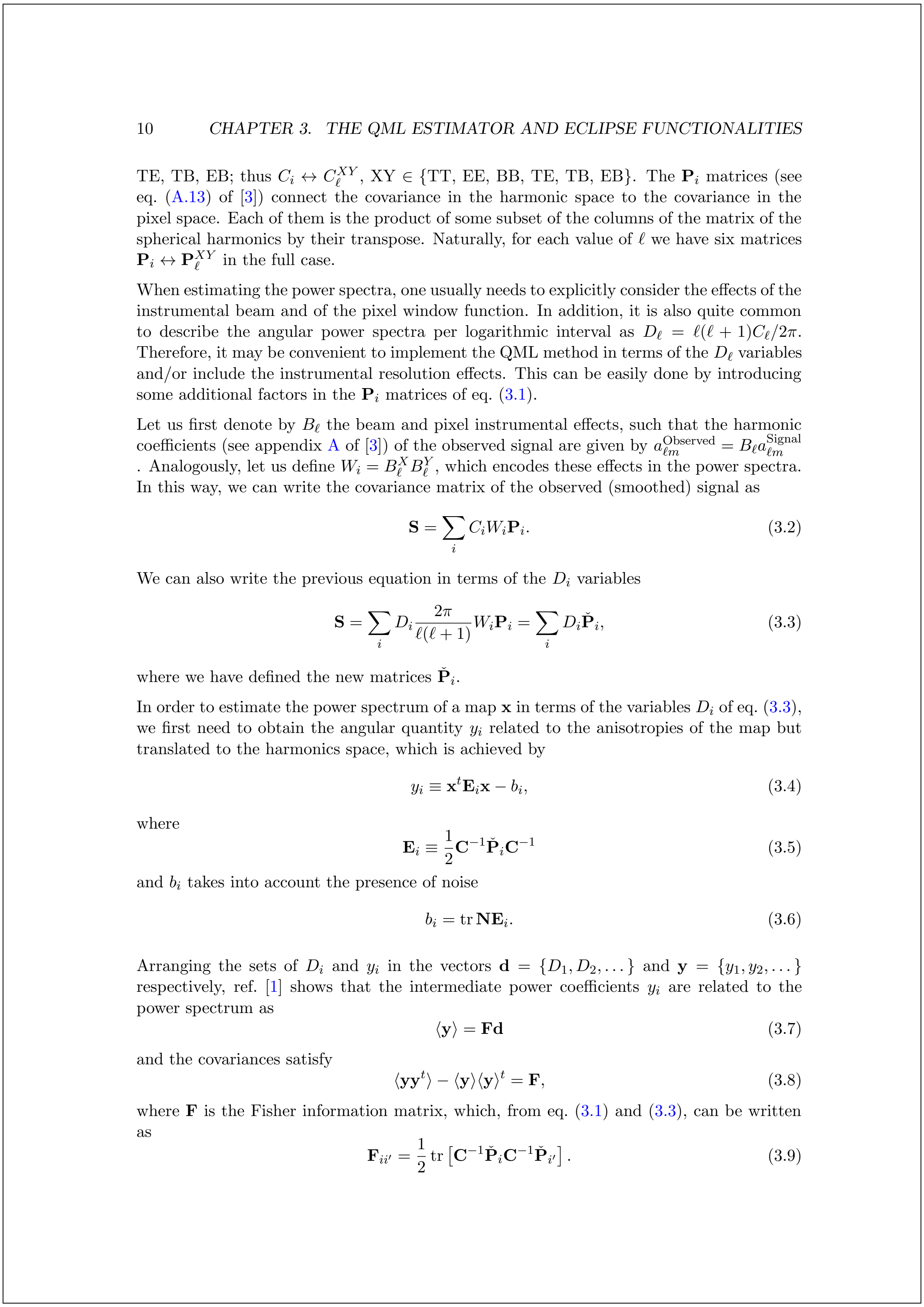
The statistical properties of $\mathbf{x}$ are characterized by the power spectra C_i, and assuming that the signal and the noise are uncorrelated, we have

$$\mathbf{C} \equiv \langle \mathbf{x}\mathbf{x}^t \rangle = \mathbf{S} + \mathbf{N} = \sum_i C_i \mathbf{P}_i + \mathbf{N}. \tag{3.1}$$

Note that in the only-temperature case the index i can be directly substituted by the multipole index ℓ, while in the full case i includes ℓ and one of the six pairs TT, EE, BB,

---

[1] Actually, `ECLIPSE` can work with a different number of pixels in intensity and polarization, but for simplicity we will assume that they are the same.



TE, TB, EB; thus $C_i \leftrightarrow C_\ell^{XY}$, XY $\in$ {TT, EE, BB, TE, TB, EB}. The $\mathbf{P}_i$ matrices (see eq. (A.13) of [3]) connect the covariance in the harmonic space to the covariance in the pixel space. Each of them is the product of some subset of the columns of the matrix of the spherical harmonics by their transpose. Naturally, for each value of ℓ we have six matrices $\mathbf{P}_i \leftrightarrow \mathbf{P}_\ell^{XY}$ in the full case.

When estimating the power spectra, one usually needs to explicitly consider the effects of the instrumental beam and of the pixel window function. In addition, it is also quite common to describe the angular power spectra per logarithmic interval as $D_\ell = \ell(\ell+1)C_\ell/2\pi$. Therefore, it may be convenient to implement the QML method in terms of the D_ℓ variables and/or include the instrumental resolution effects. This can be easily done by introducing some additional factors in the $\mathbf{P}_i$ matrices of eq. (3.1).

Let us first denote by B_ℓ the beam and pixel instrumental effects, such that the harmonic coefficients (see appendix A of [3]) of the observed signal are given by $a_{\ell m}^{\text{Observed}} = B_\ell a_{\ell m}^{\text{Signal}}$. Analogously, let us define $W_i = B_\ell^X B_\ell^Y$, which encodes these effects in the power spectra. In this way, we can write the covariance matrix of the observed (smoothed) signal as

$$\mathbf{S} = \sum_i C_i W_i \mathbf{P}_i. \tag{3.2}$$

We can also write the previous equation in terms of the D_i variables

$$\mathbf{S} = \sum_i D_i \frac{2\pi}{\ell(\ell+1)} W_i \mathbf{P}_i = \sum_i D_i \check{\mathbf{P}}_i, \tag{3.3}$$

where we have defined the new matrices $\check{\mathbf{P}}_i$.

In order to estimate the power spectrum of a map $\mathbf{x}$ in terms of the variables D_i of eq. (3.3), we first need to obtain the angular quantity y_i related to the anisotropies of the map but translated to the harmonics space, which is achieved by

$$y_i \equiv \mathbf{x}^t \mathbf{E}_i \mathbf{x} - b_i, \tag{3.4}$$

where

$$\mathbf{E}_i \equiv \frac{1}{2} \mathbf{C}^{-1} \check{\mathbf{P}}_i \mathbf{C}^{-1} \tag{3.5}$$

and b_i takes into account the presence of noise

$$b_i = \operatorname{tr} \mathbf{N}\mathbf{E}_i. \tag{3.6}$$

Arranging the sets of D_i and y_i in the vectors $\mathbf{d} = \{D_1, D_2, \dots\}$ and $\mathbf{y} = \{y_1, y_2, \dots\}$ respectively, ref. [1] shows that the intermediate power coefficients y_i are related to the power spectrum as

$$\langle \mathbf{y} \rangle = \mathbf{F}\mathbf{d} \tag{3.7}$$

and the covariances satisfy

$$\langle \mathbf{y}\mathbf{y}^t \rangle - \langle \mathbf{y} \rangle \langle \mathbf{y} \rangle^t = \mathbf{F}, \tag{3.8}$$

where $\mathbf{F}$ is the Fisher information matrix, which, from eq. (3.1) and (3.3), can be written as

$$\mathbf{F}_{ii'} = \frac{1}{2} \operatorname{tr} \left[\mathbf{C}^{-1} \check{\mathbf{P}}_i \mathbf{C}^{-1} \check{\mathbf{P}}_{i'} \right]. \tag{3.9}$$

If $\mathbf{F}$ is regular, the power spectrum estimator can be defined as

$$\hat{\mathbf{d}} \equiv \mathbf{F}^{-1}\mathbf{y}. \tag{3.10}$$

Combining this definition with eq. (3.7), we get $\langle\hat{\mathbf{d}}\rangle = \mathbf{d}$. The covariance matrix of this estimator is the inverse of the Fisher matrix

$$\langle(\hat{\mathbf{d}} - \mathbf{d})(\hat{\mathbf{d}} - \mathbf{d})^t\rangle = \mathbf{F}^{-1}[\langle\mathbf{y}\mathbf{y}^t\rangle - \langle\mathbf{y}\rangle\langle\mathbf{y}\rangle^t]\mathbf{F}^{-1} = \mathbf{F}^{-1}. \tag{3.11}$$

Therefore, the estimator is unbiased and, by the Cramer-Rao inequality, of minimal error bars: QML is mathematically equivalent to the Maximum Likelihood Estimator, but, since it does not require a brutal force maximization, it reduces the computational costs.

Note that from eq. (3.1) and (3.3) the covariance matrix of the signal $\mathbf{S}$ is computed from the fiducial model the user provides with. Therefore, an initial guess for the sought power spectra is required in order to compute the QML estimator. This leads naturally to the possibility of using an iterative scheme in order to update the initial fiducial model and, thus, to improve the final estimation of the power spectra.[2]

It is essential to mention that `ECLIPSE` incorporates an implementation of the method such that the covariance matrix of the signal on the observed maps is computed from the last term of eq. (3.3). Therefore, the fiducial model needed to compute the covariance matrix is given in the form of the variables D_i, and must reflect the power in the signal. In parallel, `ECLIPSE` estimates the power spectra of the signal of the maps —corrected from the beam and pixel effects— in format D_i.[3]

The QML estimator requires matrices $\mathbf{C}$ and $\mathbf{F}$ to be regular. If necessary, the covariance matrix can be regularized by adding a small amount of noise (a detailed analysis on the conditions on which $\mathbf{C}$ is regular can be found in [4]). If the Fisher matrix is singular, an optimal binned QML can be implemented as described below.

3.1.1 The binned estimator

In section 4 of [3] we describe an optimal binned estimator of the power spectrum

$$\hat{\mathbf{b}} = [\mathbf{R}^t\mathbf{F}\mathbf{R}]^{-1}\mathbf{R}^t\mathbf{y}, \tag{3.12}$$

where $\hat{\mathbf{b}}$ is the vector of the estimated binned power spectrum, $\mathbf{F}$ is the Fisher matrix and $\mathbf{y}$ is the vector of elements y_i of eq. (3.4). The matrix $\mathbf{R}$, computed internally by `ECLIPSE`, is defined as

$$R_{ib} = \left\{ \begin{array}{ll} f_i^b & \text{if} \quad \ell \in L_b \\ 0 & \text{otherwise} \end{array} \right. , \tag{3.13}$$

where i refers to the index of the multipoles, b to the index of the bins and L_b is the collection of indexes i in the bin, that is, a collection of some consecutive ℓ's of some of the six components of the power spectrum. The factors f_i^b are

---

[2] `ECLIPSE` does not implement an iterative scheme. Details about iterative QML can be found in section 5.2 of [3].

[3] Section 3.2.5 shows how to work in terms of the observed power in the maps for both fiducial and estimated power spectrum.

$$f_i^b = D_i/B_b, \tag{3.14}$$

where D_i is the fiducial power spectrum provided by the user and B_b is the fiducial power-band, computed from the fiducial spectrum. In the configuration file there is a set of `key = value` pairs which defines how B_b will be computed.

Considering

`Type_of_Bin_Center = 1`

the values B_b would be

$$B_b = \frac{\sum_{i\in L_b} \frac{D_i}{(\Delta D_i)^2}}{\sum_{i\in L_b} \frac{1}{(\Delta D_i)^2}} \tag{3.15}$$

and the positions of the centers of the bins

$$\ell_b^* = \frac{\sum_{i\in L_b} \frac{\ell(i)}{(\Delta D_i)^2}}{\sum_{i\in L_b} \frac{1}{(\Delta D_i)^2}}, \tag{3.16}$$

where ΔD_i are the theoretical errors [5] taking into account cosmic variance, noise and sky fraction. $\ell(i)$ makes reference to the value of ℓ that corresponds to each index i.[4]

In the case of `Type_of_Bin_Center = 0` the means above are computed assuming $\Delta D_\ell = 1$.

The Fisher matrix of the binned spectrum, which we will call $\mathbf{G}$, is computed from the Fisher matrix of the multipoles and the matrix $\mathbf{R}$

$$\mathbf{G} = \mathbf{R}^t\mathbf{F}\mathbf{R}. \tag{3.17}$$

The covariances of the binned spectrum are

$$\langle(\hat{\mathbf{b}} - \mathbf{b})(\hat{\mathbf{b}} - \mathbf{b})^t\rangle = [\mathbf{R}^t\,\mathbf{F}\,\mathbf{R}]^{-1}, \tag{3.18}$$

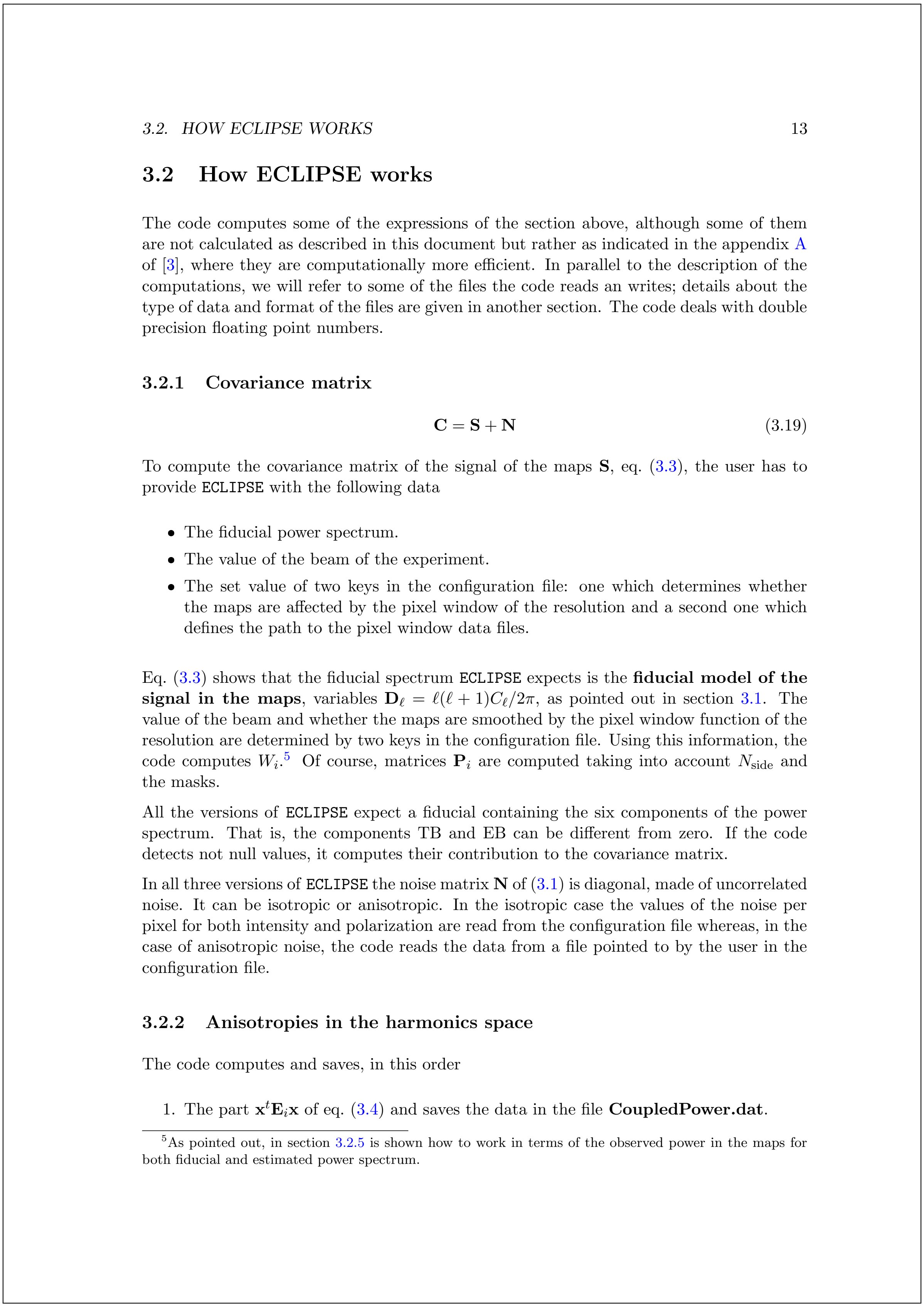
that is, they are equal to the inverse of the Fisher matrix of the estimated binned spectrum.

The two binning options in ECLIPSE

Although the optimal binned estimator is achieved being the factors f_ℓ^b as defined if eq. (3.14), there is a key in the configuration file that tells `ECLIPSE` how to compute them. If `Type_Of_Grouping = 1`, eq. (3.14) is used. If `Type_Of_Grouping = 0`, the code assumes that the non null values of f_ℓ^b are equal to 1. This option allows to instruct `ECLIPSE` to compute the binned estimation without using the information in the fiducial spectrum.

---

[4] For example, in the case of an intensity and polarization estimation there are six different values of i that point to the same value $\ell(i)$, one for each component of the power spectrum.



3.2 How ECLIPSE works

The code computes some of the expressions of the section above, although some of them are not calculated as described in this document but rather as indicated in the appendix A of [3], where they are computationally more efficient. In parallel to the description of the computations, we will refer to some of the files the code reads an writes; details about the type of data and format of the files are given in another section. The code deals with double precision floating point numbers.

3.2.1 Covariance matrix

$$\mathbf{C} = \mathbf{S} + \mathbf{N} \tag{3.19}$$

To compute the covariance matrix of the signal of the maps $\mathbf{S}$, eq. (3.3), the user has to provide `ECLIPSE` with the following data

- The fiducial power spectrum.
- The value of the beam of the experiment.
- The set value of two keys in the configuration file: one which determines whether the maps are affected by the pixel window of the resolution and a second one which defines the path to the pixel window data files.

Eq. (3.3) shows that the fiducial spectrum `ECLIPSE` expects is the **fiducial model of the signal in the maps**, variables $\mathbf{D}_\ell = \ell(\ell+1)C_\ell/2\pi$, as pointed out in section 3.1. The value of the beam and whether the maps are smoothed by the pixel window function of the resolution are determined by two keys in the configuration file. Using this information, the code computes W_i.[5] Of course, matrices $\mathbf{P}_i$ are computed taking into account $N_{\rm side}$ and the masks.

All the versions of `ECLIPSE` expect a fiducial containing the six components of the power spectrum. That is, the components TB and EB can be different from zero. If the code detects not null values, it computes their contribution to the covariance matrix.

In all three versions of `ECLIPSE` the noise matrix $\mathbf{N}$ of (3.1) is diagonal, made of uncorrelated noise. It can be isotropic or anisotropic. In the isotropic case the values of the noise per pixel for both intensity and polarization are read from the configuration file whereas, in the case of anisotropic noise, the code reads the data from a file pointed to by the user in the configuration file.

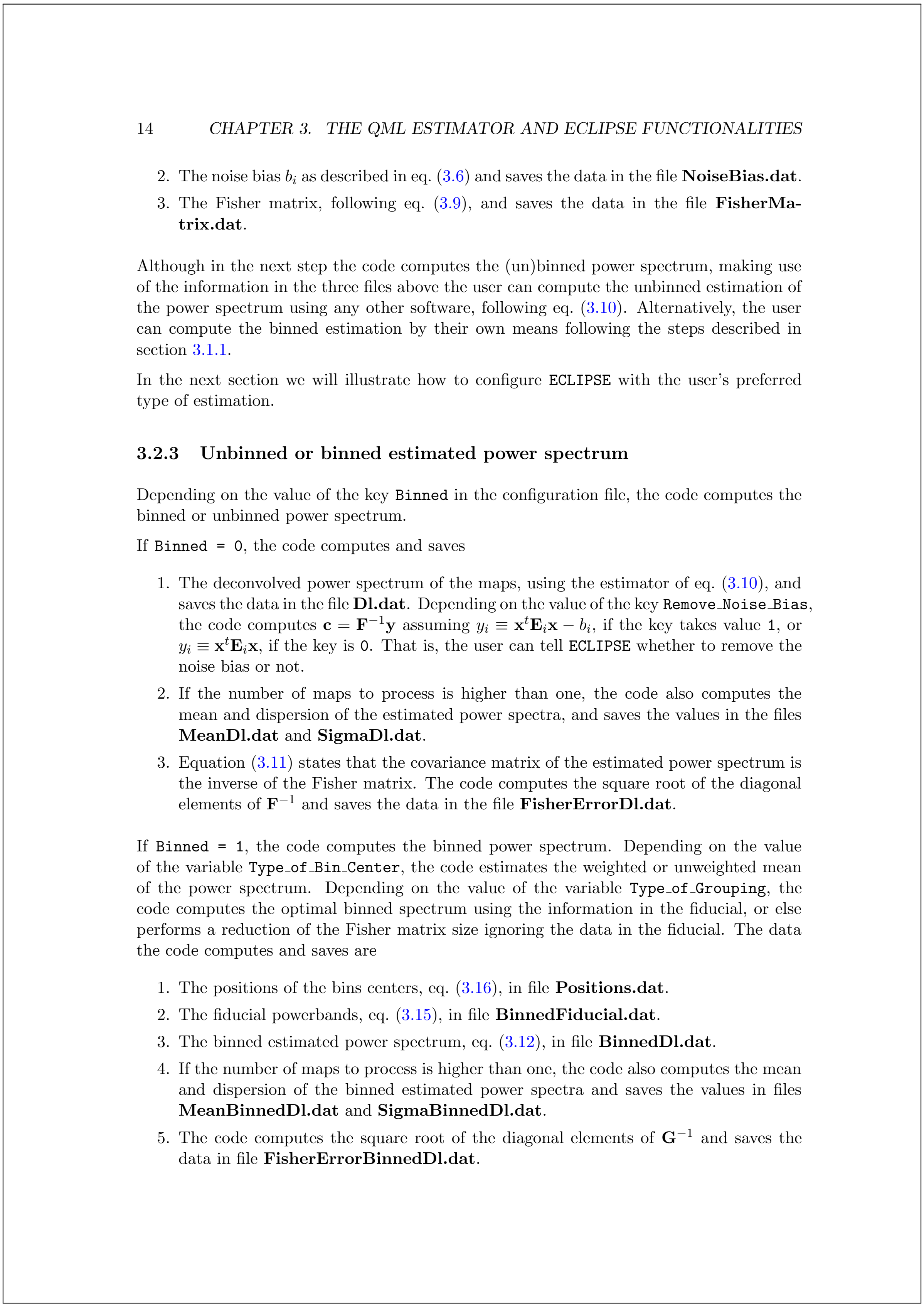
3.2.2 Anisotropies in the harmonics space

The code computes and saves, in this order

1. The part $\mathbf{x}^t\mathbf{E}_i\mathbf{x}$ of eq. (3.4) and saves the data in the file **CoupledPower.dat**.

---

[5] As pointed out, in section 3.2.5 is shown how to work in terms of the observed power in the maps for both fiducial and estimated power spectrum.



2. The noise bias b_i as described in eq. (3.6) and saves the data in the file **NoiseBias.dat**.
3. The Fisher matrix, following eq. (3.9), and saves the data in the file **FisherMatrix.dat**.

Although in the next step the code computes the (un)binned power spectrum, making use of the information in the three files above the user can compute the unbinned estimation of the power spectrum using any other software, following eq. (3.10). Alternatively, the user can compute the binned estimation by their own means following the steps described in section 3.1.1.

In the next section we will illustrate how to configure `ECLIPSE` with the user's preferred type of estimation.

3.2.3 Unbinned or binned estimated power spectrum

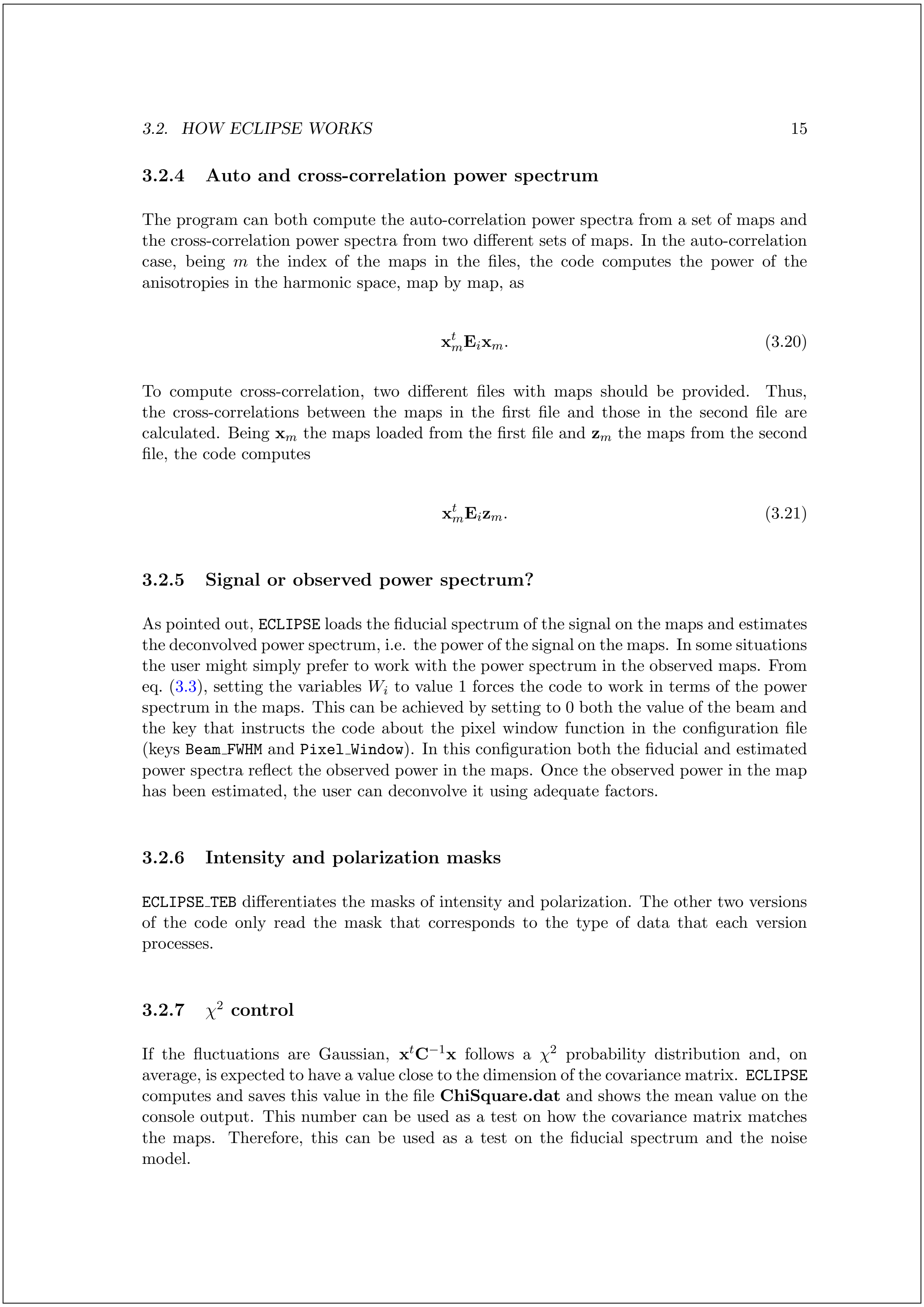
Depending on the value of the key `Binned` in the configuration file, the code computes the binned or unbinned power spectrum.

