# CMB Data Analysis Pipeline

Davide Maino

Departement of Physics
University of Milano

Lab Course I

# Outline

From TOD to maps

The Likelihood function

Application to CMB experiments

CMB Power Spectrum extraction

- MASTER approach

Extracting and Forecasting cosmological parameters

# Outline

From TOD to maps

The Likelihood function

Application to CMB experiments

CMB Power Spectrum extraction

MASTER approach

Extracting and Forecasting cosmological parameters

# Time Ordered Data

- The starting point for any CMB data analysis is the “Time Ordered Data” (TOD) stream coming out of the instrument
- We assume a data model:

$$d_t = P_{ti} m_i + n_t$$

where $i$ is the pixel position index, $d_t$ is the data stream ordered in time $t$, $n_t$ is the instrument noise contribution as a function of time and $P_{ti}$ is the pointing matrix which maps time space into pixel space
- Noise is expected to be known at least in its statistical properties (e.g. variance or frequency spectrum)

$$\langle n_t n_{t'} \rangle = \mathcal{N}_{tt'}$$

- We want to extract $m_i$ from the data (inversion problem)

# Time Ordered Data

- The starting point for any CMB data analysis is the “Time Ordered Data” (TOD) stream coming out of the instrument
- We assume a data model:

$$d_t = P_{ti} m_i + n_t$$

  where $i$ is the pixel position index, $d_t$ is the data stream ordered in time $t$, $n_t$ is the instrument noise contribution as a function of time and $P_{ti}$ is the pointing matrix which maps time space into pixel space
- Noise is expected to be known at least in its statistical properties (e.g. variance or frequency spectrum)

$$\langle n_t n_{t'} \rangle = \mathcal{N}_{tt'}$$

- We want to extract $m_i$ from the data (inversion problem)

# Time Ordered Data

- The starting point for any CMB data analysis is the "Time Ordered Data" (TOD) stream coming out of the instrument
- We assume a data model:

$$d_t = P_{ti} m_i + n_t$$

  where $i$ is the pixel position index, $d_t$ is the data stream ordered in time $t$, $n_t$ is the instrument noise contribution as a function of time and $P_{ti}$ is the pointing matrix which maps time space into pixel space
- Noise is expected to be known at least in its statistical properties (e.g. variance or frequency spectrum)

$$\langle n_t n_{t'} \rangle = \mathcal{N}_{tt'}$$

- We want to extract $m_i$ from the data (inversion problem)

# Time Ordered Data

- The starting point for any CMB data analysis is the “Time Ordered Data” (TOD) stream coming out of the instrument
- We assume a data model:

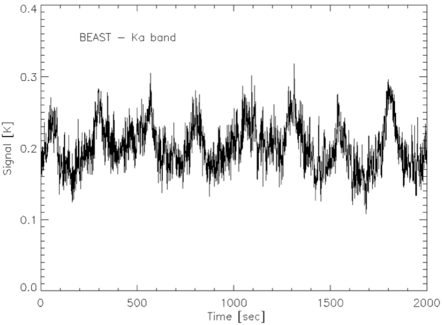
$$d_t = P_{ti} m_i + n_t$$

  where $i$ is the pixel position index, $d_t$ is the data stream ordered in time $t$, $n_t$ is the instrument noise contribution as a function of time and $P_{ti}$ is the pointing matrix which maps time space into pixel space
- Noise is expected to be known at least in its statistical properties (e.g. variance or frequency spectrum)

$$\langle n_t n_{t'} \rangle = \mathcal{N}_{tt'}$$

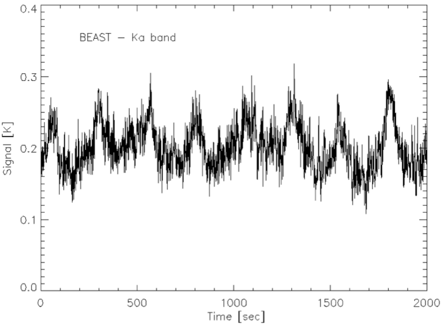
- We want to extract $m_i$ from the data (inversion problem)

# Practical Issues




- Present ground based, balloon-borne experiments and even more from satellite have TOD of the order of $10^{10}$ time samples with output maps of about $10^6 \div 10^7$ pixels
- Inversion problem scales as $N^3_{\rm pixels}$: it is not a trivial task $\longrightarrow$ CPU time and memory requirement

# Practical Issues




- Present ground based, balloon-borne experiments and even more from satellite have TOD of the order of $10^{10}$ time samples with output maps of about $10^6 \div 10^7$ pixels
- Inversion problem scales as $N^3_{\rm pixels}$: it is not a trivial task $\longrightarrow$ CPU time and memory requirement