If `Binned = 0`, the code computes and saves

1. The deconvolved power spectrum of the maps, using the estimator of eq. (3.10), and saves the data in the file **Dl.dat**. Depending on the value of the key `Remove_Noise_Bias`, the code computes $\mathbf{c} = \mathbf{F}^{-1}\mathbf{y}$ assuming $y_i \equiv \mathbf{x}^t\mathbf{E}_i\mathbf{x} - b_i$, if the key takes value `1`, or $y_i \equiv \mathbf{x}^t\mathbf{E}_i\mathbf{x}$, if the key is `0`. That is, the user can tell `ECLIPSE` whether to remove the noise bias or not.
2. If the number of maps to process is higher than one, the code also computes the mean and dispersion of the estimated power spectra, and saves the values in the files **MeanDl.dat** and **SigmaDl.dat**.
3. Equation (3.11) states that the covariance matrix of the estimated power spectrum is the inverse of the Fisher matrix. The code computes the square root of the diagonal elements of $\mathbf{F}^{-1}$ and saves the data in the file **FisherErrorDl.dat**.

If `Binned = 1`, the code computes the binned power spectrum. Depending on the value of the variable `Type_of_Bin_Center`, the code estimates the weighted or unweighted mean of the power spectrum. Depending on the value of the variable `Type_of_Grouping`, the code computes the optimal binned spectrum using the information in the fiducial, or else performs a reduction of the Fisher matrix size ignoring the data in the fiducial. The data the code computes and saves are

1. The positions of the bins centers, eq. (3.16), in file **Positions.dat**.
2. The fiducial powerbands, eq. (3.15), in file **BinnedFiducial.dat**.
3. The binned estimated power spectrum, eq. (3.12), in file **BinnedDl.dat**.
4. If the number of maps to process is higher than one, the code also computes the mean and dispersion of the binned estimated power spectra and saves the values in files **MeanBinnedDl.dat** and **SigmaBinnedDl.dat**.
5. The code computes the square root of the diagonal elements of $\mathbf{G}^{-1}$ and saves the data in file **FisherErrorBinnedDl.dat**.



3.2.4 Auto and cross-correlation power spectrum

The program can both compute the auto-correlation power spectra from a set of maps and the cross-correlation power spectra from two different sets of maps. In the auto-correlation case, being m the index of the maps in the files, the code computes the power of the anisotropies in the harmonic space, map by map, as

$$\mathbf{x}_m^t \mathbf{E}_i \mathbf{x}_m. \tag{3.20}$$

To compute cross-correlation, two different files with maps should be provided. Thus, the cross-correlations between the maps in the first file and those in the second file are calculated. Being $\mathbf{x}_m$ the maps loaded from the first file and $\mathbf{z}_m$ the maps from the second file, the code computes

$$\mathbf{x}_m^t \mathbf{E}_i \mathbf{z}_m. \tag{3.21}$$

3.2.5 Signal or observed power spectrum?

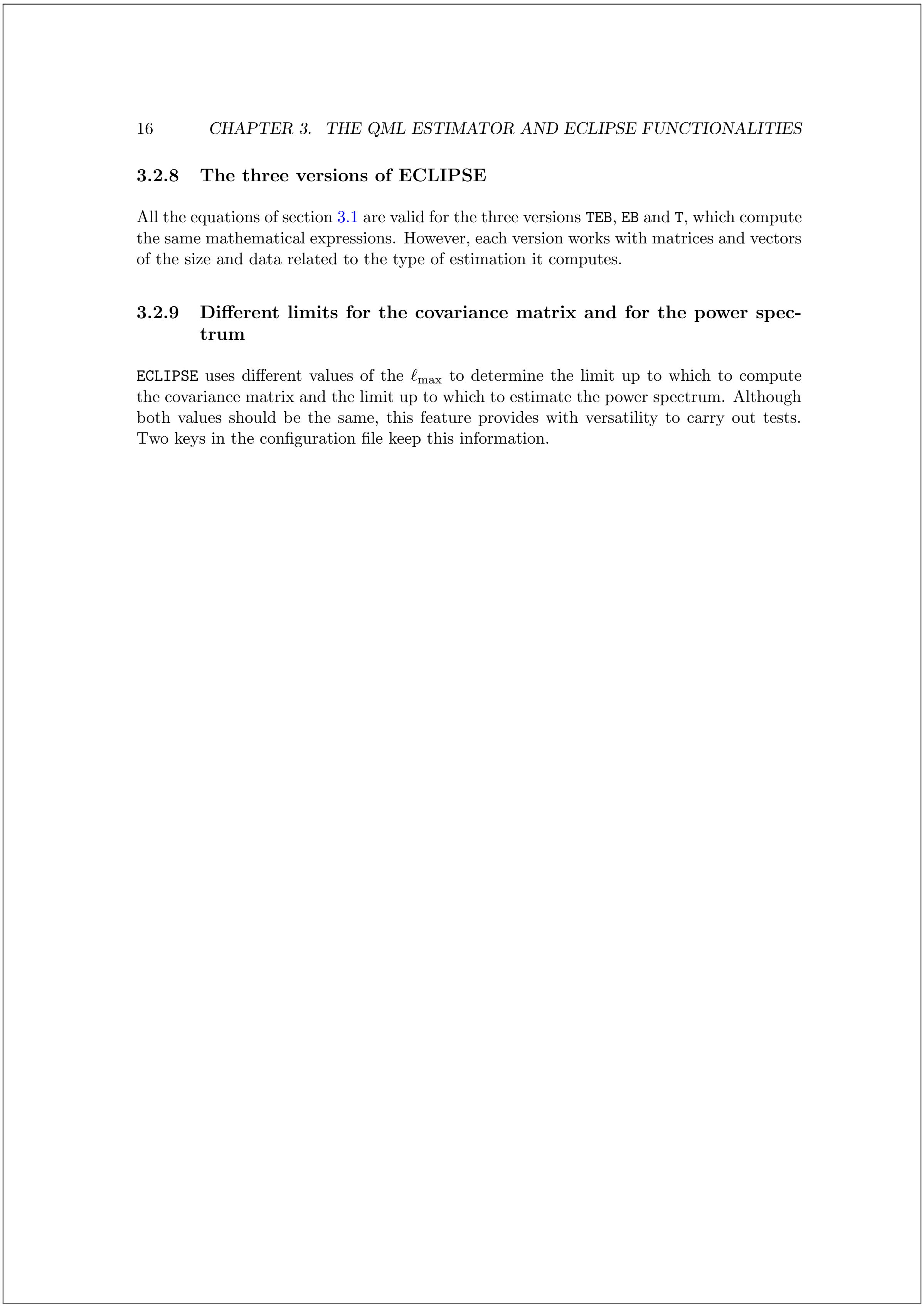
As pointed out, `ECLIPSE` loads the fiducial spectrum of the signal on the maps and estimates the deconvolved power spectrum, i.e. the power of the signal on the maps. In some situations the user might simply prefer to work with the power spectrum in the observed maps. From eq. (3.3), setting the variables W_i to value 1 forces the code to work in terms of the power spectrum in the maps. This can be achieved by setting to 0 both the value of the beam and the key that instructs the code about the pixel window function in the configuration file (keys `Beam_FWHM` and `Pixel_Window`). In this configuration both the fiducial and estimated power spectra reflect the observed power in the maps. Once the observed power in the map has been estimated, the user can deconvolve it using adequate factors.

3.2.6 Intensity and polarization masks

`ECLIPSE_TEB` differentiates the masks of intensity and polarization. The other two versions of the code only read the mask that corresponds to the type of data that each version processes.

3.2.7 χ^2 control

If the fluctuations are Gaussian, $\mathbf{x}^t\mathbf{C}^{-1}\mathbf{x}$ follows a χ^2 probability distribution and, on average, is expected to have a value close to the dimension of the covariance matrix. `ECLIPSE` computes and saves this value in the file **ChiSquare.dat** and shows the mean value on the console output. This number can be used as a test on how the covariance matrix matches the maps. Therefore, this can be used as a test on the fiducial spectrum and the noise model.

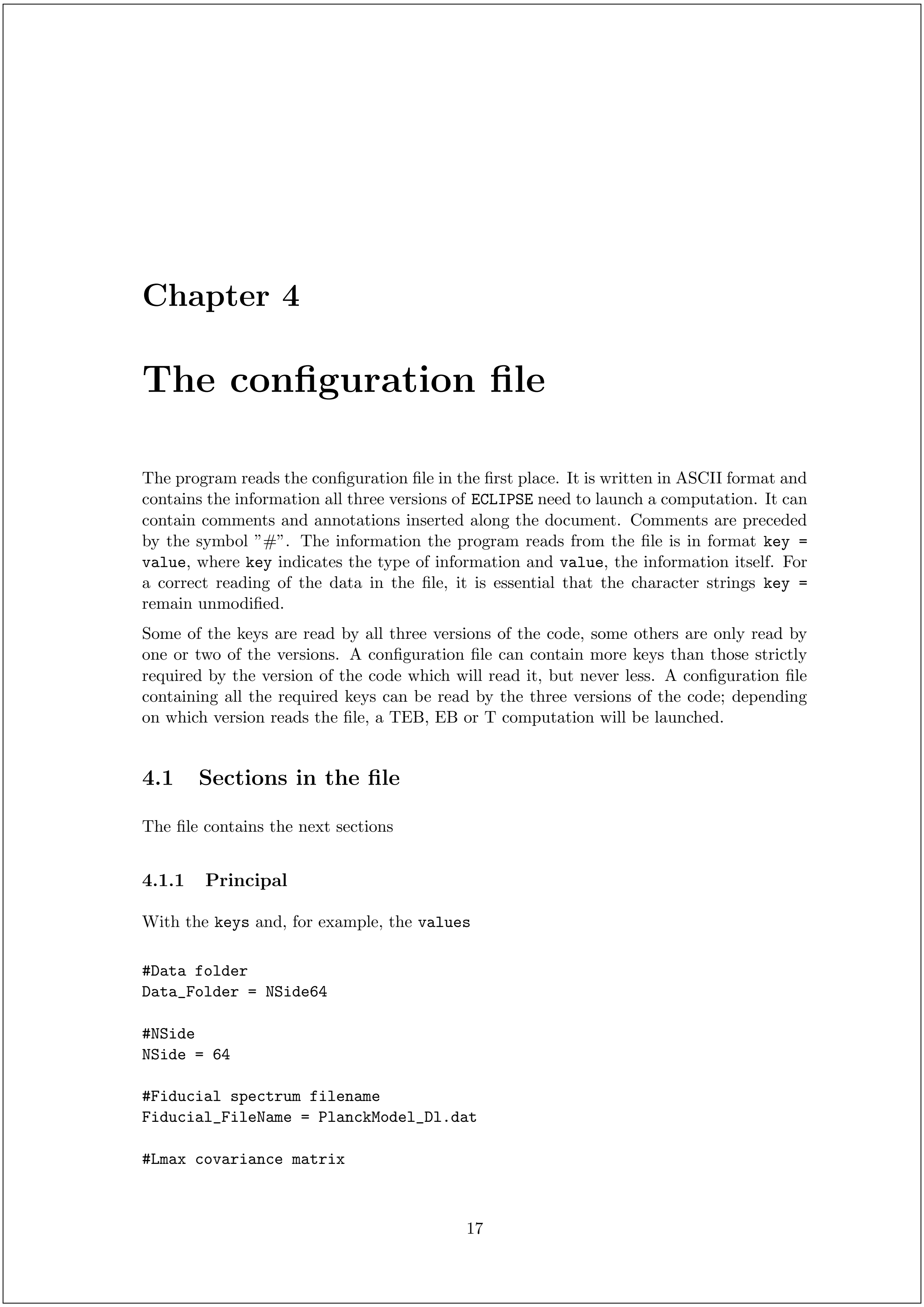


3.2.8 The three versions of ECLIPSE

All the equations of section 3.1 are valid for the three versions `TEB`, `EB` and `T`, which compute the same mathematical expressions. However, each version works with matrices and vectors of the size and data related to the type of estimation it computes.

3.2.9 Different limits for the covariance matrix and for the power spectrum

`ECLIPSE` uses different values of the $\ell_{\max}$ to determine the limit up to which to compute the covariance matrix and the limit up to which to estimate the power spectrum. Although both values should be the same, this feature provides with versatility to carry out tests. Two keys in the configuration file keep this information.



Chapter 4

The configuration file

The program reads the configuration file in the first place. It is written in ASCII format and contains the information all three versions of `ECLIPSE` need to launch a computation. It can contain comments and annotations inserted along the document. Comments are preceded by the symbol "#". The information the program reads from the file is in format `key = value`, where `key` indicates the type of information and `value`, the information itself. For a correct reading of the data in the file, it is essential that the character strings `key =` remain unmodified.

Some of the keys are read by all three versions of the code, some others are only read by one or two of the versions. A configuration file can contain more keys than those strictly required by the version of the code which will read it, but never less. A configuration file containing all the required keys can be read by the three versions of the code; depending on which version reads the file, a TEB, EB or T computation will be launched.

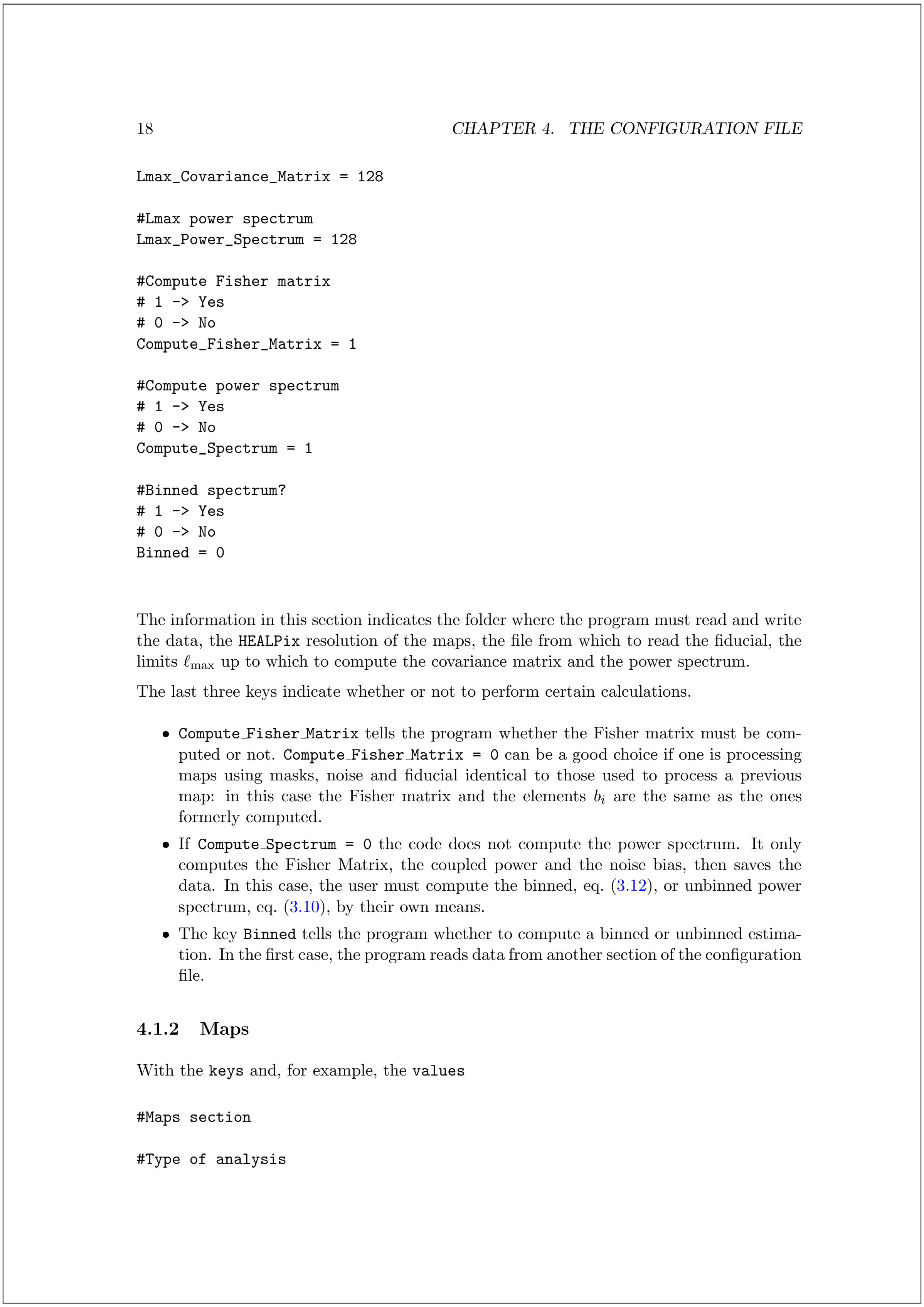
4.1 Sections in the file

The file contains the next sections

4.1.1 Principal

With the `keys` and, for example, the `values`

```
#Data folder
Data_Folder = NSide64

#NSide
NSide = 64

#Fiducial spectrum filename
Fiducial_FileName = PlanckModel_Dl.dat

#Lmax covariance matrix
```



```
Lmax_Covariance_Matrix = 128

#Lmax power spectrum
Lmax_Power_Spectrum = 128

#Compute Fisher matrix
# 1 -> Yes
# 0 -> No
Compute_Fisher_Matrix = 1

#Compute power spectrum
# 1 -> Yes
# 0 -> No
Compute_Spectrum = 1

#Binned spectrum?
# 1 -> Yes
# 0 -> No
Binned = 0
```

The information in this section indicates the folder where the program must read and write the data, the `HEALPix` resolution of the maps, the file from which to read the fiducial, the limits $\ell_{\max}$ up to which to compute the covariance matrix and the power spectrum.

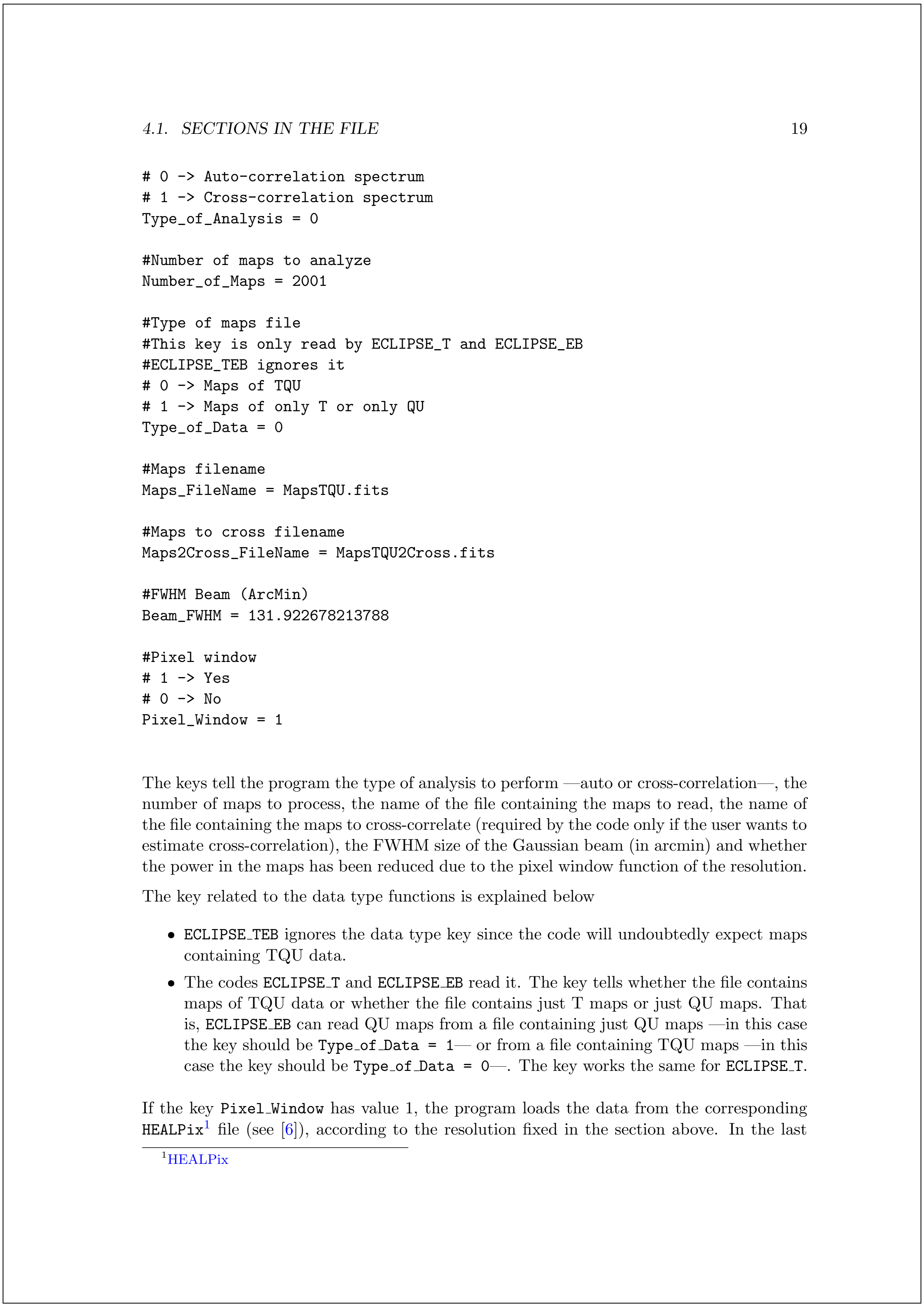
The last three keys indicate whether or not to perform certain calculations.

- `Compute_Fisher_Matrix` tells the program whether the Fisher matrix must be computed or not. `Compute_Fisher_Matrix = 0` can be a good choice if one is processing maps using masks, noise and fiducial identical to those used to process a previous map: in this case the Fisher matrix and the elements b_i are the same as the ones formerly computed.
- If `Compute_Spectrum = 0` the code does not compute the power spectrum. It only computes the Fisher Matrix, the coupled power and the noise bias, then saves the data. In this case, the user must compute the binned, eq. (3.12), or unbinned power spectrum, eq. (3.10), by their own means.
- The key `Binned` tells the program whether to compute a binned or unbinned estimation. In the first case, the program reads data from another section of the configuration file.

4.1.2 Maps

With the `keys` and, for example, the `values`

```
#Maps section

#Type of analysis
```



```
# 0 -> Auto-correlation spectrum
# 1 -> Cross-correlation spectrum
Type_of_Analysis = 0

#Number of maps to analyze
Number_of_Maps = 2001

#Type of maps file
#This key is only read by ECLIPSE_T and ECLIPSE_EB
#ECLIPSE_TEB ignores it
# 0 -> Maps of TQU
# 1 -> Maps of only T or only QU
Type_of_Data = 0

#Maps filename
Maps_FileName = MapsTQU.fits

#Maps to cross filename
Maps2Cross_FileName = MapsTQU2Cross.fits

#FWHM Beam (ArcMin)
Beam_FWHM = 131.922678213788

#Pixel window
# 1 -> Yes
# 0 -> No
Pixel_Window = 1
```

The keys tell the program the type of analysis to perform —auto or cross-correlation—, the number of maps to process, the name of the file containing the maps to read, the name of the file containing the maps to cross-correlate (required by the code only if the user wants to estimate cross-correlation), the FWHM size of the Gaussian beam (in arcmin) and whether the power in the maps has been reduced due to the pixel window function of the resolution.

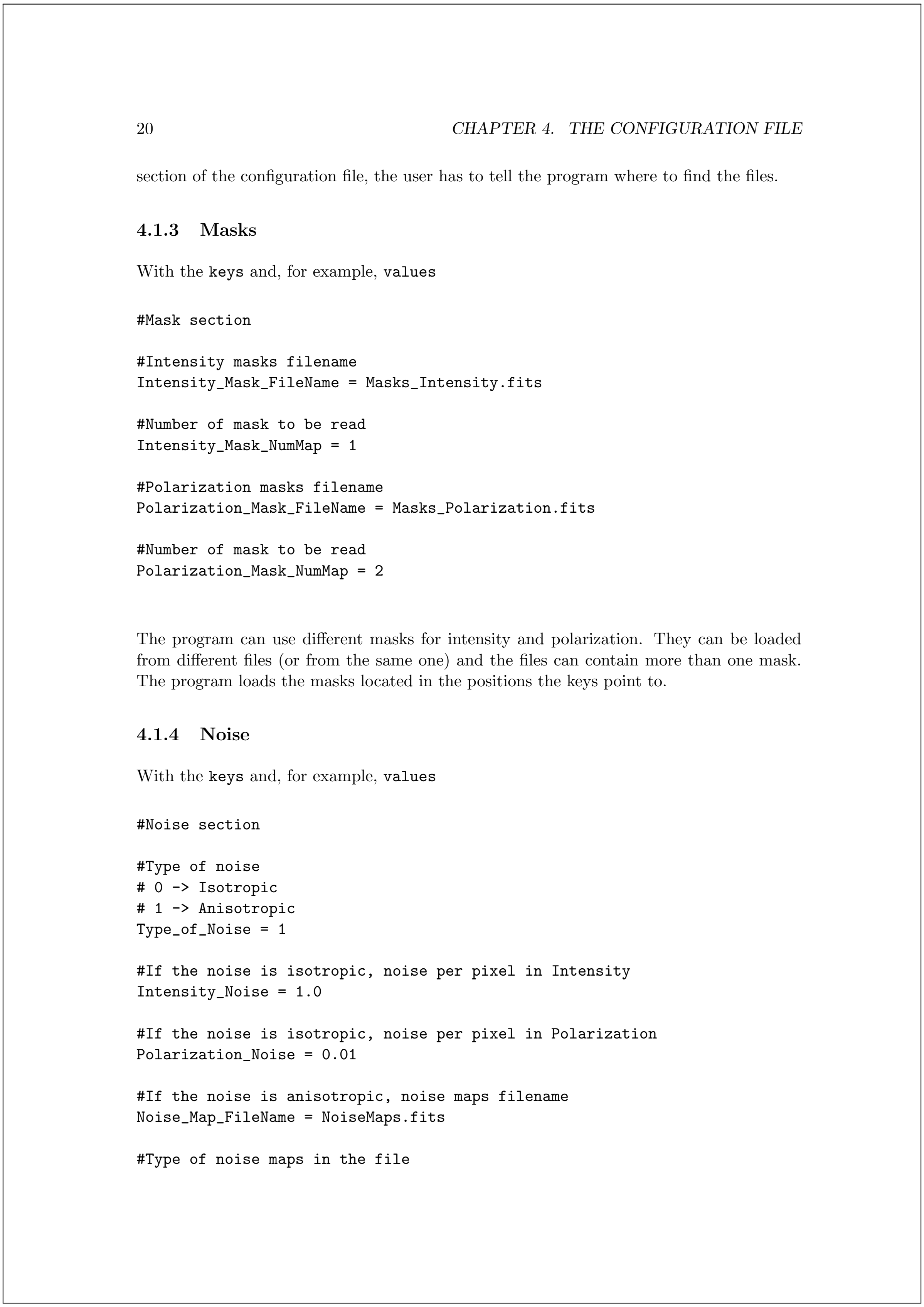
The key related to the data type functions is explained below

- `ECLIPSE_TEB` ignores the data type key since the code will undoubtedly expect maps containing TQU data.
- The codes `ECLIPSE_T` and `ECLIPSE_EB` read it. The key tells whether the file contains maps of TQU data or whether the file contains just T maps or just QU maps. That is, `ECLIPSE_EB` can read QU maps from a file containing just QU maps —in this case the key should be `Type_of_Data = 1`— or from a file containing TQU maps —in this case the key should be `Type_of_Data = 0`—. The key works the same for `ECLIPSE_T`.

If the key `Pixel_Window` has value 1, the program loads the data from the corresponding `HEALPix`[1] file (see [6]), according to the resolution fixed in the section above. In the last

---

[1] HEALPix



section of the configuration file, the user has to tell the program where to find the files.

4.1.3 Masks

With the `keys` and, for example, `values`

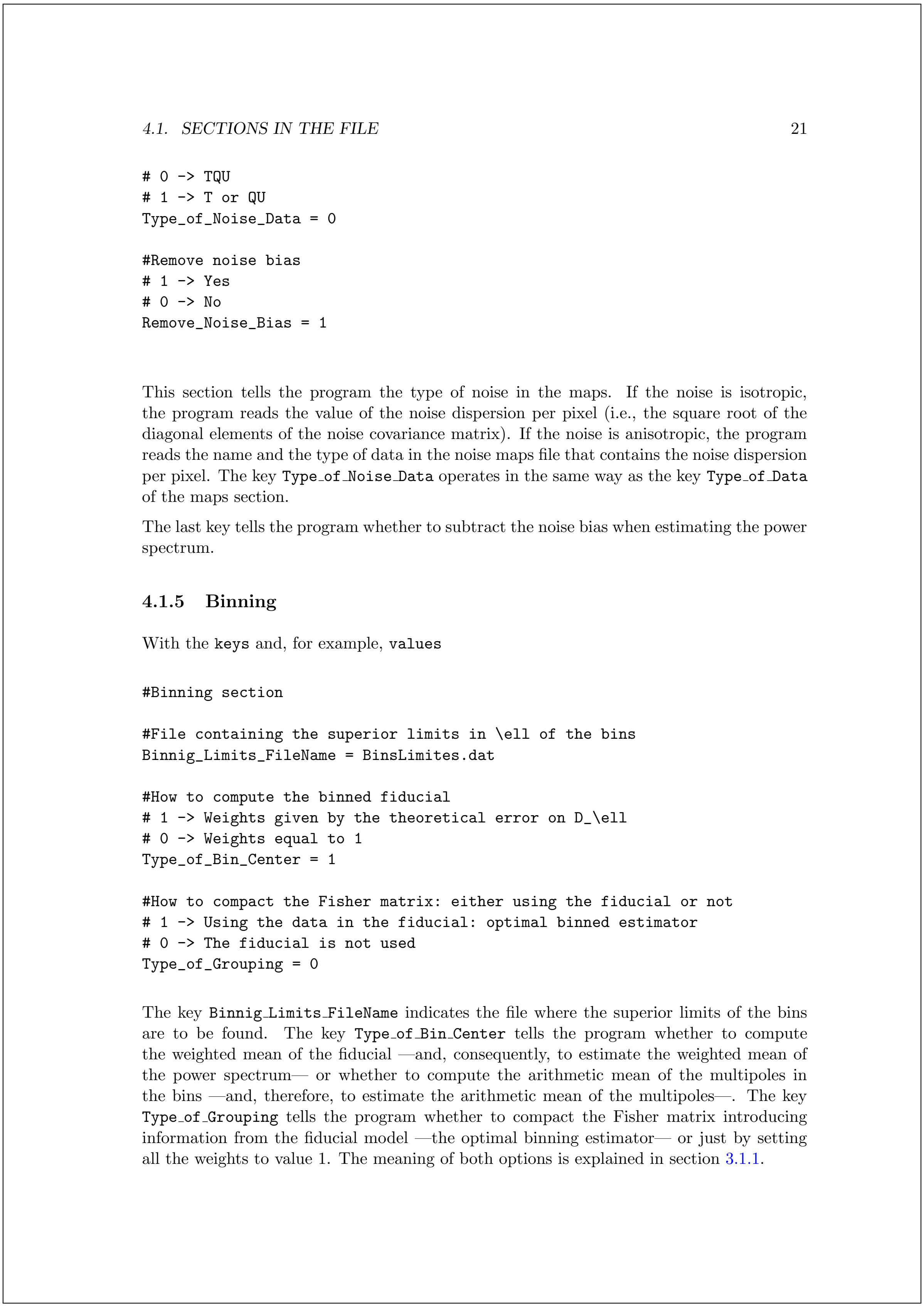
```
#Mask section

#Intensity masks filename
Intensity_Mask_FileName = Masks_Intensity.fits

#Number of mask to be read
Intensity_Mask_NumMap = 1

#Polarization masks filename
Polarization_Mask_FileName = Masks_Polarization.fits

#Number of mask to be read
Polarization_Mask_NumMap = 2
```

The program can use different masks for intensity and polarization. They can be loaded from different files (or from the same one) and the files can contain more than one mask. The program loads the masks located in the positions the keys point to.

4.1.4 Noise

With the `keys` and, for example, `values`

```
#Noise section

#Type of noise
# 0 -> Isotropic
# 1 -> Anisotropic
Type_of_Noise = 1

#If the noise is isotropic, noise per pixel in Intensity
Intensity_Noise = 1.0

#If the noise is isotropic, noise per pixel in Polarization
Polarization_Noise = 0.01

#If the noise is anisotropic, noise maps filename
Noise_Map_FileName = NoiseMaps.fits

#Type of noise maps in the file
```



```
# 0 -> TQU
# 1 -> T or QU
Type_of_Noise_Data = 0

#Remove noise bias
# 1 -> Yes
# 0 -> No
Remove_Noise_Bias = 1
```

This section tells the program the type of noise in the maps. If the noise is isotropic, the program reads the value of the noise dispersion per pixel (i.e., the square root of the diagonal elements of the noise covariance matrix). If the noise is anisotropic, the program reads the name and the type of data in the noise maps file that contains the noise dispersion per pixel. The key `Type_of_Noise_Data` operates in the same way as the key `Type_of_Data` of the maps section.

The last key tells the program whether to subtract the noise bias when estimating the power spectrum.

4.1.5 Binning

With the `keys` and, for example, `values`

```
#Binning section

#File containing the superior limits in \ell of the bins
Binnig_Limits_FileName = BinsLimites.dat

#How to compute the binned fiducial
# 1 -> Weights given by the theoretical error on D_\ell
# 0 -> Weights equal to 1
Type_of_Bin_Center = 1

#How to compact the Fisher matrix: either using the fiducial or not
# 1 -> Using the data in the fiducial: optimal binned estimator
# 0 -> The fiducial is not used
Type_of_Grouping = 0
```

The key `Binnig_Limits_FileName` indicates the file where the superior limits of the bins are to be found. The key `Type_of_Bin_Center` tells the program whether to compute the weighted mean of the fiducial —and, consequently, to estimate the weighted mean of the power spectrum— or whether to compute the arithmetic mean of the multipoles in the bins —and, therefore, to estimate the arithmetic mean of the multipoles—. The key `Type_of_Grouping` tells the program whether to compact the Fisher matrix introducing information from the fiducial model —the optimal binning estimator— or just by setting all the weights to value 1. The meaning of both options is explained in section 3.1.1.

4.1.6 Other information

That includes

```
#Size of the blocks of the distributed matrices
Matrices_Cyclic_Block_Size = 11

#Control of the regularity of the covariance matrix?
# 1 -> Yes
# 0 -> No
Inverse_Covariance_Matrix_Control = 1

#Show memory allocated by matrices?
# 1 -> Yes
# 0 -> No
Show_Memory_Allocated = 1

#Where to find the Pixel Window data
Healpix_Data_Folfer = '/home/<user>/Healpix_3.70/data/'
```

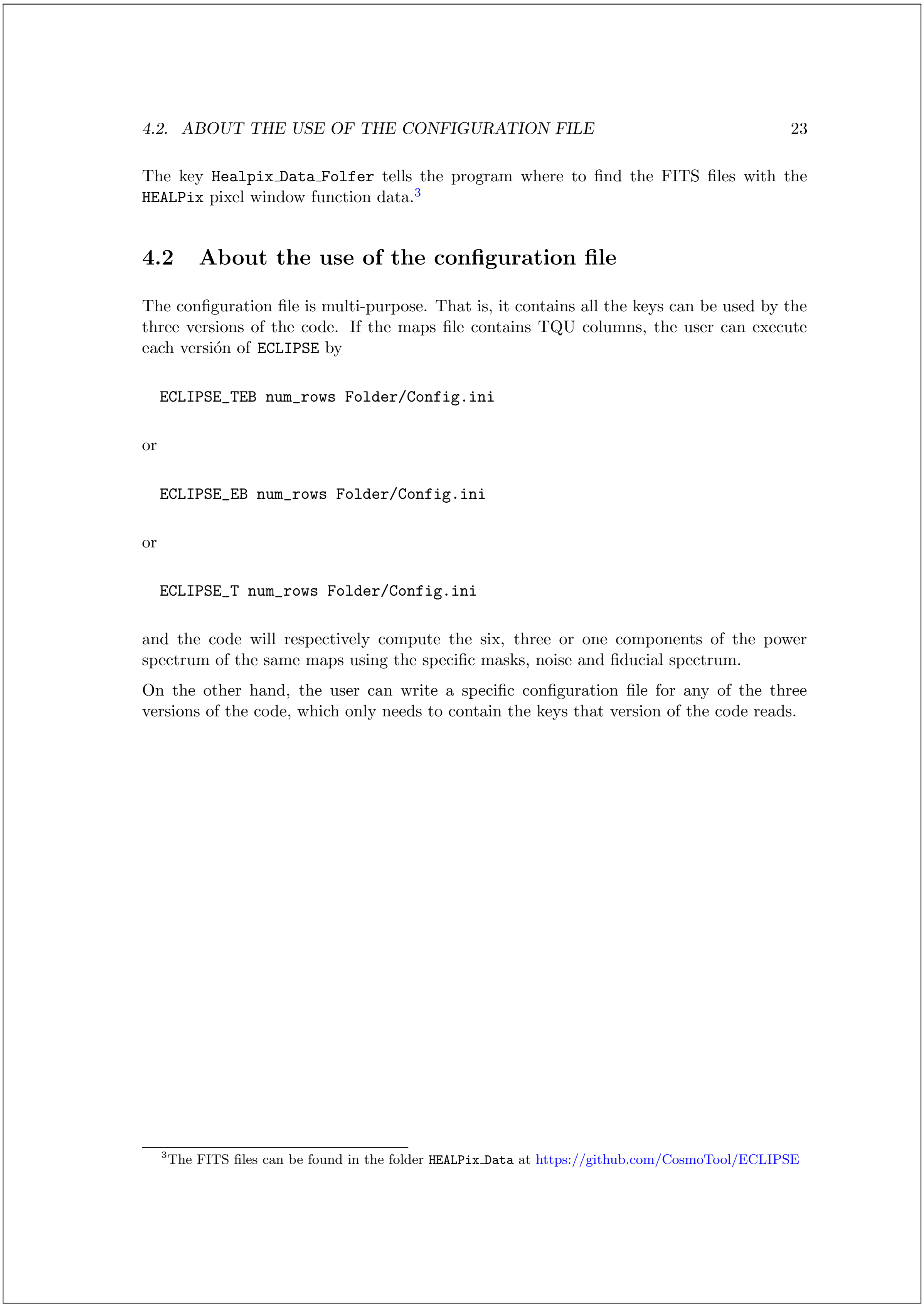
The first key tells the program the size of the blocks in which the matrices are distributed through the grid processors. `ECLIPSE` breaks the matrices into blocks of size `Matrices_Cyclic_Block_Size` and distributes them cyclically through the grid (more information in The Two-dimensional Block-Cyclic Distribution). The value of the key needs to be increased as the size of the matrices increases (with $N_{\rm side}$, with the number of observed pixels and/or with $\ell_{\rm max}$) and has to be such that each processor contains several blocks of the matrices. For example, the parameter cannot be greater than the number of rows in the covariance matrix divided by the number of rows in the grid of processors, otherwise the last rows of processors would not participate in the calculations. The user must take this point into account for two reasons: to achieve the highest efficiency in the calculations and because otherwise the program could crash.

The key `Inverse_Covariance_Matrix_Control` tells the program whether to check the regularity of the covariance matrix as explained in the section 7.1. If we are certain that the matrix is regular, the key can take the value 0. If we are not or just want to check it anyway, its value must be set to 1. The ScaLAPACK functions used to invert the matrices execute a control of the regularity, but our additional control is a good complement and we have observed it to be stronger.[2]

The key `Show_Memory_Allocated` tells the program to show, in different stages of the calculation, the memory allocated by the entire processors grid to store the matrices. This information gives a very good estimate of the memory required by the code in each step. The user must take into account that there are some stages of the computation where the memory used by the program is larger than the one directly allocated by the `ECLIPSE` code: this happens because ScaLAPACK internally allocates temporal memory to compute matricial operations.

---

[2] In some calculations the two ScaLAPACK functions returned value 0, but the product of the diagonal elements differed significantly from 1, which happens when the covariance matrix is in the limit singular/regular, evaluated in terms of floating point data.



The key `Healpix_Data_Folfer` tells the program where to find the FITS files with the `HEALPix` pixel window function data.[3]

4.2 About the use of the configuration file

The configuration file is multi-purpose. That is, it contains all the keys can be used by the three versions of the code. If the maps file contains TQU columns, the user can execute each versión of `ECLIPSE` by

```
ECLIPSE_TEB num_rows Folder/Config.ini
```

or

```
ECLIPSE_EB num_rows Folder/Config.ini
```

or

```
ECLIPSE_T num_rows Folder/Config.ini
```

and the code will respectively compute the six, three or one components of the power spectrum of the same maps using the specific masks, noise and fiducial spectrum.

On the other hand, the user can write a specific configuration file for any of the three versions of the code, which only needs to contain the keys that version of the code reads.

---

[3] The FITS files can be found in the folder `HEALPix_Data` at https://github.com/CosmoTool/ECLIPSE

Chapter 5

Input and output data files

In this section we describe in detail the format of the files which `ECLIPSE` works with.

5.1 Input files

The first file the program reads is the configuration file. The name and location are passed to the program when it is launched, as shown in section 2.2.

All the names of the input files are specified in the configuration file, whereas the names of the data files the program saves are fixed. The program expects to find and save them in the folder given by the key `Data_Folder`.

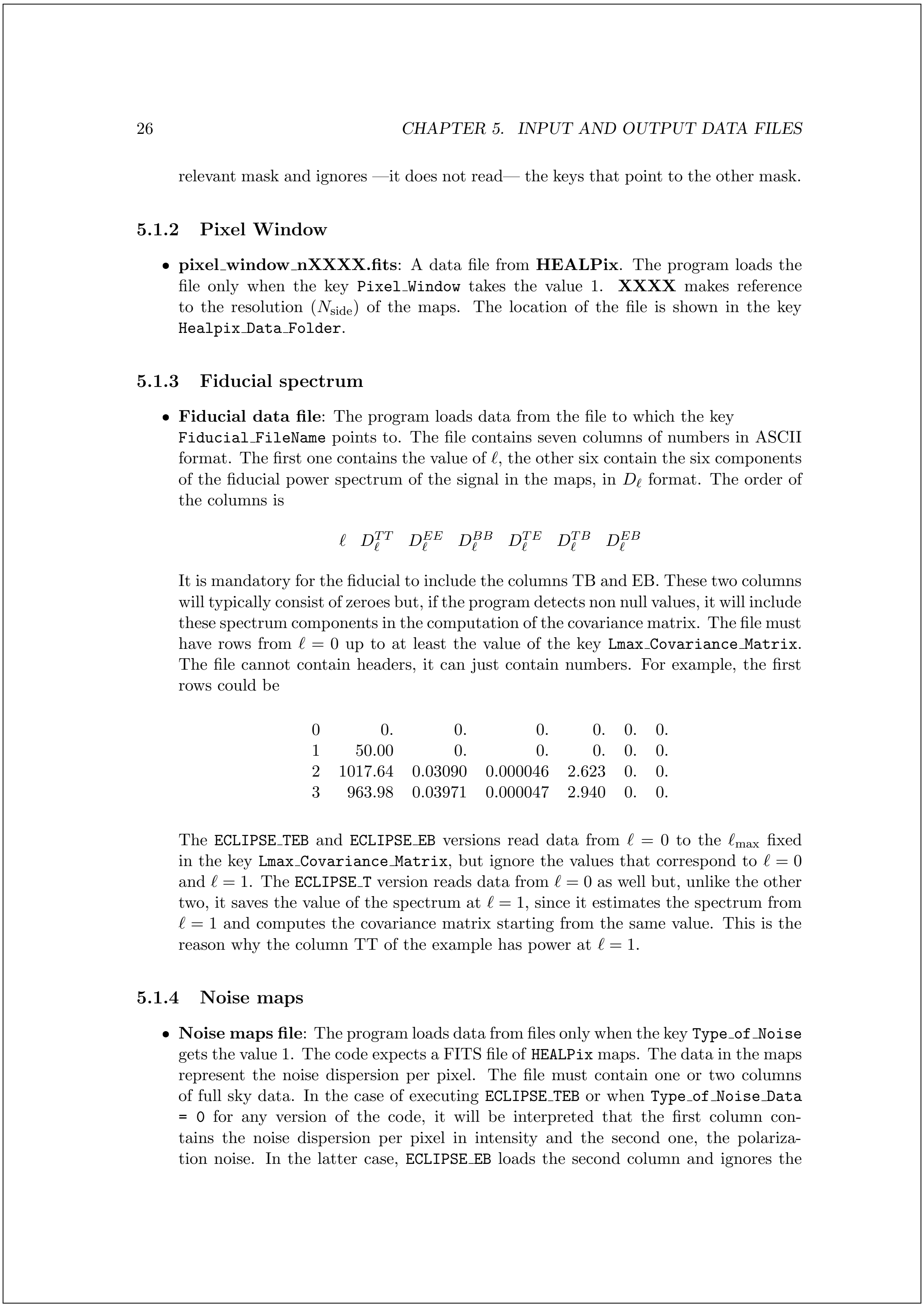
Referred to CMB, masks and noise maps, they are loaded as `HEALPix` FITS files. All maps (CMB, noise or mask) read by the program should be in full-sky format (even if null values are provided for those pixels that are discarded by the mask) and `RING` ordering. `ECLIPSE` first loads the masks for intensity and polarization and selects those pixels with value 1. Later, after loading the CMB and noise maps, only those pixels allowed by the mask will be kept in the computer´s memory.

The files and format the program reads are

5.1.1 Masks

- **Intensity mask file**: the file to which points the `Intensity_Mask_FileName` key, from which the program reads the intensity mask. It is a `HEALPix` maps file —FITS format— with type `REAL` data. The file can have several full sky maps. The maps are made of ones and zeroes (the NullVall of `HEALPix` is admitted, and the program interprets it as zero). The number of masks —maps— in the file must be equal to or greater than the value of the key `Intensity_Mask_NumMap`.
- **Polarization mask file**: the file to which points the `Polarization_Mask_FileName` key, from which the program reads the polarization mask. It has the same format as the previous file. The number of masks in the file must be equal to or greater than the value of the key `Polarization_Mask_NumMap`.

  The code `ECLIPSE_TEB` loads both masks. Each of the other two codes loads the



relevant mask and ignores —it does not read— the keys that point to the other mask.

5.1.2 Pixel Window

- **pixel_window_nXXXX.fits**: A data file from **HEALPix**. The program loads the file only when the key `Pixel_Window` takes the value 1. **XXXX** makes reference to the resolution (N_{side}) of the maps. The location of the file is shown in the key `Healpix_Data_Folder`.

5.1.3 Fiducial spectrum

- **Fiducial data file**: The program loads data from the file to which the key `Fiducial_FileName` points to. The file contains seven columns of numbers in ASCII format. The first one contains the value of ℓ, the other six contain the six components of the fiducial power spectrum of the signal in the maps, in D_ℓ format. The order of the columns is

$$\ell \quad D_\ell^{TT} \quad D_\ell^{EE} \quad D_\ell^{BB} \quad D_\ell^{TE} \quad D_\ell^{TB} \quad D_\ell^{EB}$$

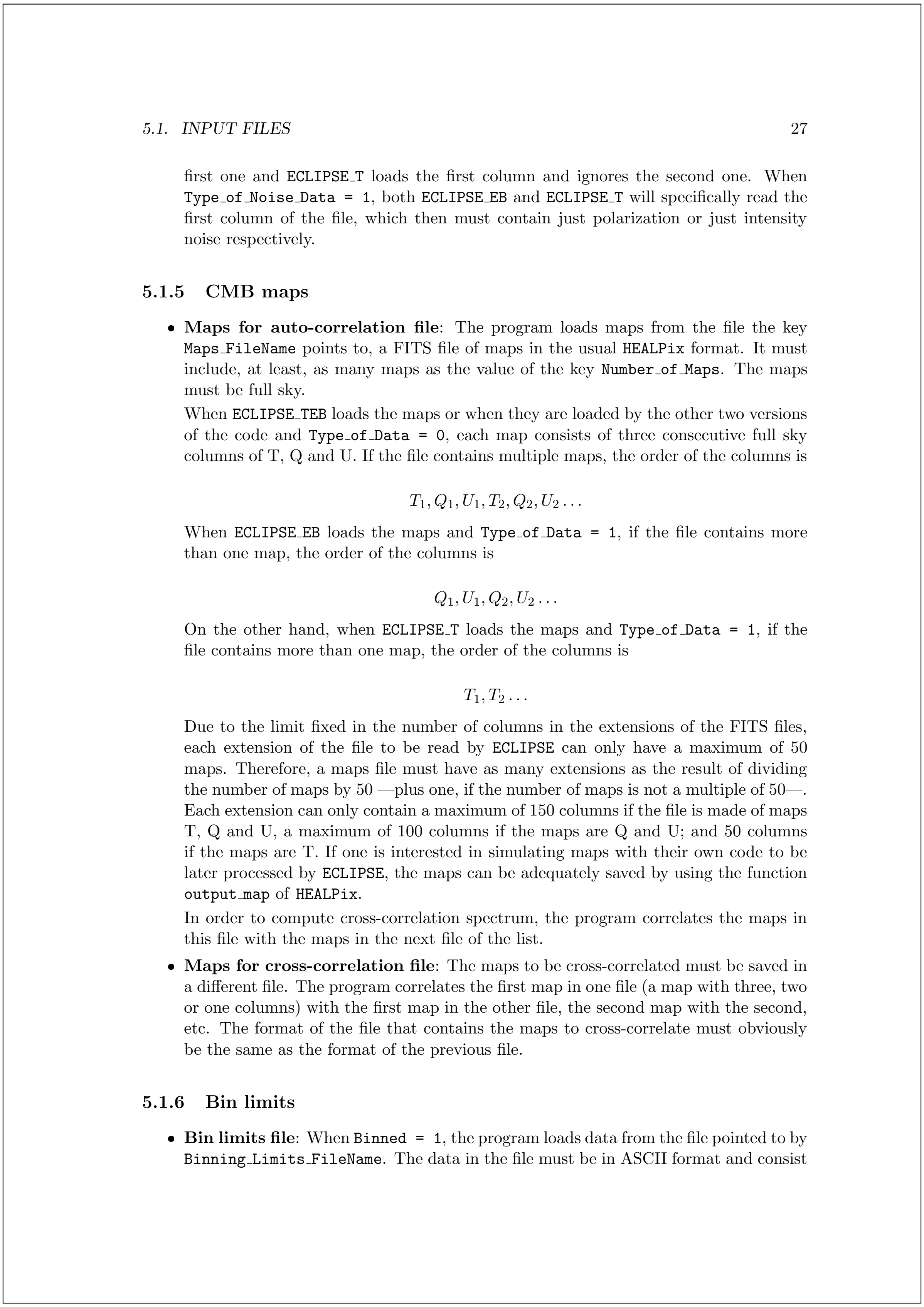
It is mandatory for the fiducial to include the columns TB and EB. These two columns will typically consist of zeroes but, if the program detects non null values, it will include these spectrum components in the computation of the covariance matrix. The file must have rows from $\ell = 0$ up to at least the value of the key `Lmax_Covariance_Matrix`. The file cannot contain headers, it can just contain numbers. For example, the first rows could be

0	0.	0.	0.	0.	0.	0.
1	50.00	0.	0.	0.	0.	0.
2	1017.64	0.03090	0.000046	2.623	0.	0.
3	963.98	0.03971	0.000047	2.940	0.	0.

The `ECLIPSE_TEB` and `ECLIPSE_EB` versions read data from $\ell = 0$ to the ℓ_{max} fixed in the key `Lmax_Covariance_Matrix`, but ignore the values that correspond to $\ell = 0$ and $\ell = 1$. The `ECLIPSE_T` version reads data from $\ell = 0$ as well but, unlike the other two, it saves the value of the spectrum at $\ell = 1$, since it estimates the spectrum from $\ell = 1$ and computes the covariance matrix starting from the same value. This is the reason why the column TT of the example has power at $\ell = 1$.

5.1.4 Noise maps

- **Noise maps file**: The program loads data from files only when the key `Type_of_Noise` gets the value 1. The code expects a FITS file of `HEALPix` maps. The data in the maps represent the noise dispersion per pixel. The file must contain one or two columns of full sky data. In the case of executing `ECLIPSE_TEB` or when `Type_of_Noise_Data = 0` for any version of the code, it will be interpreted that the first column contains the noise dispersion per pixel in intensity and the second one, the polarization noise. In the latter case, `ECLIPSE_EB` loads the second column and ignores the



first one and `ECLIPSE_T` loads the first column and ignores the second one. When `Type_of_Noise_Data = 1`, both `ECLIPSE_EB` and `ECLIPSE_T` will specifically read the first column of the file, which then must contain just polarization or just intensity noise respectively.

5.1.5 CMB maps

- **Maps for auto-correlation file**: The program loads maps from the file the key `Maps_FileName` points to, a FITS file of maps in the usual `HEALPix` format. It must include, at least, as many maps as the value of the key `Number_of_Maps`. The maps must be full sky.

  When `ECLIPSE_TEB` loads the maps or when they are loaded by the other two versions of the code and `Type_of_Data = 0`, each map consists of three consecutive full sky columns of T, Q and U. If the file contains multiple maps, the order of the columns is

  $$T_1, Q_1, U_1, T_2, Q_2, U_2 \ldots$$

  When `ECLIPSE_EB` loads the maps and `Type_of_Data = 1`, if the file contains more than one map, the order of the columns is

  $$Q_1, U_1, Q_2, U_2 \ldots$$

  On the other hand, when `ECLIPSE_T` loads the maps and `Type_of_Data = 1`, if the file contains more than one map, the order of the columns is

  $$T_1, T_2 \ldots$$

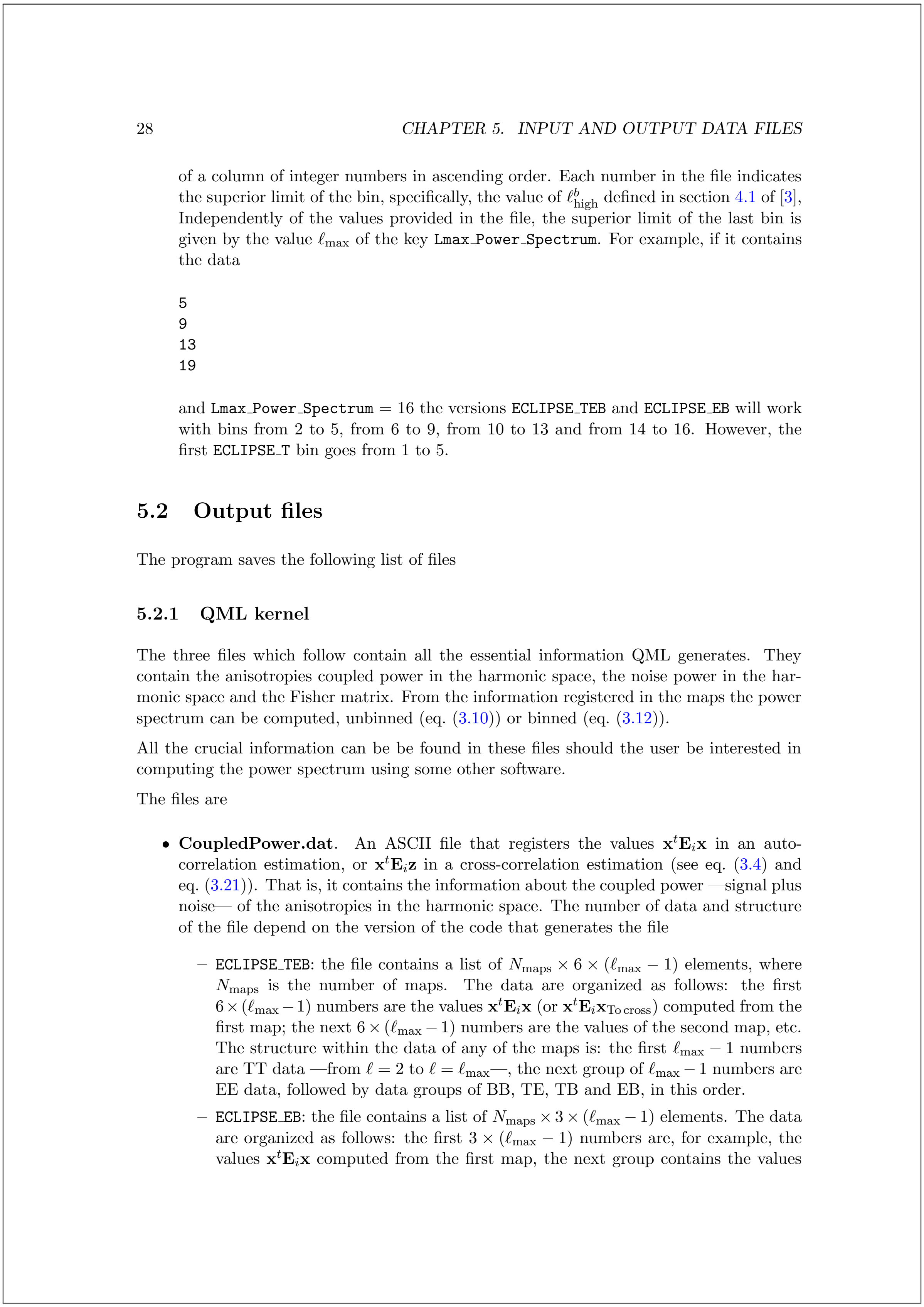
  Due to the limit fixed in the number of columns in the extensions of the FITS files, each extension of the file to be read by `ECLIPSE` can only have a maximum of 50 maps. Therefore, a maps file must have as many extensions as the result of dividing the number of maps by 50 —plus one, if the number of maps is not a multiple of 50—. Each extension can only contain a maximum of 150 columns if the file is made of maps T, Q and U, a maximum of 100 columns if the maps are Q and U; and 50 columns if the maps are T. If one is interested in simulating maps with their own code to be later processed by `ECLIPSE`, the maps can be adequately saved by using the function `output_map` of `HEALPix`.

  In order to compute cross-correlation spectrum, the program correlates the maps in this file with the maps in the next file of the list.

- **Maps for cross-correlation file**: The maps to be cross-correlated must be saved in a different file. The program correlates the first map in one file (a map with three, two or one columns) with the first map in the other file, the second map with the second, etc. The format of the file that contains the maps to cross-correlate must obviously be the same as the format of the previous file.

5.1.6 Bin limits

- **Bin limits file**: When `Binned = 1`, the program loads data from the file pointed to by `Binning_Limits_FileName`. The data in the file must be in ASCII format and consist



of a column of integer numbers in ascending order. Each number in the file indicates the superior limit of the bin, specifically, the value of $\ell^b_{\rm high}$ defined in section 4.1 of [3], Independently of the values provided in the file, the superior limit of the last bin is given by the value $\ell_{\rm max}$ of the key `Lmax_Power_Spectrum`. For example, if it contains the data

```
5
9
13
19
```

and `Lmax_Power_Spectrum` $= 16$ the versions `ECLIPSE_TEB` and `ECLIPSE_EB` will work with bins from 2 to 5, from 6 to 9, from 10 to 13 and from 14 to 16. However, the first `ECLIPSE_T` bin goes from 1 to 5.

5.2 Output files

The program saves the following list of files

5.2.1 QML kernel

The three files which follow contain all the essential information QML generates. They contain the anisotropies coupled power in the harmonic space, the noise power in the harmonic space and the Fisher matrix. From the information registered in the maps the power spectrum can be computed, unbinned (eq. (3.10)) or binned (eq. (3.12)).

All the crucial information can be be found in these files should the user be interested in computing the power spectrum using some other software.

The files are

- **CoupledPower.dat**. An ASCII file that registers the values $\mathbf{x}^t\mathbf{E}_i\mathbf{x}$ in an auto-correlation estimation, or $\mathbf{x}^t\mathbf{E}_i\mathbf{z}$ in a cross-correlation estimation (see eq. (3.4) and eq. (3.21)). That is, it contains the information about the coupled power —signal plus noise— of the anisotropies in the harmonic space. The number of data and structure of the file depend on the version of the code that generates the file
    - `ECLIPSE_TEB`: the file contains a list of $N_{\rm maps} \times 6 \times (\ell_{\rm max} - 1)$ elements, where $N_{\rm maps}$ is the number of maps. The data are organized as follows: the first $6 \times (\ell_{\rm max} - 1)$ numbers are the values $\mathbf{x}^t\mathbf{E}_i\mathbf{x}$ (or $\mathbf{x}^t\mathbf{E}_i\mathbf{x}_{\rm To\,cross}$) computed from the first map; the next $6 \times (\ell_{\rm max} - 1)$ numbers are the values of the second map, etc. The structure within the data of any of the maps is: the first $\ell_{\rm max} - 1$ numbers are TT data —from $\ell = 2$ to $\ell = \ell_{\rm max}$—, the next group of $\ell_{\rm max} - 1$ numbers are EE data, followed by data groups of BB, TE, TB and EB, in this order.
    - `ECLIPSE_EB`: the file contains a list of $N_{\rm maps} \times 3 \times (\ell_{\rm max} - 1)$ elements. The data are organized as follows: the first $3 \times (\ell_{\rm max} - 1)$ numbers are, for example, the values $\mathbf{x}^t\mathbf{E}_i\mathbf{x}$ computed from the first map, the next group contains the values

obtained from the second map, etc. The structure within the data of any of the maps is: the first $\ell_{\max} - 1$ numbers are EE data —from $\ell = 2$ to $\ell = \ell_{\max}$—, the next group of $\ell_{\max} - 1$ numbers are BB data, and the last group of $\ell_{\max} - 1$ numbers are EB data.

  - `ECLIPSE_T`: the file contains a list of $N_{\text{maps}} \times \ell_{\max}$ elements. The data are organized as follows: the first $\ell_{\max}$ numbers are, for example, the values $\mathbf{x}^t \mathbf{E}_i \mathbf{x}$ computed from the first map —ordered from $\ell = 1$ to $\ell_{\max}$—; the next group contains the values obtained from the second map, etc.

- **NoiseBias.dat**. A file in ASCII format. It contains the values of eq. (3.6). Depending on the program that saves the file, the data are
  - `ECLIPSE_TEB`: a list of $6 \times (\ell_{\max} - 1)$ elements. The first $\ell_{\max} - 1$ elements are the TT values —ordered from $\ell = 2$ to $\ell_{\max}$ —. The rest of the data are the EE, BB, TE, TB and EB values, in this order.
  - `ECLIPSE_EB`: a list of $3 \times (\ell_{\max} - 1)$ elements. The first $\ell_{\max} - 1$ elements are the EE values —ordered from $\ell = 2$ to $\ell_{\max}$ —. The rest of the data are the BB, and EB values.
  - `ECLIPSE_T`: a list of $\ell_{\max}$ elements, the values of TT ordered from $\ell = 1$ to $\ell_{\max}$.

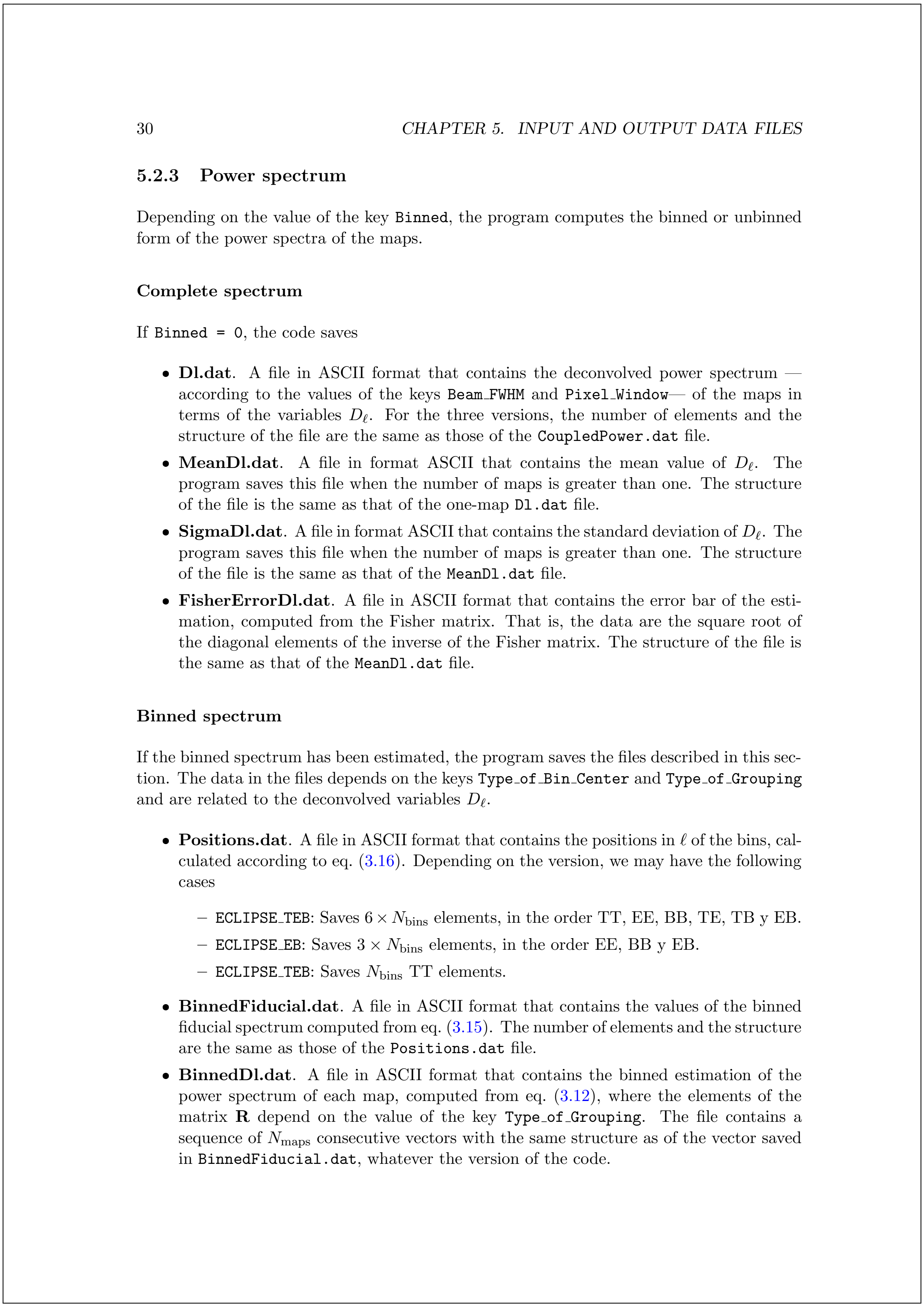
- **FisherMatrix.dat**. A file in ASCII format that contains the data given by eq. (3.9). The saved data do not conform a matrix structure, they must be interpreted as a simple list of numbers whose structure depends on the version that saves the file
  - `ECLIPSE_TEB`: the file — the list of numbers — contains $[6 \times (\ell_{\max} - 1)] \times [6 \times (\ell_{\max} - 1)]$ elements. Each group of $6 \times (\ell_{\max} - 1)$ numbers makes up a row of the matrix. The rows (and columns) of the matrix are organized as follows: six groups of $\ell_{\max} - 1$ numbers. The groups are ordered: TT, EE, BB, TE, TB, EB. The $\ell_{\max} - 1$ numbers in each group range from $\ell = 2$ to $\ell_{\max}$.
  - `ECLIPSE_EB`: the file contains $[3 \times (\ell_{\max} - 1)] \times [3 \times (\ell_{\max} - 1)]$ numbers. Each group of $3 \times (\ell_{\max} - 1)$ numbers makes up a row of the matrix. The rows (and columns) of the matrix are organized as follows: three groups of $\ell_{\max} - 1$ numbers. The groups are ordered: EE, BB, EB. The $\ell_{\max} - 1$ numbers in each group range from $\ell = 2$ to $\ell_{\max}$.
  - `ECLIPSE_T`: the file contains $\ell_{\max} \times \ell_{\max}$ numbers. Each group of $\ell_{\max}$ numbers makes up a row of the matrix, from $\ell = 1$ to $\ell_{\max}$.

  `ECLIPSE` saves this file when the key `Compute_Fisher_Matrix` takes the value 1.

5.2.2 Control

The program computes and saves $\mathbf{x}^t \mathbf{C}^{-1} \mathbf{x}$, a list of N_{maps} numbers. This data can be used to check the concordance between the maps and the covariance matrix. If cross-correlation is computed, the program saves $\mathbf{x}^t C^{-1} \mathbf{z}$, where $\mathbf{z}$ is the map to cross. Therefore, another file saved by the program is

- **ChiSquare.dat**. A file in ASCII format made of N_{maps} numbers.



5.2.3 Power spectrum

Depending on the value of the key `Binned`, the program computes the binned or unbinned form of the power spectra of the maps.

Complete spectrum

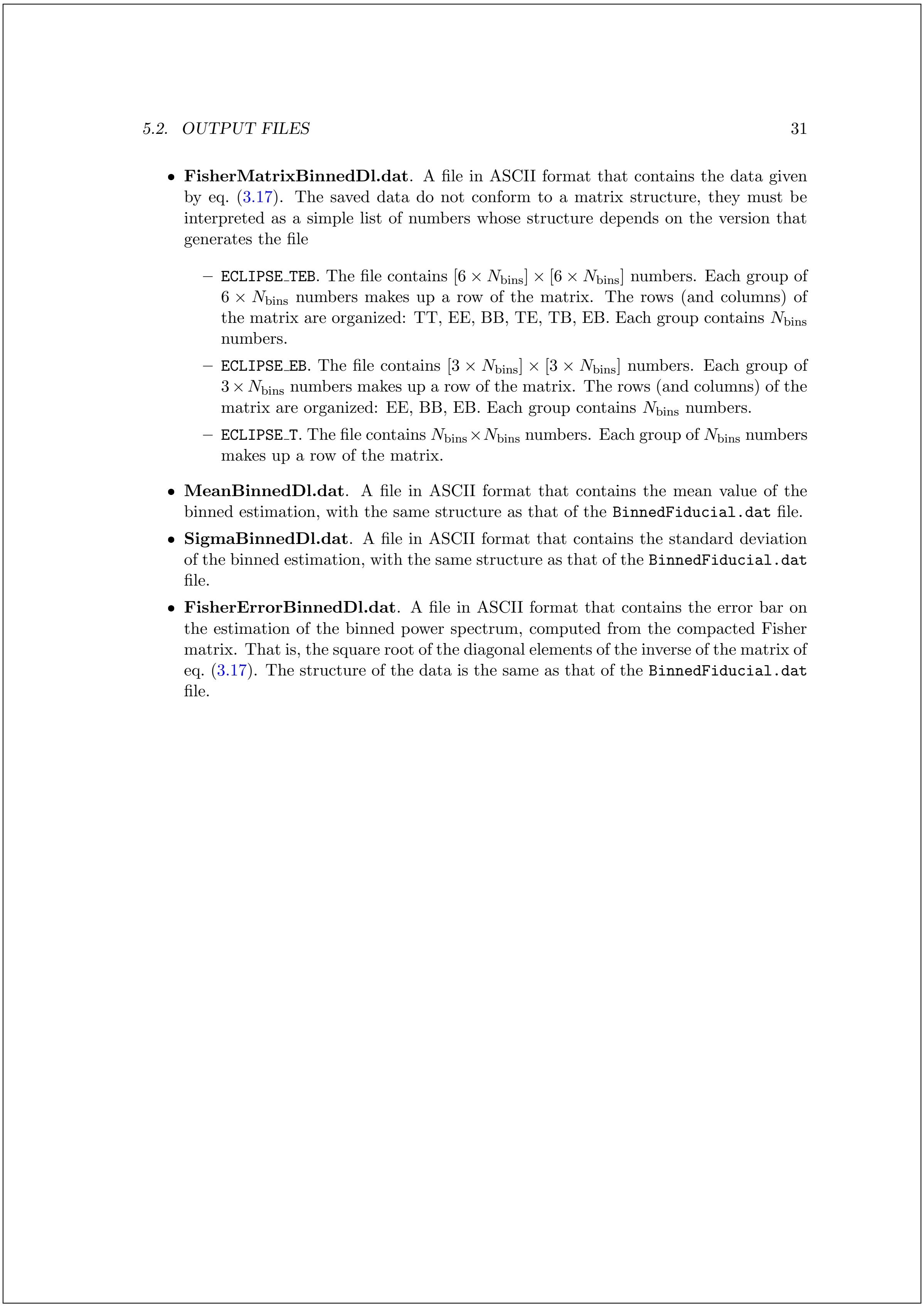
If `Binned = 0`, the code saves

- **Dl.dat**. A file in ASCII format that contains the deconvolved power spectrum — according to the values of the keys `Beam_FWHM` and `Pixel_Window`— of the maps in terms of the variables D_ℓ. For the three versions, the number of elements and the structure of the file are the same as those of the `CoupledPower.dat` file.
- **MeanDl.dat**. A file in format ASCII that contains the mean value of D_ℓ. The program saves this file when the number of maps is greater than one. The structure of the file is the same as that of the one-map `Dl.dat` file.
- **SigmaDl.dat**. A file in format ASCII that contains the standard deviation of D_ℓ. The program saves this file when the number of maps is greater than one. The structure of the file is the same as that of the `MeanDl.dat` file.
- **FisherErrorDl.dat**. A file in ASCII format that contains the error bar of the estimation, computed from the Fisher matrix. That is, the data are the square root of the diagonal elements of the inverse of the Fisher matrix. The structure of the file is the same as that of the `MeanDl.dat` file.

Binned spectrum

If the binned spectrum has been estimated, the program saves the files described in this section. The data in the files depends on the keys `Type_of_Bin_Center` and `Type_of_Grouping` and are related to the deconvolved variables D_ℓ.

- **Positions.dat**. A file in ASCII format that contains the positions in ℓ of the bins, calculated according to eq. (3.16). Depending on the version, we may have the following cases
  - `ECLIPSE_TEB`: Saves $6 \times N_{\text{bins}}$ elements, in the order TT, EE, BB, TE, TB y EB.
  - `ECLIPSE_EB`: Saves $3 \times N_{\text{bins}}$ elements, in the order EE, BB y EB.
  - `ECLIPSE_TEB`: Saves N_{bins} TT elements.
- **BinnedFiducial.dat**. A file in ASCII format that contains the values of the binned fiducial spectrum computed from eq. (3.15). The number of elements and the structure are the same as those of the `Positions.dat` file.
- **BinnedDl.dat**. A file in ASCII format that contains the binned estimation of the power spectrum of each map, computed from eq. (3.12), where the elements of the matrix $\mathbf{R}$ depend on the value of the key `Type_of_Grouping`. The file contains a sequence of N_{maps} consecutive vectors with the same structure as of the vector saved in `BinnedFiducial.dat`, whatever the version of the code.



- **FisherMatrixBinnedDl.dat**. A file in ASCII format that contains the data given by eq. (3.17). The saved data do not conform to a matrix structure, they must be interpreted as a simple list of numbers whose structure depends on the version that generates the file
  - `ECLIPSE_TEB`. The file contains $[6 \times N_{\rm bins}] \times [6 \times N_{\rm bins}]$ numbers. Each group of $6 \times N_{\rm bins}$ numbers makes up a row of the matrix. The rows (and columns) of the matrix are organized: TT, EE, BB, TE, TB, EB. Each group contains $N_{\rm bins}$ numbers.
  - `ECLIPSE_EB`. The file contains $[3 \times N_{\rm bins}] \times [3 \times N_{\rm bins}]$ numbers. Each group of $3 \times N_{\rm bins}$ numbers makes up a row of the matrix. The rows (and columns) of the matrix are organized: EE, BB, EB. Each group contains $N_{\rm bins}$ numbers.
  - `ECLIPSE_T`. The file contains $N_{\rm bins} \times N_{\rm bins}$ numbers. Each group of $N_{\rm bins}$ numbers makes up a row of the matrix.
- **MeanBinnedDl.dat**. A file in ASCII format that contains the mean value of the binned estimation, with the same structure as that of the `BinnedFiducial.dat` file.
- **SigmaBinnedDl.dat**. A file in ASCII format that contains the standard deviation of the binned estimation, with the same structure as that of the `BinnedFiducial.dat` file.
- **FisherErrorBinnedDl.dat**. A file in ASCII format that contains the error bar on the estimation of the binned power spectrum, computed from the compacted Fisher matrix. That is, the square root of the diagonal elements of the inverse of the matrix of eq. (3.17). The structure of the data is the same as that of the `BinnedFiducial.dat` file.

Chapter 6

Computer requirements

Depending on the maps dimension and the $\ell_{\max}$ up to which to compute the power spectrum, the dimensions of the matrices the code stores can vary significantly. The user needs some rules to determine the memory required and, consequently, the number of processors. The package `ECLIPSE` contains a small program, `ECLIPSE_Memory.f90`, which reads any given configuration file and —according to eq. (6.2)-(6.9)— tells the user the memory required by all three versions of the code to store matrices at two crucial steps of the computation. For example, if typing

```
> ./ECLIPSE_Memory NSide64/Example.ini
```

the program shows

```
Loading configuration from: NSide64/TEB.ini
                        Data_Folder:   NSide64
                              NSide:   64
                Lmax_Power_Spectrum:   192
            Intensity_Mask_FileName:   MascaraSateliteN64.fits
              Intensity_Mask_NumMap:   1
         Polarization_Mask_FileName:   MascaraSateliteN64.fits
           Polarization_Mask_NumMap:   1
  Inverse_Covariance_Matrix_Control:   1
*******************************************************************
Loading masks
Number of observed pixels in temperature:                  29009
Number of observed pixels in polarization:                 29009
Lmax power spectrum:                                         192
Number of spherical harmonics:                             37245
*******************************************************************
*******************************************************************
ECLIPSE_TEB - [Gb]
Eq. (6.1):    112.856905594468
Eq. (6.2):    222.766825564206
Eq. (6.3):    252.822962254286
*******************************************************************
ECLIPSE_EB - [Gb]
Eq. (6.4):    50.1586247086525
Eq. (6.5):    121.678194284439
```

```
Eq. (6.6):    190.810735747218
*****************************************************************
ECLIPSE_T - [Gb]
Eq. (6.7):    12.5396561771631
Eq. (6.8):    30.4214937761426
Eq. (6.9):    52.8762931823730
Number of spherical harmonics:                              37248
*****************************************************************
```

That is, the program reads the configuration file, determines the number of pixels allowed by the intensity and polarization masks and takes the value of $\ell_{\max}$ up to which the power spectrum must be estimated. From these values the program calculates the memory required to store matrices —at two stages of the computation— by all three versions of the code. In the example, if one needs to estimate the six components of the power spectrum in a case determined by the configuration file `Example.ini`, at least 7.36 Gb of memory are required; if one needs to compute the intensity power spectrum, at least 1.17 Gb are required.

Note that these values only consider the memory required to store the matrices, and not the memory required to run the code. However, they can be taken as a very valuable reference to determine the total memory that needs to be requested.

In order to build `ECLIPSE_Memory.f90`, the user has to run the same script which builds `ECLIPSE`. For example, in NERSC

```
> ./cCrayNERSC.sh ECLIPSE_Memory
```

For the interested readed, in the next section we explain where these numbers come from. However, if not interested in the details, one can go directly to section 6.2.

6.1 Determining the memory required

The size of the matrices depends on three principal numbers

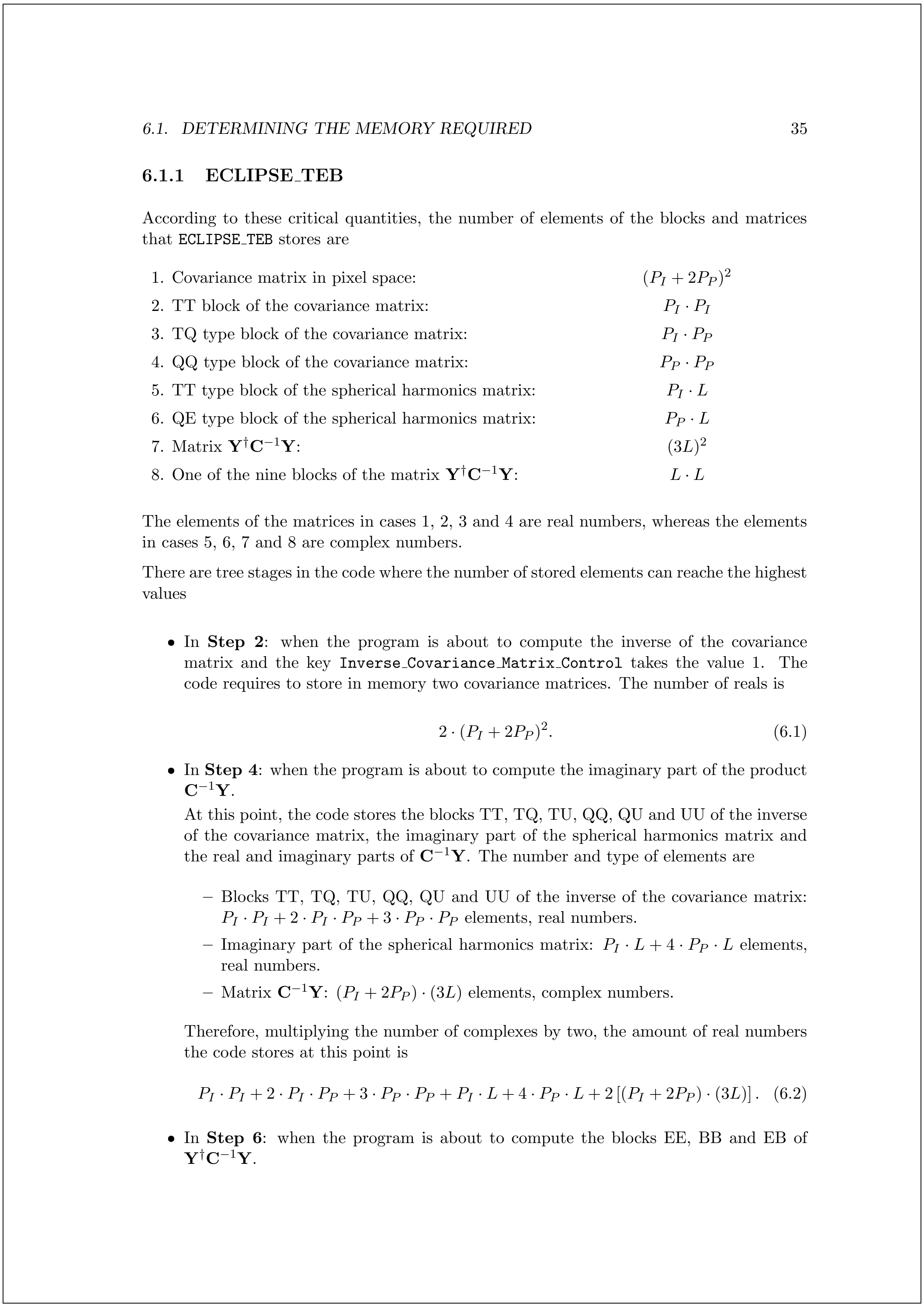
- Number of observed pixels in the intensity maps: P_I.
- Number of observed pixels in the polarization maps: P_P.
- Multipole up to which to compute the power spectrum: $\ell_{\max}$.

When the code estimates polarization components, the number of spherical harmonics, L, depends on $\ell_{\max}$

$$L = \sum_{\ell=2}^{\ell_{\max}} 2\ell + 1.$$

The summation starts from $\ell = 1$ when the code estimates only intensity.

In the following sections we detail the critical stages regarding memory requirements for the three versions of `ECLIPSE`.



6.1.1 ECLIPSE_TEB

According to these critical quantities, the number of elements of the blocks and matrices that `ECLIPSE_TEB` stores are

1. Covariance matrix in pixel space:	$(P_I + 2P_P)^2$
2. TT block of the covariance matrix:	$P_I \cdot P_I$
3. TQ type block of the covariance matrix:	$P_I \cdot P_P$
4. QQ type block of the covariance matrix:	$P_P \cdot P_P$
5. TT type block of the spherical harmonics matrix:	$P_I \cdot L$
6. QE type block of the spherical harmonics matrix:	$P_P \cdot L$
7. Matrix $\mathbf{Y}^\dagger \mathbf{C}^{-1} \mathbf{Y}$:	$(3L)^2$
8. One of the nine blocks of the matrix $\mathbf{Y}^\dagger \mathbf{C}^{-1} \mathbf{Y}$:	$L \cdot L$

The elements of the matrices in cases 1, 2, 3 and 4 are real numbers, whereas the elements in cases 5, 6, 7 and 8 are complex numbers.

There are tree stages in the code where the number of stored elements can reache the highest values

- In **Step 2**: when the program is about to compute the inverse of the covariance matrix and the key `Inverse_Covariance_Matrix_Control` takes the value 1. The code requires to store in memory two covariance matrices. The number of reals is

  $$2 \cdot (P_I + 2P_P)^2. \tag{6.1}$$

- In **Step 4**: when the program is about to compute the imaginary part of the product $\mathbf{C}^{-1}\mathbf{Y}$.

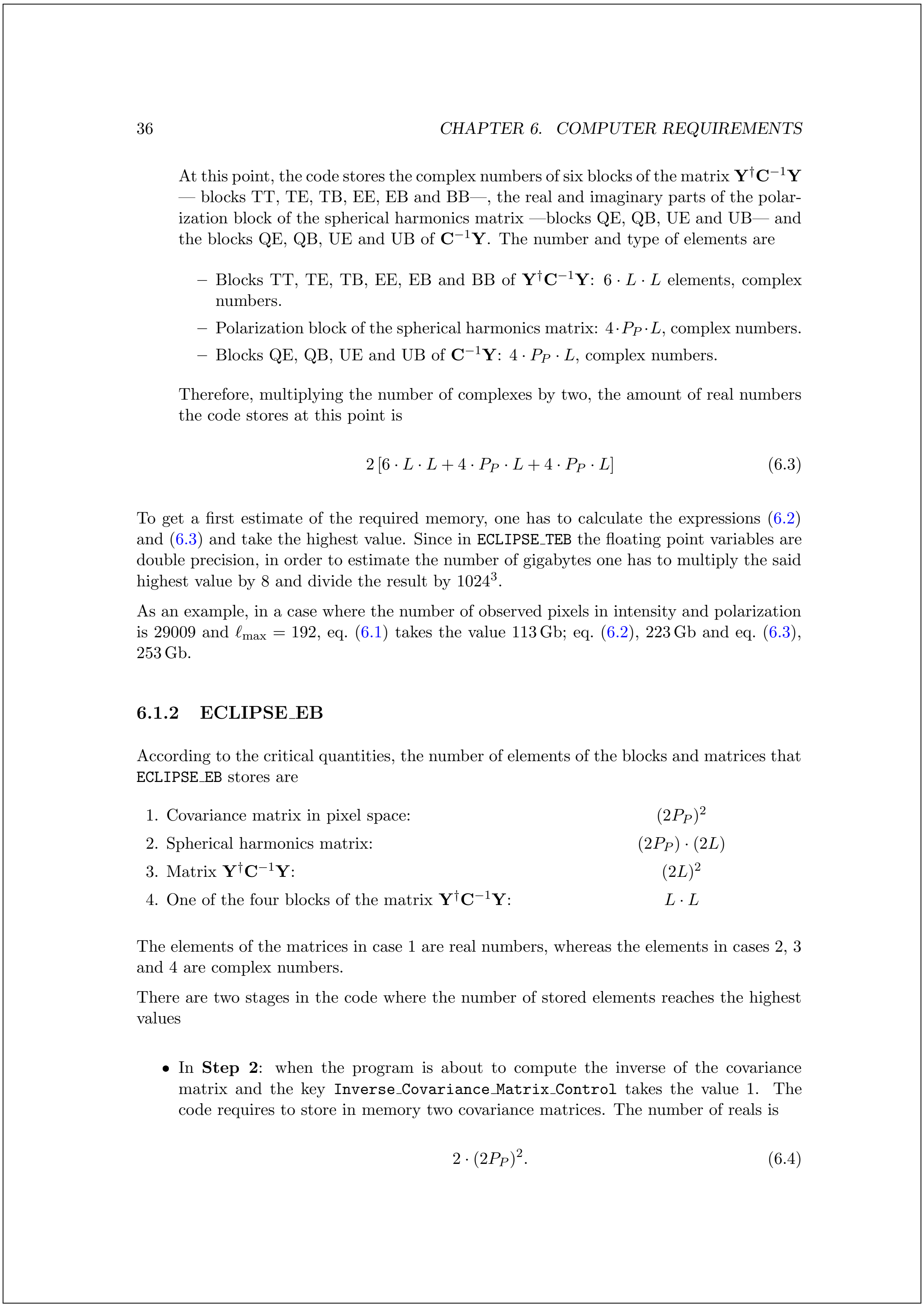
  At this point, the code stores the blocks TT, TQ, TU, QQ, QU and UU of the inverse of the covariance matrix, the imaginary part of the spherical harmonics matrix and the real and imaginary parts of $\mathbf{C}^{-1}\mathbf{Y}$. The number and type of elements are

  - Blocks TT, TQ, TU, QQ, QU and UU of the inverse of the covariance matrix: $P_I \cdot P_I + 2 \cdot P_I \cdot P_P + 3 \cdot P_P \cdot P_P$ elements, real numbers.
  - Imaginary part of the spherical harmonics matrix: $P_I \cdot L + 4 \cdot P_P \cdot L$ elements, real numbers.
  - Matrix $\mathbf{C}^{-1}\mathbf{Y}$: $(P_I + 2P_P) \cdot (3L)$ elements, complex numbers.

  Therefore, multiplying the number of complexes by two, the amount of real numbers the code stores at this point is

  $$P_I \cdot P_I + 2 \cdot P_I \cdot P_P + 3 \cdot P_P \cdot P_P + P_I \cdot L + 4 \cdot P_P \cdot L + 2\left[(P_I + 2P_P) \cdot (3L)\right]. \tag{6.2}$$

- In **Step 6**: when the program is about to compute the blocks EE, BB and EB of $\mathbf{Y}^\dagger \mathbf{C}^{-1} \mathbf{Y}$.



36 CHAPTER 6. COMPUTER REQUIREMENTS

At this point, the code stores the complex numbers of six blocks of the matrix $\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y}$ — blocks TT, TE, TB, EE, EB and BB—, the real and imaginary parts of the polarization block of the spherical harmonics matrix —blocks QE, QB, UE and UB— and the blocks QE, QB, UE and UB of $\mathbf{C}^{-1}\mathbf{Y}$. The number and type of elements are

- Blocks TT, TE, TB, EE, EB and BB of $\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y}$: $6 \cdot L \cdot L$ elements, complex numbers.
- Polarization block of the spherical harmonics matrix: $4 \cdot P_P \cdot L$, complex numbers.
- Blocks QE, QB, UE and UB of $\mathbf{C}^{-1}\mathbf{Y}$: $4 \cdot P_P \cdot L$, complex numbers.

Therefore, multiplying the number of complexes by two, the amount of real numbers the code stores at this point is

$$2\left[6 \cdot L \cdot L + 4 \cdot P_P \cdot L + 4 \cdot P_P \cdot L\right] \tag{6.3}$$

To get a first estimate of the required memory, one has to calculate the expressions (6.2) and (6.3) and take the highest value. Since in `ECLIPSE_TEB` the floating point variables are double precision, in order to estimate the number of gigabytes one has to multiply the said highest value by 8 and divide the result by 1024^3.

As an example, in a case where the number of observed pixels in intensity and polarization is 29009 and $\ell_{\max} = 192$, eq. (6.1) takes the value 113 Gb; eq. (6.2), 223 Gb and eq. (6.3), 253 Gb.

6.1.2 ECLIPSE_EB

According to the critical quantities, the number of elements of the blocks and matrices that `ECLIPSE_EB` stores are

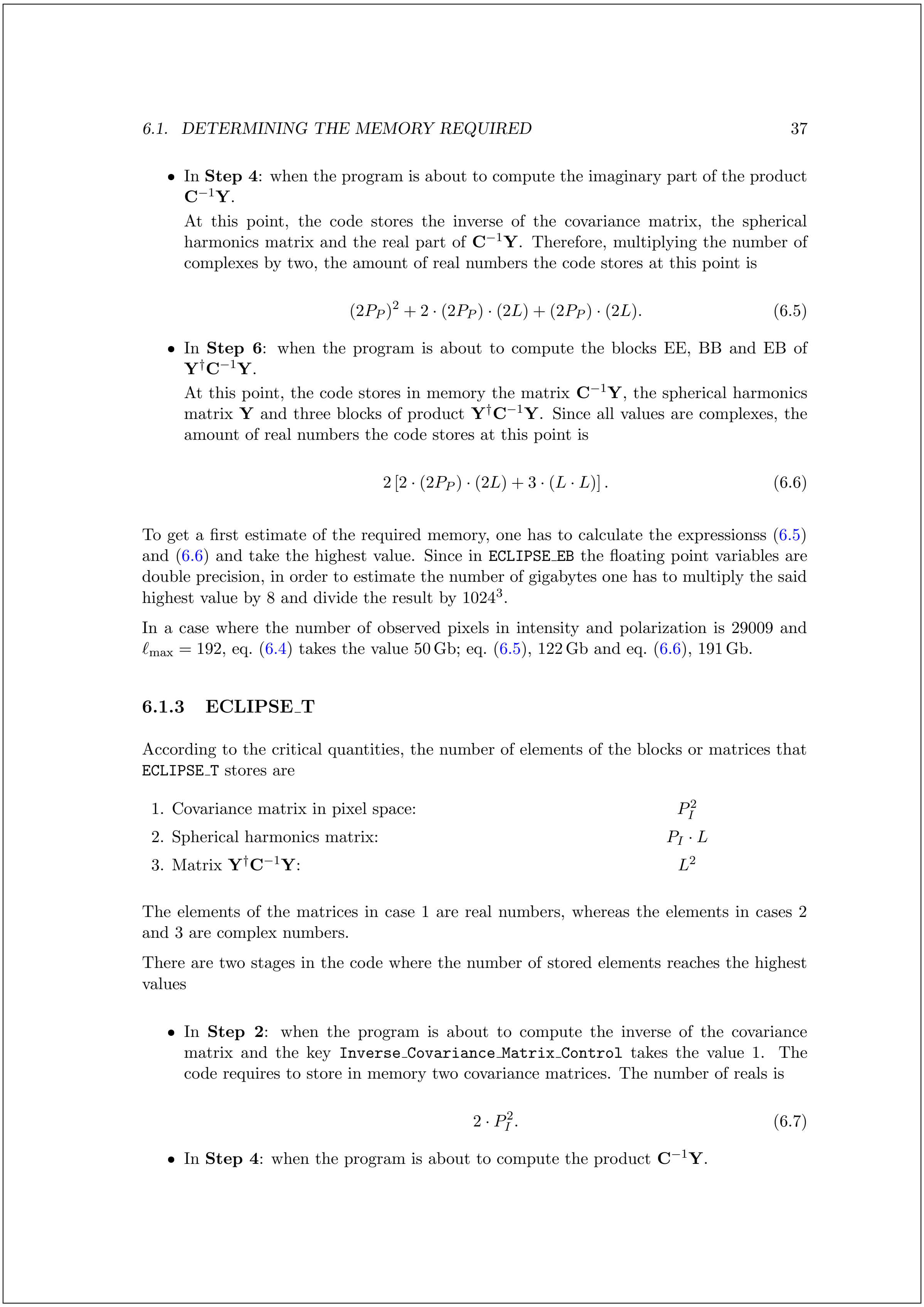
1. Covariance matrix in pixel space:	$(2P_P)^2$
2. Spherical harmonics matrix:	$(2P_P) \cdot (2L)$
3. Matrix $\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y}$:	$(2L)^2$
4. One of the four blocks of the matrix $\mathbf{Y}^{\dagger}\mathbf{C}^{-1}\mathbf{Y}$:	$L \cdot L$

The elements of the matrices in case 1 are real numbers, whereas the elements in cases 2, 3 and 4 are complex numbers.

There are two stages in the code where the number of stored elements reaches the highest values

- In **Step 2**: when the program is about to compute the inverse of the covariance matrix and the key `Inverse_Covariance_Matrix_Control` takes the value 1. The code requires to store in memory two covariance matrices. The number of reals is

$$2 \cdot (2P_P)^2. \tag{6.4}$$



- In **Step 4**: when the program is about to compute the imaginary part of the product $\mathbf{C}^{-1}\mathbf{Y}$.

  At this point, the code stores the inverse of the covariance matrix, the spherical harmonics matrix and the real part of $\mathbf{C}^{-1}\mathbf{Y}$. Therefore, multiplying the number of complexes by two, the amount of real numbers the code stores at this point is

  $$(2P_P)^2 + 2 \cdot (2P_P) \cdot (2L) + (2P_P) \cdot (2L). \tag{6.5}$$

- In **Step 6**: when the program is about to compute the blocks EE, BB and EB of $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}$.

  At this point, the code stores in memory the matrix $\mathbf{C}^{-1}\mathbf{Y}$, the spherical harmonics matrix $\mathbf{Y}$ and three blocks of product $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}$. Since all values are complexes, the amount of real numbers the code stores at this point is

  $$2\left[2 \cdot (2P_P) \cdot (2L) + 3 \cdot (L \cdot L)\right]. \tag{6.6}$$

To get a first estimate of the required memory, one has to calculate the expressionss (6.5) and (6.6) and take the highest value. Since in `ECLIPSE_EB` the floating point variables are double precision, in order to estimate the number of gigabytes one has to multiply the said highest value by 8 and divide the result by 1024^3.

In a case where the number of observed pixels in intensity and polarization is 29009 and $\ell_{\max} = 192$, eq. (6.4) takes the value 50 Gb; eq. (6.5), 122 Gb and eq. (6.6), 191 Gb.

6.1.3 ECLIPSE_T

According to the critical quantities, the number of elements of the blocks or matrices that `ECLIPSE_T` stores are

1. Covariance matrix in pixel space: P_I^2
2. Spherical harmonics matrix: $P_I \cdot L$
3. Matrix $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}$: L^2

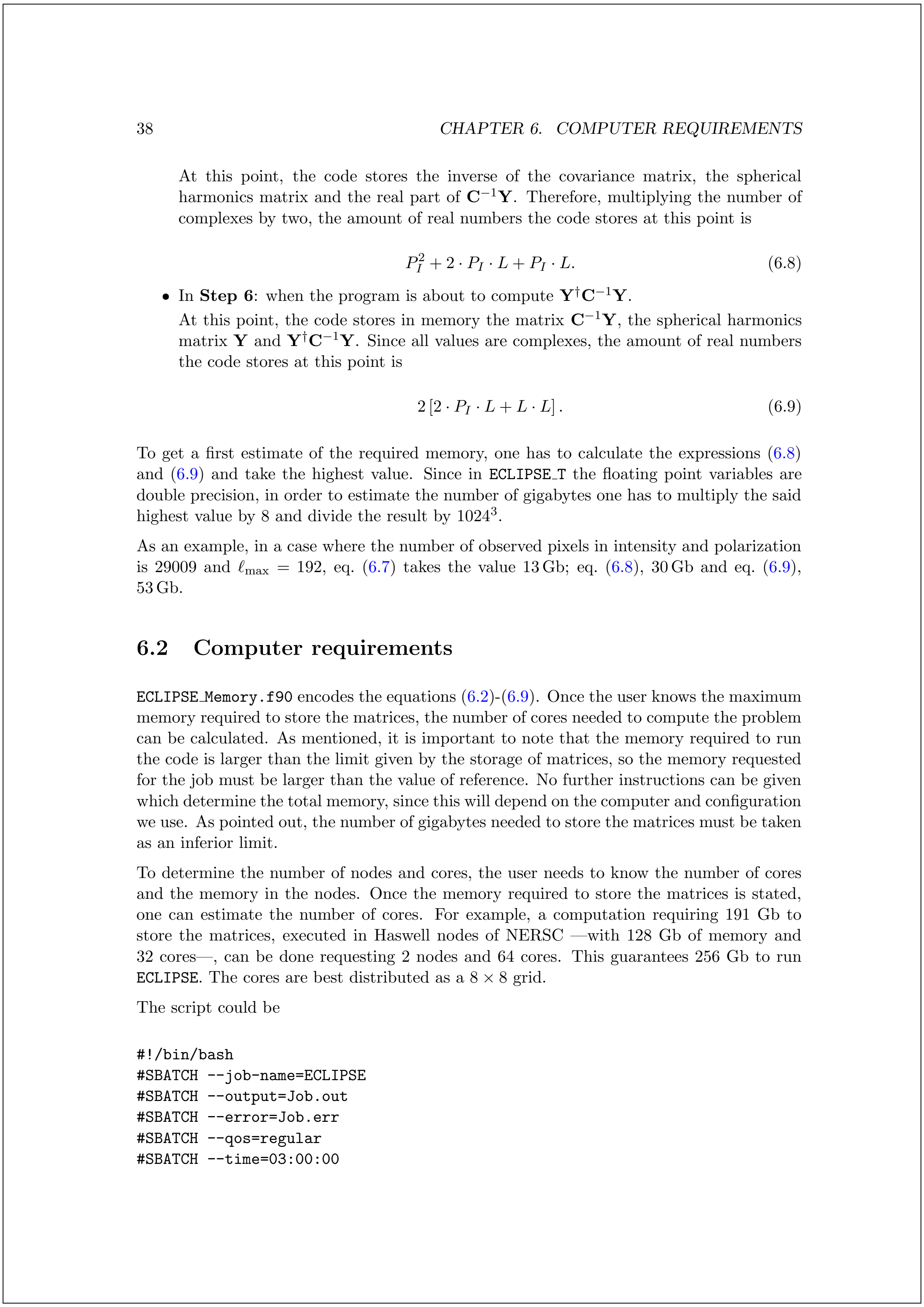
The elements of the matrices in case 1 are real numbers, whereas the elements in cases 2 and 3 are complex numbers.

There are two stages in the code where the number of stored elements reaches the highest values

- In **Step 2**: when the program is about to compute the inverse of the covariance matrix and the key `Inverse_Covariance_Matrix_Control` takes the value 1. The code requires to store in memory two covariance matrices. The number of reals is

  $$2 \cdot P_I^2. \tag{6.7}$$

- In **Step 4**: when the program is about to compute the product $\mathbf{C}^{-1}\mathbf{Y}$.



At this point, the code stores the inverse of the covariance matrix, the spherical harmonics matrix and the real part of $\mathbf{C}^{-1}\mathbf{Y}$. Therefore, multiplying the number of complexes by two, the amount of real numbers the code stores at this point is

$$P_I^2 + 2 \cdot P_I \cdot L + P_I \cdot L. \tag{6.8}$$

- In **Step 6**: when the program is about to compute $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}$.
  At this point, the code stores in memory the matrix $\mathbf{C}^{-1}\mathbf{Y}$, the spherical harmonics matrix $\mathbf{Y}$ and $\mathbf{Y}^\dagger\mathbf{C}^{-1}\mathbf{Y}$. Since all values are complexes, the amount of real numbers the code stores at this point is

$$2\left[2 \cdot P_I \cdot L + L \cdot L\right]. \tag{6.9}$$

To get a first estimate of the required memory, one has to calculate the expressions (6.8) and (6.9) and take the highest value. Since in `ECLIPSE_T` the floating point variables are double precision, in order to estimate the number of gigabytes one has to multiply the said highest value by 8 and divide the result by 1024^3.

As an example, in a case where the number of observed pixels in intensity and polarization is 29009 and $\ell_{\max} = 192$, eq. (6.7) takes the value 13 Gb; eq. (6.8), 30 Gb and eq. (6.9), 53 Gb.

6.2 Computer requirements

`ECLIPSE_Memory.f90` encodes the equations (6.2)-(6.9). Once the user knows the maximum memory required to store the matrices, the number of cores needed to compute the problem can be calculated. As mentioned, it is important to note that the memory required to run the code is larger than the limit given by the storage of matrices, so the memory requested for the job must be larger than the value of reference. No further instructions can be given which determine the total memory, since this will depend on the computer and configuration we use. As pointed out, the number of gigabytes needed to store the matrices must be taken as an inferior limit.

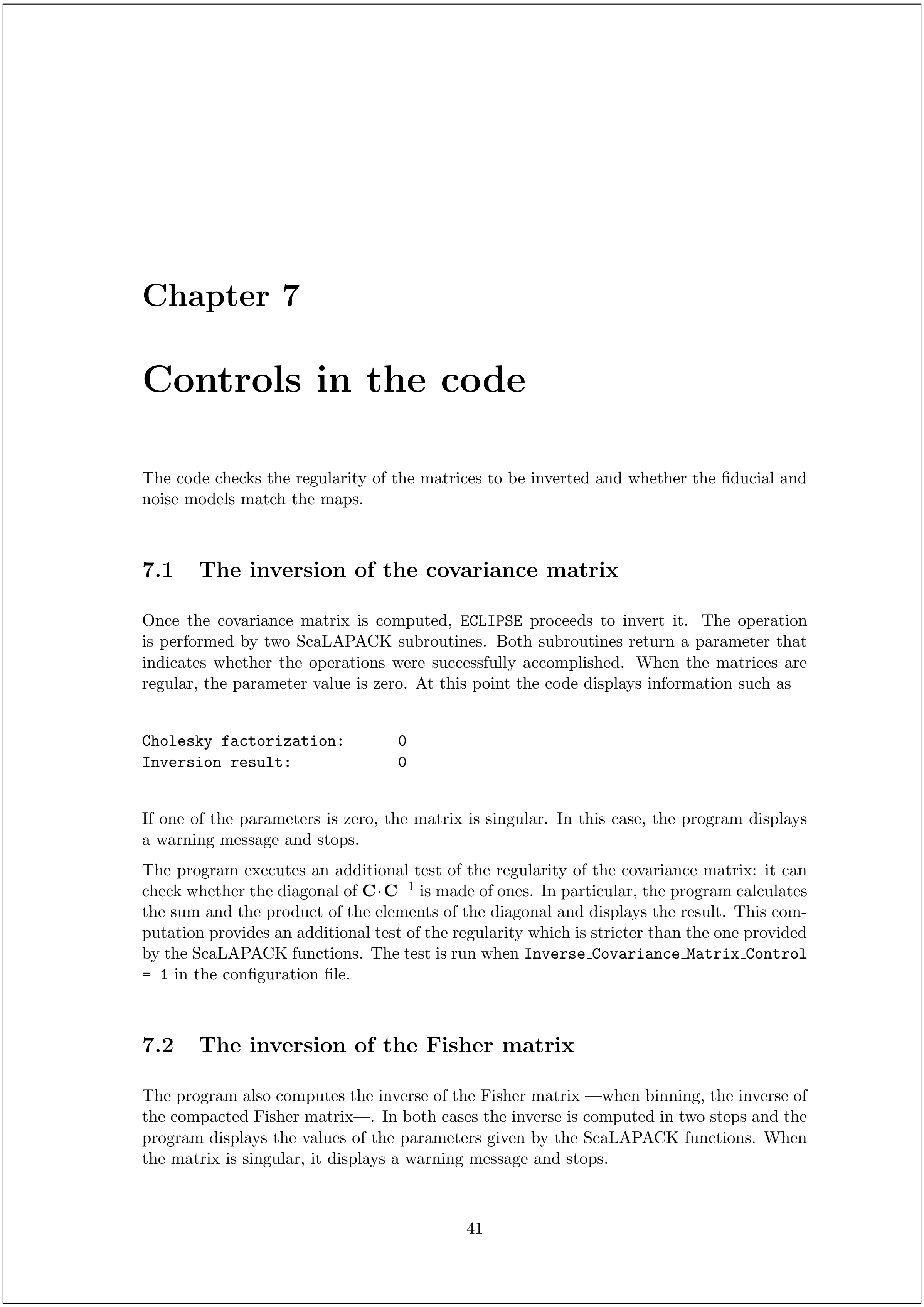
To determine the number of nodes and cores, the user needs to know the number of cores and the memory in the nodes. Once the memory required to store the matrices is stated, one can estimate the number of cores. For example, a computation requiring 191 Gb to store the matrices, executed in Haswell nodes of NERSC —with 128 Gb of memory and 32 cores—, can be done requesting 2 nodes and 64 cores. This guarantees 256 Gb to run `ECLIPSE`. The cores are best distributed as a 8 × 8 grid.

The script could be

```
#!/bin/bash
#SBATCH --job-name=ECLIPSE
#SBATCH --output=Job.out
#SBATCH --error=Job.err
#SBATCH --qos=regular
#SBATCH --time=03:00:00
```

```
#SBATCH --nodes=2
#SBATCH --tasks-per-node=32
#SBATCH --constraint=haswell
#SBATCH --mail-user=<user>@<domain>.com
#SBATCH --mail-type=ALL

srun ./ECLIPSE_TEB 8 NSide64/Config.ini
```

40 CHAPTER 6. COMPUTER REQUIREMENTS



Chapter 7

Controls in the code

The code checks the regularity of the matrices to be inverted and whether the fiducial and noise models match the maps.

7.1 The inversion of the covariance matrix

Once the covariance matrix is computed, `ECLIPSE` proceeds to invert it. The operation is performed by two ScaLAPACK subroutines. Both subroutines return a parameter that indicates whether the operations were successfully accomplished. When the matrices are regular, the parameter value is zero. At this point the code displays information such as

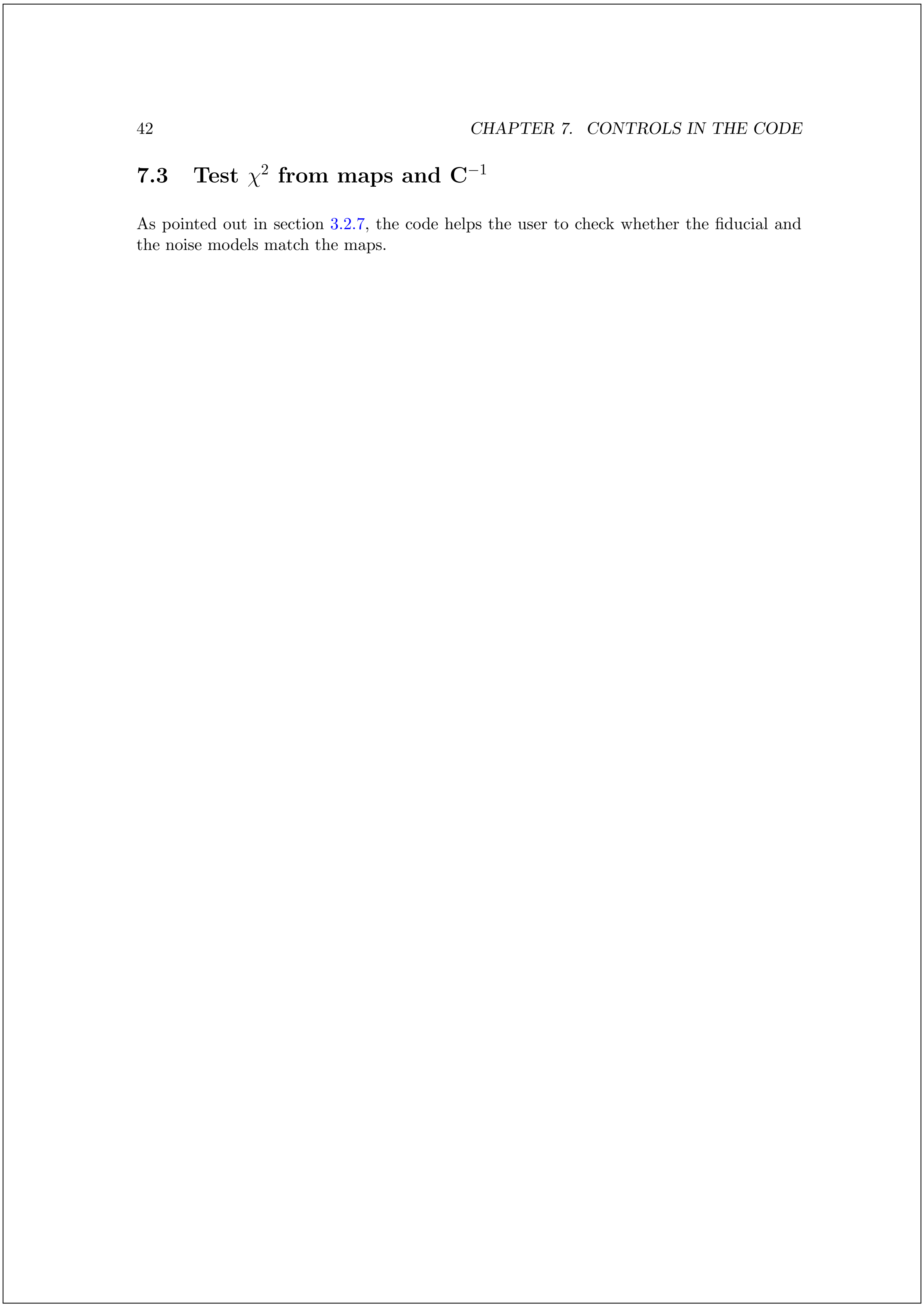
```
Cholesky factorization:      0
Inversion result:            0
```

If one of the parameters is zero, the matrix is singular. In this case, the program displays a warning message and stops.

The program executes an additional test of the regularity of the covariance matrix: it can check whether the diagonal of $\mathbf{C}\cdot\mathbf{C}^{-1}$ is made of ones. In particular, the program calculates the sum and the product of the elements of the diagonal and displays the result. This computation provides an additional test of the regularity which is stricter than the one provided by the ScaLAPACK functions. The test is run when `Inverse_Covariance_Matrix_Control = 1` in the configuration file.

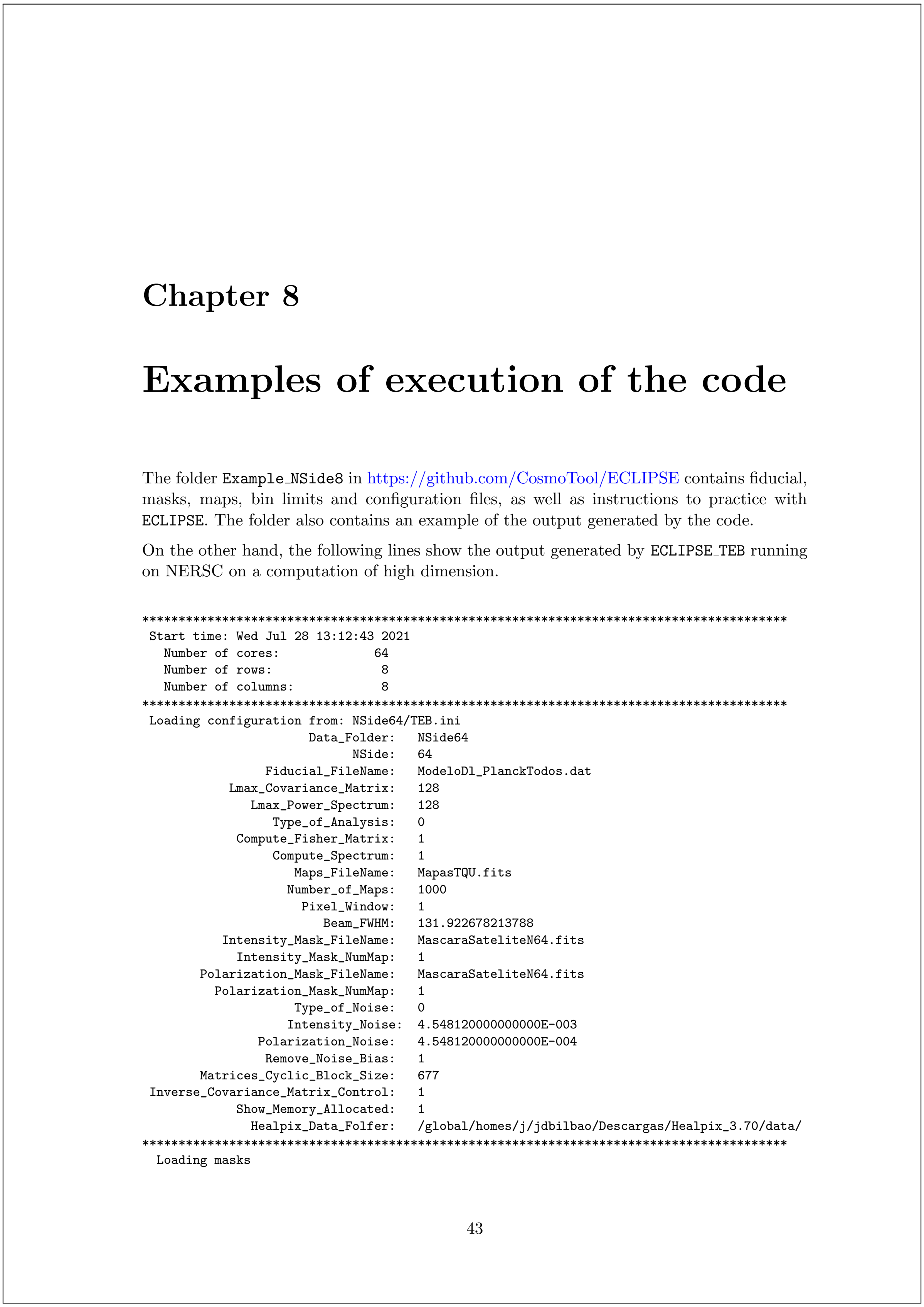
7.2 The inversion of the Fisher matrix

The program also computes the inverse of the Fisher matrix —when binning, the inverse of the compacted Fisher matrix—. In both cases the inverse is computed in two steps and the program displays the values of the parameters given by the ScaLAPACK functions. When the matrix is singular, it displays a warning message and stops.



7.3 Test χ^2 from maps and $\mathbf{C}^{-1}$

As pointed out in section 3.2.7, the code helps the user to check whether the fiducial and the noise models match the maps.



Chapter 8

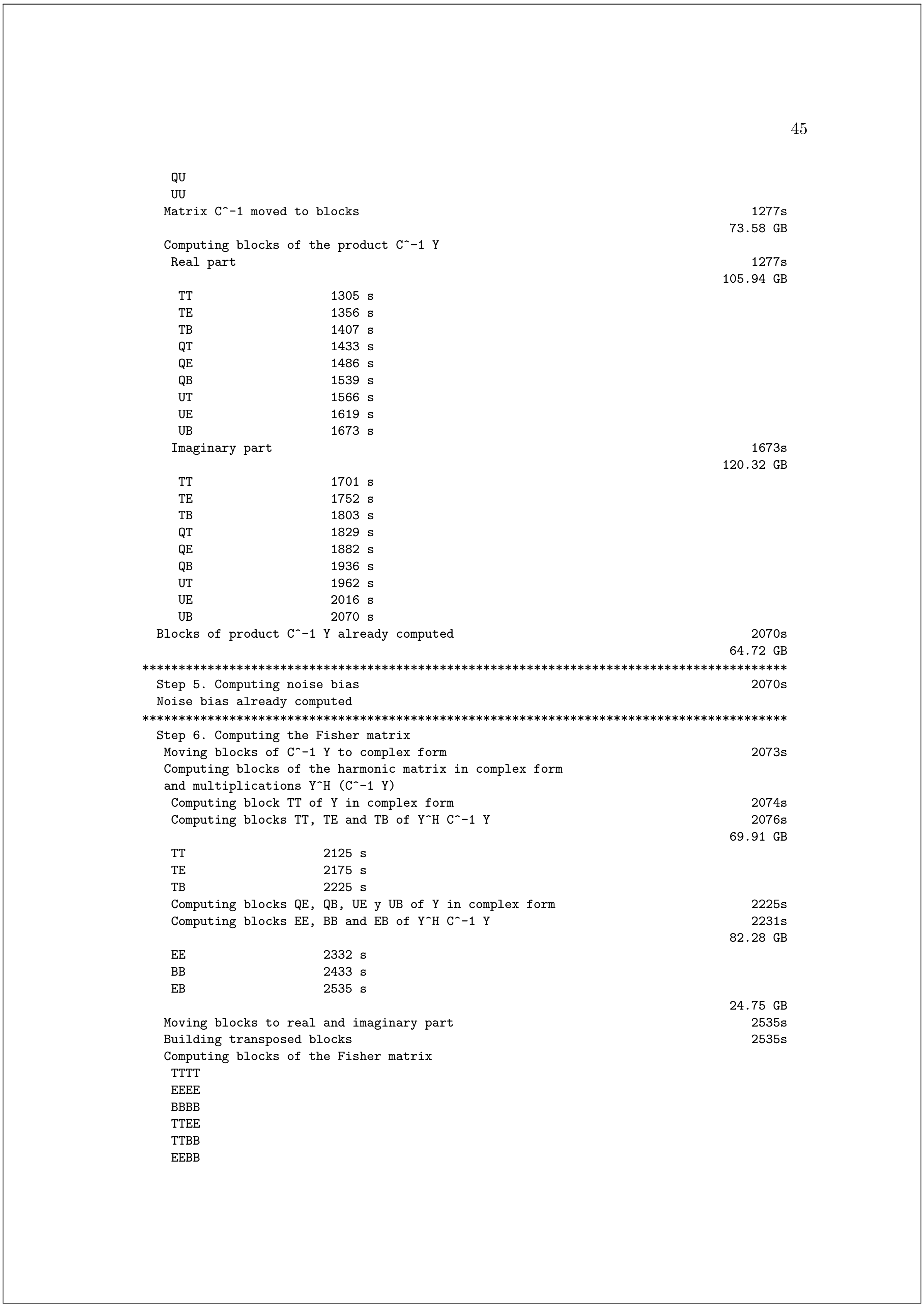
Examples of execution of the code

The folder `Example_NSide8` in https://github.com/CosmoTool/ECLIPSE contains fiducial, masks, maps, bin limits and configuration files, as well as instructions to practice with `ECLIPSE`. The folder also contains an example of the output generated by the code.

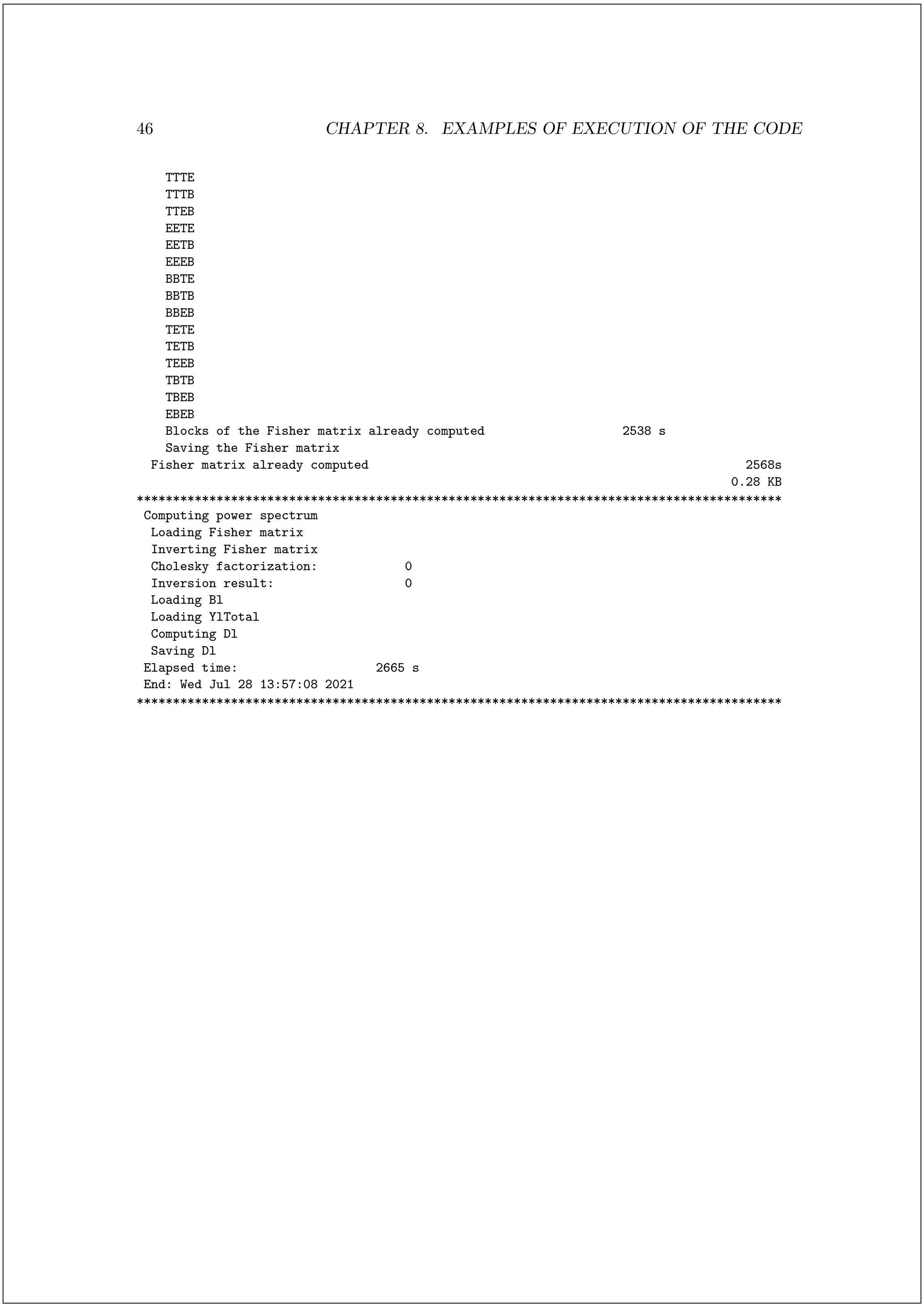
On the other hand, the following lines show the output generated by `ECLIPSE_TEB` running on NERSC on a computation of high dimension.

```
*****************************************************************************************
 Start time: Wed Jul 28 13:12:43 2021
   Number of cores:             64
   Number of rows:               8
   Number of columns:            8
*****************************************************************************************
 Loading configuration from: NSide64/TEB.ini
                       Data_Folder:   NSide64
                             NSide:   64
                 Fiducial_FileName:   ModeloDl_PlanckTodos.dat
            Lmax_Covariance_Matrix:   128
               Lmax_Power_Spectrum:   128
                  Type_of_Analysis:   0
             Compute_Fisher_Matrix:   1
                  Compute_Spectrum:   1
                     Maps_FileName:   MapasTQU.fits
                    Number_of_Maps:   1000
                      Pixel_Window:   1
                         Beam_FWHM:   131.922678213788
           Intensity_Mask_FileName:   MascaraSateliteN64.fits
             Intensity_Mask_NumMap:   1
        Polarization_Mask_FileName:   MascaraSateliteN64.fits
          Polarization_Mask_NumMap:   1
                     Type_of_Noise:   0
                    Intensity_Noise:  4.548120000000000E-003
                Polarization_Noise:   4.548120000000000E-004
                 Remove_Noise_Bias:   1
        Matrices_Cyclic_Block_Size:   677
 Inverse_Covariance_Matrix_Control:   1
             Show_Memory_Allocated:   1
               Healpix_Data_Folfer:   /global/homes/j/jdbilbao/Descargas/Healpix_3.70/data/
*****************************************************************************************
  Loading masks
```

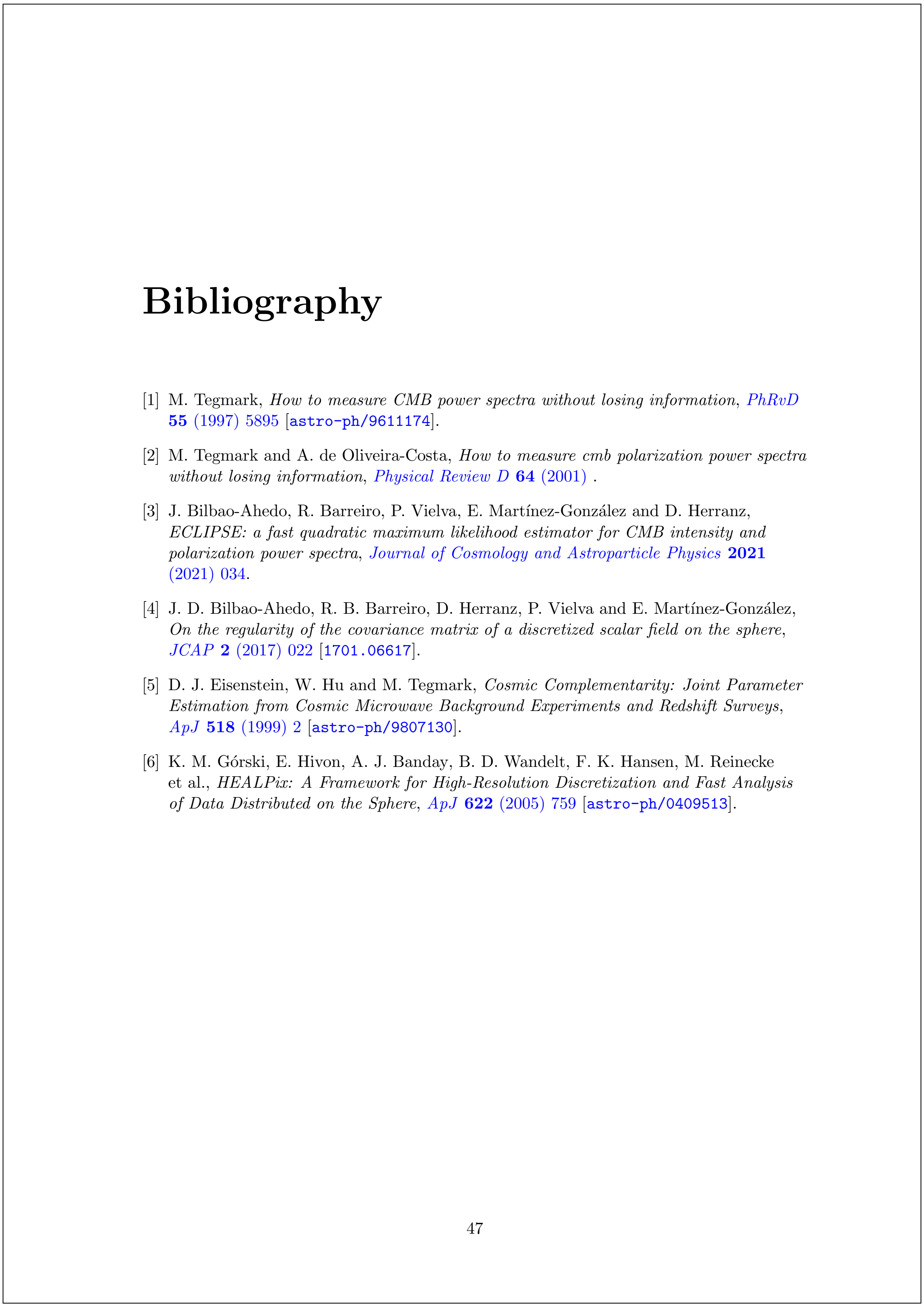
```
   Number of observed pixels in temperature:                                       29009
   Number of observed pixels in polarization:                                      29009
  Loading HEALPix Pixel Window
  Loading Fiducial Power Spectrum
****************************************************************************************
  Step 1. Computing covariance matrix
    Computing blocks of the spherical harmonics matrix                                0s
                                                                                35.96 GB
     Block YTT done                                                                   2s
     Block YPP done                                                                   7s
    Computing blocks of the covariance matrix                                         8s
                                                                                92.39 GB
     Block TT done                    57 s
     Block TQ done                   107 s
     Block QT done                   109 s
     Block TU done                   159 s
     Block UT done                   160 s
     Block QQ done                   259 s
     Block QU done                   358 s
     Block UQ done                   360 s
     Block UU done                   459 s
   Intensity diagonal element:      3073.14818800938
   Polarization diagonal element:   5.834018299670649E-002
  Covariance matrix already computed
                                                                                56.43 GB
****************************************************************************************
  Step 2. Inverting the covariance matrix                                           459s
                                                                               112.86 GB
    Cholesky factorization:      0                                                  728s
    Inversion result:            0                                                 1164s
                                                                                56.43 GB
    Diagonal product:   0.999692798028858
    Diagonal sum:        87026.9999865500
    Matrix size:             87027
  Covariance matrix inverted                                                       1167s
                                                                                56.43 GB
****************************************************************************************
  Step 3. Computing coupled power in the harmonics space
    Computing blocks of the spherical harmonics matrix
                                                                                92.39 GB
     Block YTT done                                                                1169s
     Block YPP done                                                                1175s
    Loading maps
                                                                                93.04 GB
    Computing C^-1 Maps                                                            1211s
     <m^t C^-1 m>:    87042.2726098239
     Matrix size:         87027
                                                                                93.79 GB
    Maps transformed to harmonis space
  Auto-corrrelation power already computed                                         1273s
                                                                                92.39 GB
****************************************************************************************
  Step 4. Computing product C^-1 Y
   Moving from matrix C^-1 to blocks of the matrix C^-1                            1273s
                                                                               130.01 GB
    TT
    TQ
    TU
    QQ
```



```
    QU
    UU
   Matrix C^-1 moved to blocks                                                      1277s
                                                                                 73.58 GB
   Computing blocks of the product C^-1 Y
    Real part                                                                       1277s
                                                                                105.94 GB
     TT                 1305 s
     TE                 1356 s
     TB                 1407 s
     QT                 1433 s
     QE                 1486 s
     QB                 1539 s
     UT                 1566 s
     UE                 1619 s
     UB                 1673 s
    Imaginary part                                                                  1673s
                                                                                120.32 GB
     TT                 1701 s
     TE                 1752 s
     TB                 1803 s
     QT                 1829 s
     QE                 1882 s
     QB                 1936 s
     UT                 1962 s
     UE                 2016 s
     UB                 2070 s
  Blocks of product C^-1 Y already computed                                         2070s
                                                                                 64.72 GB
*****************************************************************************************
  Step 5. Computing noise bias                                                      2070s
  Noise bias already computed
*****************************************************************************************
  Step 6. Computing the Fisher matrix
   Moving blocks of C^-1 Y to complex form                                          2073s
   Computing blocks of the harmonic matrix in complex form
   and multiplications Y^H (C^-1 Y)
    Computing block TT of Y in complex form                                         2074s
    Computing blocks TT, TE and TB of Y^H C^-1 Y                                    2076s
                                                                                 69.91 GB
    TT                 2125 s
    TE                 2175 s
    TB                 2225 s
    Computing blocks QE, QB, UE y UB of Y in complex form                           2225s
    Computing blocks EE, BB and EB of Y^H C^-1 Y                                    2231s
                                                                                 82.28 GB
    EE                 2332 s
    BB                 2433 s
    EB                 2535 s

                                                                                 24.75 GB
   Moving blocks to real and imaginary part                                         2535s
   Building transposed blocks                                                       2535s
   Computing blocks of the Fisher matrix
    TTTT
    EEEE
    BBBB
    TTEE
    TTBB
    EEBB
```



```
    TTTE
    TTTB
    TTEB
    EETE
    EETB
    EEEB
    BBTE
    BBTB
    BBEB
    TETE
    TETB
    TEEB
    TBTB
    TBEB
    EBEB
    Blocks of the Fisher matrix already computed                   2538 s
    Saving the Fisher matrix
  Fisher matrix already computed                                                    2568s
                                                                                  0.28 KB
*****************************************************************************************
 Computing power spectrum
  Loading Fisher matrix
  Inverting Fisher matrix
  Cholesky factorization:            0
  Inversion result:                  0
  Loading Bl
  Loading YlTotal
  Computing Dl
  Saving Dl
 Elapsed time:                   2665 s
 End: Wed Jul 28 13:57:08 2021
*****************************************************************************************
```



Bibliography

[1] M. Tegmark, *How to measure CMB power spectra without losing information*, PhRvD **55** (1997) 5895 [astro-ph/9611174].

[2] M. Tegmark and A. de Oliveira-Costa, *How to measure cmb polarization power spectra without losing information*, *Physical Review D* **64** (2001) .

[3] J. Bilbao-Ahedo, R. Barreiro, P. Vielva, E. Martínez-González and D. Herranz, *ECLIPSE: a fast quadratic maximum likelihood estimator for CMB intensity and polarization power spectra*, *Journal of Cosmology and Astroparticle Physics* **2021** (2021) 034.

[4] J. D. Bilbao-Ahedo, R. B. Barreiro, D. Herranz, P. Vielva and E. Martínez-González, *On the regularity of the covariance matrix of a discretized scalar field on the sphere*, *JCAP* **2** (2017) 022 [1701.06617].

[5] D. J. Eisenstein, W. Hu and M. Tegmark, *Cosmic Complementarity: Joint Parameter Estimation from Cosmic Microwave Background Experiments and Redshift Surveys*, *ApJ* **518** (1999) 2 [astro-ph/9807130].

[6] K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke et al., *HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere*, *ApJ* **622** (2005) 759 [astro-ph/0409513].

RESUMEN EN CASTELLANO

El fondo cósmico de microondas (FCM) incide sobre nosotros desde todas las direcciones del cielo. Al observarlo, encontramos una radiación prácticamente isótropa. Pero debido a que nos movemos respecto al marco de referencia del FCM, las frecuencias de la señal están afectadas por el corrimiento Doppler. Tras operar sobre los datos para revertir el efecto y determinar el FCM tal como se sería visto por un observador en reposo, se encuentra que presenta ligeras desviaciones locales, del orden de 1 en 100 000. Esas pequeñas diferencias codifican información muy valiosa sobre las características del universo en la época en que la radiación quedó liberada del equilibrio termodinámico con la materia (desacoplamiento) y comenzó a viajar libremente por el espacio.

Las anisotropías que hoy medimos en el FCM nos hablan de las inhomogeneidades en el universo en la época del desacoplamiento. El Principio Cosmológico establece que todo el universo sigue las mismas leyes de la Física y que ningún lugar en él es un lugar especial. Además, como resultado de experimentos y observaciones realizadas en el lugar y tiempo que ocupamos en el universo, contamos con un conocimiento extenso y profundo del comportamiento de la radiación, la materia, los fluidos en equilibrio y la evolución del espacio-tiempo y de las leyes de la Física en general. Combinando el Principio Cosmológico con el conocimiento adquirido localmente, podemos dar el salto a confiar en que tenemos conocimiento sobre las leyes que rigen el conjunto del universo.

Por lo tanto, a partir del patrón de inhomogeneidades que nos revela el FCM, podemos inferir las características fundamentales de nuestro universo en los instantes iniciales.

Para conectar las anisotropías que medimos con las leyes que determinan las características de nuestro universo, tenemos que transformar matemáticamente la información codificada en ellas. Podemos descomponer el mapa de anisotropías como suma de capas superpuestas de oscilaciones de intensidad de radiación en diferentes escalas angulares. Por otro lado, los valores locales de las anisotropías tienen un origen aleatorio y no nos dicen nada relativo a las características particulares de cada dirección del cielo. Lo que aporta realmente información valiosa es la potencia de las anisotropías en las diferentes escalas angulares, el espectro angular de potencias de la radiación.

El espectro angular de potencias encapsula la información esencial codificada en las anisotropías del FCM, que conecta con las características básicas de nuestro universo. Pero como la realización del FCM que medimos es, de acuerdo a nuestra teoría, una de las infinitas posibles de la ley de probabilidad dada por las características básicas y esenciales de nuestro universo, hay cierta parte de información superflua o arbitraria en el espectro angular de potencias. En definitiva, la información se puede comprimir en una colección reducida de parámetros que sí encierran las características esenciales: los parámetros cosmológicos. Uno de los objetivos clave en Cosmología es determinar las leyes que definen las características de nuestro universo y los valores de los parámetros libres de esas leyes.

El espectro angular de potencias caracteriza las propiedades estadísticas del FCM. Los modelos de inflación favorecidos actualmente predicen que las fluctuaciones son gaussianas (o muy aproximadamente gaussianas); en ese caso, el espectro de potencia las determina completamente. A su vez, los valores del espectro se pueden deducir completamente a partir de los parámetros cosmológicos. Lo anterior conduce a que el espectro angular de potencias conserva la información registrada en el FCM acerca de los parámetros cosmológicos. Y, sin llegar a contenerla toda si las fluctuaciones no son gaussianas, atesora una información muy valiosa, en todo caso. Además, en el camino desde el instante del desacoplamiento hasta nuestros detectores, el FCM sufre otros procesos físicos que dejan un rastro en él, desde el que se pueden inferir propiedades de la evolución y contenido del universo.

Las ondas electromagnéticas del FCM transportan información acerca de la intensidad y polarización, que codificamos mediante los parámetros de Stokes I, Q y U. La intensidad está relacionada con la temperatura de la radiación del cuerpo negro que conformaba el fluido en equilibrio formado por radiación y materia. La polarización que encontramos en cada dirección del cielo conecta con la distribución de las inhomogeneidades en la época del desacoplamiento. La intensidad (o temperatura) es un escalar, mientras que los parámetros Q y U son componentes cuyo valor depende de la orientación de los ejes de coordenadas locales. Se puede encontrar la forma de transformar al espacio armónico esa información en forma de los valores de los coeficientes del desarrollo en armónicos esféricos en intensidad (o temperatura) y de los coeficientes de las dos componentes de polarización, E y B. Llevado a espectro angular de potencias, encontramos el espectro angular de potencias de temperatura (TT), las tres componentes de polarización (EE, BB y EB) y las dos componentes del cruce de temperatura y polarización (TE y TB).

Las anisotropías en el FCM fueron causadas por inhomogeneidades locales en el fluido primitivo. Por otro lado, dado el tiempo transcurrido desde el Big Bang, lo limitado de la velocidad a la que se puede propagar la interacción y el tamaño del universo observable, a priori, no podemos esperar que zonas suficientemente alejadas tengan las mismas propiedades. Sin embargo, observamos que el FCM es sorprendentemente isótropo a grandes escalas. Para dar respuesta, entre otros, a este problema, se han desarrollado múltiples modelos teóricos. Los más aceptados por la comunidad científica son los modelos inflacionarios, que explican la isotropía a gran escala y el origen de las inhomogeneidades que dieron lugar a las anisotropías en el FCM. Combinando la relatividad general, el principio cosmológico y la mecánica cuántica, describen posibles estados iniciales que, tras un periodo muy corto de tiempo de crecimiento exponencial, finalizan en estados que con el paso del tiempo evolucionan hacia el universo que pudo dar lugar al FCM y a las estructuras a gran escala que observamos en el presente a diferentes corrimientos al rojo. Además, por sus propias características, el modo B de polarización y la componente BB del espectro angular de potencias juegan un papel esencial en los modelos inflacionarios; lo que los convierte en un test muy potente para estos modelos.

Aparte del FCM, hay otras fuentes de emisión de radiación de microondas. Lo que medimos es la suma de todas ellas. No podemos determinar con precisión las características de la señal del FCM en las direcciones del cielo en las que esas otras fuentes son muy intensas —como, por ejemplo, las que cubre nuestra galaxia—, por lo tanto, no contamos con el mapa completo del FCM. Lo anterior introduce una seria complicación matemática que nos impide calcular directamente las características de la descomposición en suma en escalas angulares. Por lo tanto, como no podemos calcularlo directamente, tenemos que recurrir a métodos de estimación del espectro angular de potencias.

Los métodos de estimación son herramientas matemáticas que nos permiten inferir valores de parámetros a partir de colecciones de datos que siguen ciertas leyes de probabilidad. En el caso del FCM, podemos estimar los valores del espectro angular de potencias: los datos los conforma el mapa parcial de anisotropías, la ley de probabilidad encapsula las leyes de la Física.

Como herramientas matemáticas, los métodos de estimación tienen ciertas propiedades. Esperamos que, en promedio, un estimador proporcione valores correctos de los parámetros cuando le suministremos datos que siguen la ley de probabilidad que le indicamos que siguen esos datos. Si es así, decimos que el estimador es insesgado. Por otro lado, por tratarse de una cuestión estadística-probabilística, las estimaciones vienen acompañadas de un error. Cuando utilizamos una distribución de probabilidad para producir datos que la reflejen, estamos expuestos a fluctuaciones, de modo que la muestra será tanto más representativa cuanto mayor sea. Y en caso de ser demasiado pequeña, es probable que no resulte representativa. En definitiva, los propios datos arrastran incertidumbre sobre las características de la distribución de probabilidad. La matriz de Fisher conduce directamente a las cotas mínimas en las incertidumbres en los parámetros dados por un estimador insesgado como consecuencia de la naturaleza estadística de los datos. Decimos que un estimador es óptimo cuando es insesgado y de varianza mínima.

En esencia, esta tesis trata sobre la obtención de la estimación óptima del espectro angular de potencias del fondo cósmico de microondas, lo que incluye la estimación de dichos parámetros y su correspondiente matriz de covarianza. Para ello, repasamos las propiedades generales y las limitaciones de los estimadores, los aspectos esenciales del estimador de máxima verosimilitud y del estimador pseudo-espectro y nos centramos en el estimador cuadrático de máxima verosimilitud (QML).

El estimador de máxima verosimilitud (ML) es óptimo: insesgado y de varianza mínima. Se basa en localizar los valores de los parámetros que maximizan la probabilidad de los datos, por lo que implica la búsqueda de los mejores valores en el espacio de parámetros, una tarea que implica una gran carga computacional al aplicarlo a la estimación del espectro angular de potencias del FCM, si los mapas son grandes.

El estimador cuadrático de máxima verosimilitud también es óptimo, permite encontrar valores del espectro angular de potencias con las mismas propiedades que los del ML, pero haciéndolo de una forma mucho más directa. Además, garantiza que ya en la primera tentativa se encuentran valores insesgados, aunque no de varianza óptima. En esta tesis, siguiendo el artículo original en el que fue presentado [43] , mostramos una deducción matemática muy detallada del estimador. Aparte de los propios datos, tanto QML como ML —y cualquier estimado en general— requieren, para aplicarlo, que aportemos cierta información. En la deducción de QML se supone que las fluctuaciones del FCM son gaussianas. Además, al aplicarlo, se supone que tenemos conocimiento de las características del ruido en los mapas. Por otro lado, necesitamos aportar un modelo de espectro angular de potencias con el que alimentar el método para ponerlo en marcha. En la tesis, demostramos y comprobamos que, suponiendo que el modelo de ruido sea correcto, el método es insesgado independientemente del modelo de señal aportado y que, iterando, las estimaciones convergen a los mismos valores independientemente del punto de partida.

Matemáticamente, la superficie de la esfera es un continuo, y cualquier señal en ella también lo es. Técnicamente es imposible captar datos en continuo y, aunque pudiéramos hacerlo, también técnicamente sería imposible procesarlos. Lo que se hace es integrar la señal captada en teselas predefinidas de la esfera y convertirla en un mapa discreto. Técnicamente, pixelamos la señal. De este modo, podemos representar los datos en el espacio armónico sin pérdida de información truncando el desarrollo, sin sumar infinitos términos. A la forma de teselar la superficie le llamamos pixelación.

Si suponemos que la señal y el ruido son independientes, la matriz de covarianza de los datos (los mapas) resulta ser la suma de las matrices de covarianza de la señal y del ruido. El método QML requiere la inversión de la matriz de covarianza de los mapas. Un estudio de las propiedades de la matriz de covarianza de la señal nos ha permitido encontrar una condición necesaria para que pueda ser regular: el número de armónicos esféricos requeridos para representarla tiene que ser igual o mayor al número de píxeles en los mapas. Por otro lado, hemos encontrado que, debido a las propiedades de los armónicos esféricos y a las simetrías en los puntos que utilizamos para describir la posición de los píxeles, el rango de la matriz de covarianza puede resultar ser menor que el que se espera a cuenta del número de armónicos que se utilizan para describirla. Además, independientemente de que sea isótropo o anisótropo, desde el punto de vista formal matemático, el ruido no correlacionado espacialmente regulariza la matriz. Aunque en las pruebas numéricas que hemos realizado se ha encontrado que debe alcanzar una potencia mínima; en caso contrario, los errores numéricos barren el efecto regularizador. En esta tesis hemos realizado estudios analíticos que demuestran todo lo anterior en lo referido a la regularidad de la matriz de covarianza y estudios numéricos detallados que lo ponen a prueba —algunos calculados con diferentes precisiones numéricas—. Lo hemos hecho en casos a cielo completo y con cobertura parcial, con y sin ruido, usando señales de FCM simuladas simplificadas y señales realistas, y utilizando varias pixelizaciones.

Por otro lado, el método requiere la inversión de la matriz de Fisher, pero en algunos casos la matriz es singular, y no se puede calcular el espectro angular de potencias multipolo a multipolo utilizando QML.

Si no se puede calcular la potencia multipolo a multipolo, se puede recurrir a calcular los promedios en grupos de multipolos; es decir, podemos realizar una estimación bineada.[1] Por otro lado, según los modelos, la potencia de la señal sigue una curva suave, pero en la práctica cada realización particular consiste en un conjunto de valores que fluctúan en torno a la curva central dentro de los límites dados por las barras de error; es decir, la potencia de un caso concreto forma una figura un tanto errática, que es conveniente suavizar mediante promedio de grupos de datos vecinos. Por lo tanto, sea porque la matriz de Fisher es singular o sea porque se prefiere una potencia suavizada, en ocasiones es necesario binear. En esta tesis hemos desarrollado una técnica que permite encontrar la estimación óptima del espectro angular de potencias bineado Es decir, hemos desarrollado lo que se puede entender como una extensión de QML al caso bineado. La técnica permite encontrar el valor insesgado de la potencia en los bines y la matriz de Fisher de la potencia en los bines, y demostramos que las covarianzas de la potencia en los bines igualan la que viene dada por la matriz de Fisher. Considerando que lo anterior es suficientemente bueno, hay dos aspectos que es necesario señalar. Por un lado, lo que tenemos es, dada una elección de bines, solo la mejor forma de calcular la potencia en los bines, pero no tenemos la mejor forma de determinar previamente cuáles son los mejores bines posibles. Por otro lado, formalmente, el método de bineado es insesgado siempre que se cumpla el requisito de que la información suministrada —parámetros de la ley de probabilidad de la señal y ruido— describan los mapas. Cuando el modelo de señal no refleja los mapas, se pueden producir ligeros sesgos (que se pueden compensar iterando).

Aunque QML simplifica mucho la localización de la solución óptima respecto a ML, sigue siendo un método con una alta carga computacional. Entre otras cosas, requiere el cálculo de la matriz de covarianza, de su inversa y el cálculo de la matriz de Fisher. La estructura matemática de la matriz de covarianza es mucho más simple en el espacio armónico que en el espacio de los píxeles. Haciendo uso de esta propiedad, el método se puede implementar de tal modo que buena parte de los cálculos con más carga computacional (como, por ejemplo, la determinación de la matriz de Fisher) se realicen a partir de matrices con una gran cantidad de ceros y unos pocos unos localizados en posiciones estratégicas (en vez de a partir de matrices densas). Solo este cambio reduce el número de operaciones y ya ha sido utilizado en publicaciones anteriores [53]. Lo que es novedad en esta tesis es que analizando en detalle el papel que juegan los unos y su localización en el contexto de las (abundantes) multiplicaciones de matrices, se puede predecir analíticamente el resultado de operaciones matriciales complejas. Es

---

[1] El verbo *binear* y las palabras derivadas no están reconocidas actualmente por la RAE. Debido a que se trata de términos suficientemente usados por la comunidad científica y con un significado claro, los utilizamos en esta Tesis.

decir, podemos saber cómo calcular de forma rápida y sencilla qué se va a obtener tras complejas y costosas operaciones computacionales, sin tener que realizarlas. Por lo tanto, podemos predecir los valores de los elementos de matrices y vectores esenciales en el método reduciendo de forma significativa el número de operaciones. En definitiva, hemos desarrollado una implementación formal del método que reduce el número de operaciones en varios ordenes de magnitud con respecto a implementaciones óptimas previas. Además, las operaciones sustitutas de los cálculos con matrices se pueden paralelizar. Pasando del papel a la práctica, hemos desarrollado una implementación en el lenguaje `Fortran` que hace uso de librerías optimizadas preexistentes para realizar cálculos con matrices en paralelo y que implementa las operaciones propias de nuestro método con algoritmos escritos para ser ejecutados en paralelo de forma óptima. El resultado es un código óptimo en paralelo para aplicar QML que puede hacer uso de la potencia de los supercomputadores y de la gran cantidad de memoria de que disponen para realizar cálculos grandes con relativa rapidez. El código `ECLIPSE` es una versión pública y de acceso libre de nuestra implementación, disponible en Internet. En la tesis, incorporamos el manual de uso, para ilustrar la facilidad de uso (por supuesto, asumiendo la dificultad que supone preparar toda la información necesaria para definir una situación en la que QML pueda ser aplicado) y versatilidad.

Una vez que tenemos un método matemático, QML, una formulación óptima y un código que la implementa que permite hacer múltiples pruebas con relativa rapidez y limitada carga computacional, podemos poner a prueba el método y analizar el efecto en los resultados de las circunstancias en las que se aplica. En las pruebas que hemos realizado, hemos encontrado que el método es insesgado y de varianza mínima cuando los modelos de señal y ruido que le proporcionamos se ajustan a las características de los mapas simulados, tanto cuando se calcula el espectro angular de potencias completo como cuando se aplica el bineado óptimo desarrollado en este trabajo.

Si solo estamos interesados en polarización, se puede aplicar QML para estimar solo las componentes oportunas, reduciendo el número de operaciones, con nuestra implementación, en aproximadamente un factor dos respecto del cálculo del espectro completo. Hemos comprobado que los resultados siguen siendo insesgados y que vienen acompañados de una incertidumbre muy similar a la que obtenemos cuando se calcula el espectro completo.

Como esperábamos, cuando el modelo de señal no refleja correctamente los mapas, el método es insesgado, encontrándose cierto incremento en el tamaño de las barras de error. Sin embargo, se ha observado que el método de bineado insesgado puede introducir algo de sesgo en este caso; un fenómeno esperado, debido al uso extra de la información registrada en el modelo de señal que se efectúa al compactar la matriz de Fisher. Cuando se aplica iterativamente el método (la forma de búsqueda de QML a diferencia de la más difusa de ML) mapa a mapa, las estimaciones convergen a los mismos valores, independientemente del modelo de señal de comienzo. Esto sucede tanto cuando se hace bineando como sin binear, lo que muestra que el método es robusto. Cuando lo hemos aplicado en el marco de un modelo simple para la estimación del parámetro r –la ratio de las potencias tensorial y escalar de inflación-, hemos encontrado el mismo comportamiento.

Los métodos pseudo-espectro son insesgados y mucho más rápidos computacionalmente que los métodos de máxima verosimilitud, pero no son óptimos. En las pruebas que hemos realizado hemos encontrado que producen estimaciones afectadas por incertidumbres significativamente mayores que las que genera QML, especialmente en las regiones correspondientes a las escalas grandes, que son críticas para la detección del modo tensorial de inflación. Además, debido a que se necesita apodizar las máscaras para crear zonas de transición suave entre los píxeles observados y los no observados para evitar el solapamiento en los espectros estimados, estos métodos son mucho más sensibles a la existencia de zonas dispersas en las que no se tiene señal del FCM. Debido a que QML opera sobre los mapas en el espacio de los píxeles, no está afectado por este problema, de modo que la pérdida de unos cuántos píxeles apenas supone un aumento del tamaño de las barras de error.

En definitiva, en este trabajo, presentamos un estudio detallado de las opciones, posibilidades y de los resultados en la estimación del espectro angular de potencias del FCM utilizando el estimador cuadrático de máxima verosimilitud. Con este propósito, hemos desarrollado una formulación matemática que reduce significativamente el número de operaciones y hemos escrito un código que la implementa. El código está publicado y es de acceso libre. Esperamos que se convierta en una herramienta útil de uso cómodo y sencillo para la comunidad científica.

BIBLIOGRAPHY

[1] E. Hubble, *A Relation between Distance and Radial Velocity among Extra-Galactic Nebulae*, *Proceedings of the National Academy of Science* **15** (1929) 168.

[2] F. Hoyle, *A New Model for the Expanding Universe*, *MNRAS* **108** (1948) 372.

[3] H. Bondi and T. Gold, *The Steady-State Theory of the Expanding Universe*, *MNRAS* **108** (1948) 252.

[4] R. A. Alpher, H. Bethe and G. Gamow, *The origin of chemical elements*, *Phys. Rev.* **73** (1948) 803.

[5] A. Friedmann, *Über die Krümmung des Raumes*, *Zeitschrift fur Physik* **10** (1922) 377.

[6] A. Friedmann, *Über die Möglichkeit einer Welt mit konstanter negativer Krümmung des Raumes*, *Zeitschrift fur Physik* **21** (1924) 326.

[7] G. Lemaître, *Expansion of the universe, The expanding universe*, *MNRAS* **91** (1931) 490.

[8] A. A. Penzias and R. W. Wilson, *A Measurement of Excess Antenna Temperature at 4080 Mc/s.*, *ApJ* **142** (1965) 419.

[9] R. H. Dicke, P. J. E. Peebles, P. G. Roll and D. T. Wilkinson, *Cosmic Black-Body Radiation.*, *ApJ* **142** (1965) 414.

[10] Planck Collaboration, N. Aghanim, Y. Akrami, F. Arroja, M. Ashdown, J. Aumont et al., *Planck 2018 results. I. Overview and the cosmological legacy of Planck*, *A&A* **641** (2020) A1 [1807.06205].

[11] D. J. Fixsen, *THE TEMPERATURE OF THE COSMIC MICROWAVE BACKGROUND*, *The Astrophysical Journal* **707** (2009) 916.

[12] A. Kogut, C. Lineweaver, G. F. Smoot, C. L. Bennett, A. Banday, N. W. Boggess et al., *Dipole Anisotropy in the COBE Differential Microwave Radiometers First-Year Sky Maps*, *ApJ* **419** (1993) 1 [`astro-ph/9312056`].

[13] Planck Collaboration, N. Aghanim, C. Armitage-Caplan, M. Arnaud, M. Ashdown, F. Atrio-Barandela et al., *Planck 2013 results. XXVII. Doppler boosting of the CMB: Eppur si muove*, *A&A* **571** (2014) A27 [`1303.5087`].

[14] J. A. Tauber, N. Mandolesi, J. L. Puget, T. Banos, M. Bersanelli, F. R. Bouchet et al., *Planck pre-launch status: The Planck mission*, *A&A* **520** (2010) A1.

[15] Planck Collaboration, Y. Akrami, M. Ashdown, J. Aumont, C. Baccigalupi, M. Ballardini et al., *Planck 2018 results. IV. Diffuse component separation*, *A&A* **641** (2020) A4 [`1807.06208`].

[16] B. Selmer, *Discretisation of the angular two-point correlation function*, Ph.D. thesis, University of Oslo, Jan., 2003.

[17] A. Kosowsky, *Cosmic microwave background polarization.*, *Annals of Physics* **246** (1996) 49 [`astro-ph/9501045`].

[18] W. Hu and M. White, *A CMB polarization primer*, *NewA* **2** (1997) 323 [`astro-ph/9706147`].

[19] M. Zaldarriaga and U. Seljak, *All-sky analysis of polarization in the microwave background*, *PhRvD* **55** (1997) 1830 [`astro-ph/9609170`].

[20] M. Kamionkowski, A. Kosowsky and A. Stebbins, *Statistics of cosmic microwave background polarization*, *PhRvD* **55** (1997) 7368 [`astro-ph/9611125`].

[21] Planck Collaboration, N. Aghanim, Y. Akrami, M. Ashdown, J. Aumont, C. Baccigalupi et al., *Planck 2018 results. VI. Cosmological parameters*, *A&A* **641** (2020) A6 [`1807.06209`].

[22] M. Hazumi, P. A. R. Ade, A. Adler, E. Allys, K. Arnold, D. Auguste et al., *LiteBIRD satellite: JAXA's new strategic L-class mission for all-sky surveys of cosmic microwave background polarization*, `2101.12449`.

[23] M. Li and X. Zhang, *Cosmological cpt violating effect on cmb polarization*, *Phys. Rev. D* **78** (2008) 103516.

[24] L. M. Capparelli, R. R. Caldwell and A. Melchiorri, *Cosmic birefringence test of the Hubble tension*, *PhRvD* **101** (2020) 123529 [1909.04621].

[25] A. Kosowsky and A. Loeb, *Faraday Rotation of Microwave Background Polarization by a Primordial Magnetic Field*, *ApJ* **469** (1996) 1 [astro-ph/9601055].

[26] P. Diego-Palazuelos, J. R. Eskilt, Y. Minami, M. Tristram, R. M. Sullivan, A. J. Banday et al., *Cosmic Birefringence from Planck Data Release 4*, *arXiv e-prints* (2022) arXiv:2201.07682 [2201.07682].

[27] R. K. Sachs and A. M. Wolfe, *Perturbations of a Cosmological Model and Angular Variations of the Microwave Background*, *ApJ* **147** (1967) 73.

[28] E. Martinez-Gonzalez, J. Sanz and J. Silk, *Anisotropies in the microwave sky due to nonlinear structures*, *The Astrophysical Journal* **355** (1990) L5.

[29] Zaldarriaga and Harari, *Analytic approach to the polarization of the cosmic microwave background in flat and open universes.*, *Physical review. D, Particles and fields* **52 6** (1995) 3276.

[30] M. J. Rees and D. W. Sciama, *Large-scale Density Inhomogeneities in the Universe*, *Nature* **217** (1968) 511.

[31] A. H. Jaffe and M. Kamionkowski, *Calculation of the ostriker-vishniac effect in cold dark matter models*, *Phys. Rev. D* **58** (1998) 043001.

[32] U. Seljak, *Gravitational Lensing Effect on Cosmic Microwave Background Anisotropies: A Power Spectrum Approach*, *ApJ* **463** (1996) 1 [astro-ph/9505109].

[33] M. Birkinshaw, *The sunyaev–zel'dovich effect*, *Physics Reports* **310** (1999) 97.

[34] N. Y. Gnedin and A. H. Jaffe, *Secondary cosmic microwave background anisotropies from cosmological reionization*, *The Astrophysical Journal* **551** (2001) 3.

[35] B. E. Robertson, R. S. Ellis, S. R. Furlanetto and J. S. Dunlop, *Cosmic Reionization and Early Star-forming Galaxies: A Joint Analysis of New Constraints from Planck and the Hubble Space Telescope*, *ApJL* **802** (2015) L19 [1502.02024].

[36] D. Baumann, *TASI Lectures on Inflation*, *arXiv e-prints* (2009) arXiv:0907.5424 [0907.5424].

[37] O. Lahav and A. R. Liddle, *The Cosmological Parameters (2021)*, *arXiv e-prints* (2022) arXiv:2201.08666 [2201.08666].

[38] M. Tristram, A. J. Banday, K. M. Górski, R. Keskitalo, C. R. Lawrence, K. J. Andersen et al., *Improved limits on the tensor-to-scalar ratio using BICEP and Planck*, *arXiv e-prints* (2021) arXiv:2112.07961 [2112.07961].

[39] K. M. Gorski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke et al., *HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere*, *The Astrophysical Journal* **622** (2005) 759.

[40] T. T. Soong, *Fundamentals of probability and statistics for engineers*. John Wiley & Sons, 2004.

[41] R. J. Barlow, *Statistics: a guide to the use of statistical methods in the physical sciences*, vol. 29. John Wiley & Sons, 1993.

[42] M. Tegmark, A. N. Taylor and A. F. Heavens, *Karhunen-Loève Eigenvalue Problems in Cosmology: How Should We Tackle Large Data Sets?*, *ApJ* **480** (1997) 22 [astro-ph/9603021].

[43] M. Tegmark, *How to measure cmb power spectra without losing information*, *Phys. Rev. D* **55** (1997) 5895.

[44] E. Hivon, K. M. Górski, C. B. Netterfield, B. P. Crill, S. Prunet and F. Hansen, *MASTER of the Cosmic Microwave Background Anisotropy Power Spectrum: A Fast Method for Statistical Analysis of Large and Complex Cosmic Microwave Background Data Sets*, *ApJ* **567** (2002) 2 [astro-ph/0105302].

[45] A. Kogut, D. N. Spergel, C. Barnes, C. L. Bennett, M. Halpern, G. Hinshaw et al., *First-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Temperature-Polarization Correlation*, *ApJS* **148** (2003) 161 [astro-ph/0302213].

[46] M. L. Brown, P. G. Castro and A. N. Taylor, *Cosmic microwave background temperature and polarization pseudo-C_ℓ estimators and covariances*, *MNRAS* **360** (2005) 1262 [astro-ph/0410394].

[47] J. Grain, M. Tristram and R. Stompor, *Polarized CMB power spectrum estimation using the pure pseudo-cross-spectrum approach*, *PhRvD* **79** (2009) 123515 [0903.2350].

[48] J. R. Bond, A. H. Jaffe and L. Knox, *Estimating the power spectrum of the cosmic microwave background*, *PhRvD* **57** (1998) 2117 [astro-ph/9708203].

[49] S. P. Oh, D. N. Spergel and G. Hinshaw, *An Efficient Technique to Determine the Power Spectrum from Cosmic Microwave Background Sky Maps*, *ApJ* **510** (1999) 551 [astro-ph/9805339].

[50] M. Tegmark and A. de Oliveira-Costa, *How to measure cmb polarization power spectra without losing information*, *Physical Review D* **64** (2001) .

[51] A. Gruppuso, A. de Rosa, P. Cabella, F. Paci, F. Finelli, P. Natoli et al., *New estimates of the CMB angular power spectra from the WMAP 5 year low-resolution data*, *MNRAS* **400** (2009) 463 [0904.0789].

[52] F. Schiavon, F. Finelli, A. Gruppuso, A. Marcos-Caballero, P. Vielva, R. G. Crittenden et al., *An optimal estimator for the CMB-LSS angular power spectrum and its application to WMAP and NVSS data*, *MNRAS* **427** (2012) 3044 [1203.3277].

[53] E. Gjerløw, L. P. L. Colombo, H. K. Eriksen, K. M. Górski, A. Gruppuso, J. B. Jewell et al., *Optimized Large-scale CMB Likelihood and Quadratic Maximum Likelihood Power Spectrum Estimation*, *ApJS* **221** (2015) 5 [1506.04273].

[54] S. Vanneste, S. Henrot-Versillé, T. Louis and M. Tristram, *Quadratic estimator for CMB cross-correlation*, *PhRvD* **98** (2018) 103526 [1807.02484].

[55] P. J. E. Peebles, *Statistical Analysis of Catalogs of Extragalactic Objects. I. Theory*, *ApJ* **185** (1973) 413.

[56] B. D. Wandelt, E. Hivon and K. M. Górski, *Cosmic microwave background anisotropy power spectrum statistics for high precision cosmology*, *PhRvD* **64** (2001) 083003 [astro-ph/0008111].

[57] G. Chon, A. Challinor, S. Prunet, E. Hivon and I. Szapudi, *Fast estimation of polarization power spectra using correlation functions*, *MNRAS* **350** (2004) 914 [astro-ph/0303414].

[58] M. Tristram, J. F. Macías-Pérez, C. Renault and D. Santos, *XSPECT, estimation of the angular power spectrum by computing cross-power spectra with analytical error bars*, *MNRAS* **358** (2005) 833 [astro-ph/0405575].

[59] D. Alonso, J. Sanchez, A. Slosar and LSST Dark Energy Science Collaboration, *A unified pseudo-C_ℓ framework*, *MNRAS* **484** (2019) 4127 [1809.09603].

[60] A. H. Guth, *Inflationary Universe: A Possible Solution to the Horizon and Flatness Problems*, *Physical Review D* **23** (1981) 347.

[61] A. D. Linde, *A new inflationary universe scenario: a possible solution of the horizon, flatness, homogeneity, isotropy and primordial monopole problems*, *Physics Letters B* **108** (1982) 389.

[62] A. A. Starobinsky, *Dynamics of phase transition in the new inflationary universe scenario and generation of perturbations*, *Physics Letters B* **117** (1982) 175.

[63] P. A. R. Ade, Z. Ahmed, M. Amiri, D. Barkats, R. B. Thakur, C. A. Bischoff et al., *Improved Constraints on Primordial Gravitational Waves using Planck, WMAP, and BICEP/Keck Observations through the 2018 Observing Season*, *PhRvL* **127** (2021) 151301 [2110.00483].

[64] M. Tristram, A. J. Banday, K. M. Górski, R. Keskitalo, C. R. Lawrence, K. J. Andersen et al., *Planck constraints on the tensor-to-scalar ratio*, *arXiv e-prints* (2020) arXiv:2010.01139 [2010.01139].

[65] Planck Collaboration, Y. Akrami, F. Arroja, M. Ashdown, J. Aumont, C. Baccigalupi et al., *Planck 2018 results. X. Constraints on inflation*, *A&A* **641** (2020) A10 [1807.06211].

[66] H. Hui, P. Ade, Z. Ahmed, R. Aikin, K. Alexander, D. Barkats et al., *BICEP Array: a multi-frequency degree-scale CMB polarimeter*, in *Millimeter, Submillimeter, and Far-Infrared Detectors and Instrumentation for Astronomy IX*, vol. 10708, p. 1070807, International Society for Optics and Photonics, 2018.

[67] P. Ade, J. Aguirre, Z. Ahmed, S. Aiola, A. Ali, D. Alonso et al., *The Simons Observatory: Science goals and forecasts*, *Journal of Cosmology and Astroparticle Physics* **2019** (2019) 056.

[68] K. N. Abazajian, P. Adshead, Z. Ahmed, S. W. Allen, D. Alonso, K. S. Arnold et al., *CMB-S4 Science Book*, *arXiv preprint arXiv:1610.02743* (2016) .

[69] Y. Sekimoto, P. Ade, K. Arnold, J. Aumont, J. Austermann, C. Baccigalupi et al., *Concept design of the LiteBIRD satellite for CMB B-mode polarization*, in *Proc. SPIE.*, vol. 10698 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, p. 106981Y, Aug, 2018, DOI.

[70] J. D. Bilbao-Ahedo, R. B. Barreiro, D. Herranz, P. Vielva and E. Martínez-González, *On the regularity of the covariance matrix of a discretized scalar field on the sphere*, *JCAP* 2 (2017) 022 [1701.06617].

[71] J. D. Bilbao-Ahedo, R. B. Barreiro, P. Vielva, E. Martínez-González and D. Herranz, *ECLIPSE: a fast Quadratic Maximum Likelihood estimator for CMB intensity and polarization power spectra*, *JCAP* **2021** (2021) 034 [2104.08528].

[72] A. Klinger, *Approximate pseudoinverse solutions to ill-conditioned linear systems*, *Journal of Optimization Theory and Applications* **2** (1968) 117.

[73] Planck Collaboration, P. A. R. Ade, N. Aghanim, C. Armitage-Caplan, M. Arnaud, M. Ashdown et al., *Planck 2013 results. XV. CMB power spectra and likelihood*, *A&A* **571** (2014) A15 [1303.5075].

[74] Planck Collaboration, N. Aghanim, M. Arnaud, M. Ashdown, J. Aumont, C. Baccigalupi et al., *Planck 2015 results. XI. CMB power spectra, likelihoods, and robustness of parameters*, *A&A* **594** (2016) A11 [1507.02704].

[75] G. Aslanyan, A. V. Manohar and A. P. S. Yadav, *The topology and size of the universe from cmb temperature and polarization data*, *Journal of Cosmology and Astroparticle Physics* **2013** (2013) 009.

[76] M. Tegmark, *Removing real-world foregrounds from cosmic microwave background maps*, *The Astrophysical Journal* **502** (1998) 1.

[77] H. K. Eriksen, A. J. Banday, K. M. Gorski and P. B. Lilje, *On foreground removal from the wilkinson microwave anisotropy probe data by an internal linear combination method: Limitations and implications*, *The Astrophysical Journal* **612** (2004) 633.

[78] H. K. Eriksen, G. Huey, R. Saha, F. K. Hansen, J. Dick, A. J. Banday et al., *A reanalysis of the 3 year wilkinson microwave anisotropy probe temperature power spectrum and likelihood*, *The Astrophysical Journal* **656** (2007) 641.

[79] Planck Collaboration, P. A. R. Ade, N. Aghanim, C. Armitage-Caplan, M. Arnaud, M. Ashdown et al., *Planck 2013 results. XII. Diffuse component separation*, *A&A* **571** (2014) A12 [1303.5072].

[80] P. Vielva and J. L. Sanz, *Analysis of non-gaussian cosmic microwave background maps based on the n-pdf. application to wilkinson microwave anisotropy probe data*, *Monthly Notices of the Royal Astronomical Society* **397** (2009) 837.

[81] A. F. Curto, E. Martinez-Gonzalez and R. B. Barreiro, *On the optimality of the spherical mexican hat wavelet estimator for the primordial non-gaussianity*, *Monthly Notices of the Royal Astronomical Society* **412** (2010) 1038.

[82] M. Cruz, N. Turok, P. Vielva, E. Martinez-Gonzalez and M. P. Hobson, *A cosmic microwave background feature consistent with a cosmic texture*, *Science* **318** (2007) 1612 .

[83] Planck Collaboration, P. A. R. Ade, N. Aghanim, Y. Akrami, P. K. Aluri, M. Arnaud et al., *Planck 2015 results. XVI. Isotropy and statistics of the CMB*, *A&A* **594** (2016) A16 [`1506.07135`].

[84] F. K. Chan and E. M. O'Neill, *Feasibility study of a quadrilateralized spherical cube earth data base*, 1975.

[85] M. Tegmark, *An icosahedron-based method for pixelizing the celestial sphere*, *The Astrophysical Journal* **470** (1996) .

[86] R. Crittenden, *Igloo pixelizations of the sky*, *Astrophysical Letters & Communications* **37** (1998) 377.

[87] A. G. Doroshkevich, P. D. Naselsky, O. V. Verkhodanov, D. I. Novikov, V. I. Turchaninov, I. D. Novikov et al., *Gauss-Legendre Sky Pixelization (glesp) for CMB Maps*, *International Journal of Modern Physics D* **14** (2005) 275 [`astro-ph/0305537`].

[88] C. L. Bennett, A. J. Banday, K. M. Górski, G. F. Hinshaw, P. D. Jackson, P. B. Keegstra et al., *Four-year cobe* dmr cosmic microwave background observations: Maps and basic results*, *The Astrophysical Journal* **464** (1996) .

[89] A. G. Doroshkevich, O. V. Verkhodanov, P. D. Naselsky, J. Kim, D. I. Novikov, V. I. Turchaninov et al., *The gauss-legendre sky pixelization for the cmb polarization (glesp-pol). errors due to pixelization of the cmb sky*, *International Journal of Modern Physics D* **20** (2011) 1053.

[90] Planck Collaboration, P. A. R. Ade, N. Aghanim, M. Arnaud, M. Ashdown, J. Aumont et al., *Planck 2015 results. XIII. Cosmological parameters*, *A&A* **594** (2016) A13 [`1502.01589`].

[91] K. Array, B. C. P. A. R. Ade, Z. Ahmed, R. W. Aikin, K. D. Alexander, D. Barkats et al., *Bicep2 / keck array vii: Matrix based e/b separation applied to bicep2 and the keck array*, *arXiv: Instrumentation and Methods for Astrophysics* (2016) .

[92] T. W. Anderson, *On the distribution of the two-sample cramer-von mises criterion*, *Annals of Mathematical Statistics* **33** (1962) 1148.

[93] D. J. Eisenstein, W. Hu and M. Tegmark, *Cosmic Complementarity: Joint Parameter Estimation from Cosmic Microwave Background Experiments and Redshift Surveys*, *ApJ* **518** (1999) 2 [`astro-ph/9807130`].

[94] D. Molinari, A. Gruppuso, G. Polenta, C. Burigana, A. De Rosa, P. Natoli et al., *A comparison of CMB angular power spectrum estimators at large scales: the TT case*, *MNRAS* **440** (2014) 957 [`1403.1089`].

[95] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon et al., *ScaLAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.

[96] J. A. Rubiño-Martín, R. Rebolo, M. Aguiar, R. Génova-Santos, F. Gómez-Reñasco, J. M. Herreros et al., *The QUIJOTE-CMB experiment: studying the polarisation of the galactic and cosmological microwave emissions*, in *Ground-based and Airborne Telescopes IV*, L. M. Stepp, R. Gilmozzi and H. J. Hall, eds., vol. 8444 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, p. 84442Y, Sept., 2012, DOI.

[97] J. A. Rubiño-Martín, R. Génova-Santos, R. Rebolo, J. M. Aguiar, Cózar-Castellano, F. Gómez-Reñasco, C. Gutiérrez et al., *The QUIJOTE experiment: project status and first scientific results*, in *Highlights on Spanish Astrophysics IX*, S. Arribas, A. Alonso-Herrero, F. Figueras, C. Hernández-Monteagudo, A. Sánchez-Lavega and S. Pérez-Hoyos, eds., pp. 99–107, Mar, 2017.

[98] E. De la Hoz, P. Vielva, R. B. Barreiro and E. Martínez-González, *On the detection of CMB B-modes from ground at low frequency*, *JCAP* **2020** (2020) 006 [`2002.12206`].

[99] A. Lewis and A. Challinor, *CAMB: Code for Anisotropies in the Microwave Background*, Feb., 2011.

[100] Y. Minami, H. Ochi, K. Ichiki, N. Katayama, E. Komatsu and T. Matsumura, *Simultaneous determination of the cosmic birefringence and miscalibrated polarization angles from CMB experiments*, *Progress of Theoretical and Experimental Physics* **2019** (2019) 083E02.

[101] F. Elsner, B. Leistedt and H. V. Peiris, *Unbiased pseudo-c_ℓ power spectrum estimation with mode projection*, *Monthly Notices of the Royal Astronomical Society* **465** (2016) 1847–1855.

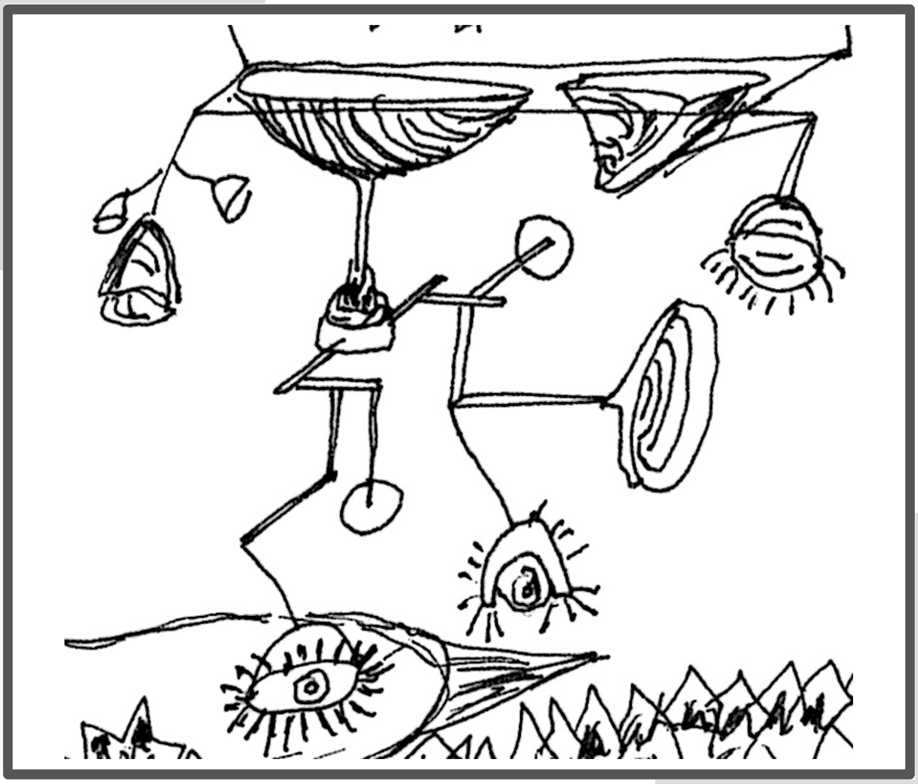
[102] G. Efstathiou, *Myths and truths concerning estimation of power spectra: the case for a hybrid estimator*, *MNRAS* **349** (2004) 603 [`astro-ph/0307515`].

[103] F. Guidi, J. A. Rubiño-Martín, A. E. Pelaez-Santos, R. T. Génova-Santos, M. Ashdown, R. B. Barreiro et al., *The PICASSO map-making code: application to a simulation of the QUIJOTE northern sky survey*, *MNRAS* **507** (2021) 3707 [`2108.09063`].

[104] E. de la Hoz, P. Diego-Palazuelos, E. Martínez-González, P. Vielva, R. B. Barreiro and J. D. Bilbao-Ahedo, *Determination of polarization angles in CMB experiments and application to CMB component separation analyses*, *JCAP* **2022** (2022) 032 [`2110.14328`].

[105] J. A. Rubiño-Martín et al., *QUIJOTE scientific results – IV. A northern sky survey in intensity and polarization at 10–20GHz with the Multi-Frequency Instrument*, *[In preparation]* .

[106] D. Scott, M. Srednicki and M. White, *“Sample Variance” in Small-Scale Cosmic Microwave Background Anisotropy Experiments*, *ApJL* **421** (1994) L5 [`astro-ph/9305030`].



El observador universal

El Fondo Cósmico de Microondas (FCM) codifica información clave que conecta de forma directa las observaciones que realizamos aquí y en el presente con el estado del universo temprano. Además, de acuerdo con nuestros modelos teóricos, el conocimiento de las propiedades estadísticas del FCM conduce al conocimiento sobre las características de nuestro Universo en el límite de los instantes iniciales, y a predecir su futuro. La información relevante que nos transmite el FCM está codificada en su espectro angular de potencias. Este trabajo está dedicado al estudio del Estimador Cuadrático de Máxima Verosimilitud (QML), un método óptimo de estimación del espectro de potencia. Analizamos sus propiedades, las condiciones de tipo matemático que se han de cumplir para utilizarlo, soluciones en el caso en el que alguna no se cumpla y estudiamos el rendimiento del método en múltiples situaciones de interés práctico en el presente y el futuro inmediato. El método QML conlleva una alta carga computacional. Tras un análisis detallado de los entresijos matemáticos propios del método, hemos desarrollado una implementación óptima que formalmente permite aplicarlo con los medios técnicos actuales en situaciones que hasta ahora eran inviables. Para ponerlo en práctica, hemos escrito un código que implementa nuestra formulación del método, capaz de aprovechar la potencia de cálculo de los supercomputadores. El código es de acceso público y libre.

The Cosmic Microwave Background (CMB) encodes key information that directly connects the observations we make here and now with the state of the early universe. Moreover, according to our theoretical models, knowledge of the statistical properties of the CMB leads to knowledge about the characteristics of our Universe in the limit of the initial instants, and to predict its future. The relevant information provided by the CMB is encoded in its angular power spectrum. This study is devoted to the exploration of the Quadratic Maximum Likelihood Estimator (QML), an optimal method of estimating the power spectrum. We analyze its properties, the mathematical conditions that must be fulfilled to use it, solutions in the case that some of them are not fulfilled and we study the performance of the method in multiple situations of practical interest in the present and the immediate future. The QML method is computationally intensive. After a detailed analysis of the mathematical intricacies of the method, we have developed an optimal implementation that formally allows it to be applied with current technical means in situations that were unfeasible until now. To put it into practice, we have written a code that implements our formulation of the method, capable of taking advantage of the computing power of supercomputers. The code is publicly and freely available.

UC
UNIVERSIDAD
DE CANTABRIA

Escuela
de Doctorado