# Pointing Matrix

- It encodes the way in which the sky is actually observed by the experiment
- Simplest case: a total power radiometer or a bolometer observing a given direction at a given time:

$$P_{ij} = \begin{pmatrix} 0 & 0 & 1 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 1 & \cdots & 0 \end{pmatrix}$$

- More complex case for e.g. *WMAP* which observed at the same time two points separate by $\approx 141^\circ$ in the sky

$$P_{ij} = \begin{pmatrix} 0 & 0 & 1 & \cdots & -1 \\ 1 & 0 & -1 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & -1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -1 & 0 & 1 & \cdots & 0 \end{pmatrix}$$

# Pointing Matrix

- It encodes the way in which the sky is actually observed by the experiment
- Simplest case: a total power radiometer or a bolometer observing a given direction at a given time:

$$P_{ij} = \begin{pmatrix} 0 & 0 & 1 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 1 & \cdots & 0 \end{pmatrix}$$

- More complex case for e.g. *WMAP* which observed at the same time two points separate by $\approx 141^\circ$ in the sky

$$P_{ij} = \begin{pmatrix} 0 & 0 & 1 & \cdots & -1 \\ 1 & 0 & -1 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & -1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -1 & 0 & 1 & \cdots & 0 \end{pmatrix}$$

# Pointing Matrix

- It encodes the way in which the sky is actually observed by the experiment
- Simplest case: a total power radiometer or a bolometer observing a given direction at a given time:

$$P_{ij} = \begin{pmatrix} 0 & 0 & 1 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 1 & \cdots & 0 \end{pmatrix}$$

- More complex case for e.g. *WMAP* which observed at the same time two points separate by $\approx 141^\circ$ in the sky

$$P_{ij} = \begin{pmatrix} 0 & 0 & 1 & \cdots & -1 \\ 1 & 0 & -1 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & -1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -1 & 0 & 1 & \cdots & 0 \end{pmatrix}$$

# Pointing Matrix

- In general pointing will includes also the effect of beam convolution

$$T(\theta_0, \phi_0) = \frac{\int d\Omega P(\theta - \theta_0, \phi - \phi_0) T(\theta, \phi)}{\int d\Omega P(\theta, \phi)}$$

- During data analysis beam and pointing are treated separately and we derive $m_i$ as convolved by instrument beam and scanning strategy

# Pointing Matrix

- In general pointing will includes also the effect of beam convolution

$$T(\theta_0, \phi_0) = \frac{\int d\Omega P(\theta - \theta_0, \phi - \phi_0) T(\theta, \phi)}{\int d\Omega P(\theta, \phi)}$$

- During data analysis beam and pointing are treated separately and we derive $m_i$ as convolved by instrument beam and scanning strategy

# Map-Making

- We would to extract the best possible map from our TOD. Let us construct a simple chi-squared estimator:

$$\chi^2 = \sum_{tt'ij} (d_t - P_{ti} m_i) \mathcal{N}^{-1}_{tt'} (d_{t'} - P_{t'j} m_j)$$

and minimize it with respect to the signal we are interested in (the map)

$$\frac{\partial \chi^2}{\partial m_i} = -2 \sum_{tt'j} P_{ti} \mathcal{N}^{-1}_{tt'} (d_{t'} - P_{t'j} m_j) = 0$$

which has a solution given by

$$\sum_{tt'j} P_{ti} \mathcal{N}^{-1}_{tt'} P_{t'j} m_j = \sum_{tt'j} P_{ti} \mathcal{N}^{-1}_{tt'} d_{t'}$$

# Map-Making

- We search for an estimator $\hat{m}_i$ solution to that equation

$$\hat{m}_i = (\sum_{tt'} P_{ti} \mathcal{N}_{tt'}^{-1} P_{t'j})^{-1} P_{ti} \mathcal{N}_{tt'}^{-1} d_{t'}$$

  where the first term states how the noise is distributed on the map according to noise properties and observing strategy

- The same equation but in matrix notation reads

$$\hat{m} = (P^T \mathcal{N}^{-1} P)^{-1} P^T \mathcal{N}^{-1} d$$

# Map-Making

- We search for an estimator $\hat{m}_i$ solution to that equation

$$\hat{m}_i = (\sum_{tt'} P_{ti}\mathcal{N}_{tt'}^{-1}P_{t'j})^{-1}P_{ti}\mathcal{N}_{tt'}^{-1}d_{t'}$$

  where the first term states how the noise is distributed on the map according to noise properties and observing strategy
- The same equation but in matrix notation reads

$$\hat{m} = (P^T\mathcal{N}^{-1}P)^{-1}P^T\mathcal{N}^{-1}d$$

# Map-Making

- Is this an optimal (un-biased & minimum variance) estimator?
- Let us rewrite $\hat{m}$ according to our data model:

$$\begin{aligned}\hat{m} &= (P^T \mathcal{N}^{-1} P)^{-1} P^T \mathcal{N}^{-1} (Pm + n) \\ &= m - (P^T \mathcal{N}^{-1} P)^{-1} P^T \mathcal{N}^{-1} n\end{aligned}$$

  and as long as $\langle n \rangle = 0$ then $\langle \hat{m} \rangle = m \rightarrow$ un-biased
- Compute the variance of the estimator

$$\begin{aligned}C_N &= \langle (m - \hat{m})(m - \hat{m})^T \rangle \\ &= (P^T \mathcal{N}^{-1} P)^{-1} P^T \mathcal{N}^{-1} \langle n n^T \rangle \mathcal{N}^{-1} P (P^T \mathcal{N}^{-1} P)^{-1}\end{aligned}$$

  since $\langle n n^T \rangle = \mathcal{N}$ we find

$$C_N = (P^T \mathcal{N}^{-1} P)^{-1}$$

  which could be demonstrated to be minimum variance.

# Map-Making

- Is this an optimal (un-biased & minimum variance) estimator?
- Let us rewrite $\hat{m}$ according to our data model:

$$\begin{aligned}\hat{m} &= (P^T \mathcal{N}^{-1} P)^{-1} P^T \mathcal{N}^{-1} (Pm + n) \\ &= m - (P^T \mathcal{N}^{-1} P)^{-1} P^T \mathcal{N}^{-1} n\end{aligned}$$

  and as long as $\langle n \rangle = 0$ then $\langle \hat{m} \rangle = m \rightarrow$ un-biased
- Compute the variance of the estimator

$$\begin{aligned}C_N &= \langle (m - \hat{m})(m - \hat{m})^T \rangle \\ &= (P^T \mathcal{N}^{-1} P)^{-1} P^T \mathcal{N}^{-1} \langle n n^T \rangle \mathcal{N}^{-1} P (P^T \mathcal{N}^{-1} P)^{-1}\end{aligned}$$

  since $\langle n n^T \rangle = \mathcal{N}$ we find

$$C_N = (P^T \mathcal{N}^{-1} P)^{-1}$$

  which could be demonstrated to be minimum variance.

# Map-Making

- Is this an optimal (un-biased & minimum variance) estimator?
- Let us rewrite $\hat{m}$ according to our data model:

$$\begin{aligned}\hat{m} &= (P^T\mathcal{N}^{-1}P)^{-1}P^T\mathcal{N}^{-1}(Pm+n)\\ &= m-(P^T\mathcal{N}^{-1}P)^{-1}P^T\mathcal{N}^{-1}n\end{aligned}$$

  and as long as $\langle n\rangle = 0$ then $\langle \hat{m}\rangle = m \rightarrow$ un-biased
- Compute the variance of the estimator

$$\begin{aligned}C_N &= \langle (m-\hat{m})(m-\hat{m})^T\rangle\\ &= (P^T\mathcal{N}^{-1}P)^{-1}P^T\mathcal{N}^{-1}\langle nn^T\rangle\mathcal{N}^{-1}P(P^T\mathcal{N}^{-1}P)^{-1}\end{aligned}$$

  since $\langle nn^T\rangle = \mathcal{N}$ we find

$$C_N = (P^T\mathcal{N}^{-1}P)^{-1}$$

  which could be demonstrated to be minimum variance.

# Exercise

- Demonstrate that if noise in TOD is un-correlated in time and uniform (i.e. white noise) the optimal map-making solution is given by

$$m_i = \frac{1}{N_{\mathrm{obs,i}}} \sum_t P_{it}^T d_t$$

  where $N_{\mathrm{obs,i}}$ is the total number of observations for pixel $i$
- comment the result

# Exercise

- Demonstrate that if noise in TOD is un-correlated in time and uniform (i.e. white noise) the optimal map-making solution is given by

$$m_i = \frac{1}{N_{\rm obs,i}} \sum_t P^T_{it} d_t$$

  where $N_{\rm obs,i}$ is the total number of observations for pixel $i$
- comment the result

# Outline

From TOD to maps

**The Likelihood function**

Application to CMB experiments

CMB Power Spectrum extraction

- MASTER approach

Extracting and Forecasting cosmological parameters

# Generalities

- The probability that a given experiment would get the data it did given a theory
  - location of the likelihood maximum in the theory parameters space
  - width of the likelihood gives error estimation on parameters
- **example:** Find out the weight of an object i.e. find the best value out of a number of measurements and its uncertainty

# Generalities

- The probability that a given experiment would get the data it did given a theory
  - location of the likelihood maximum in the theory parameters space
  - width of the likelihood gives error estimation on parameters
- **example:** Find out the weight of an object i.e. find the best value out of a number of measurements and its uncertainty

# Generalities

- The probability that a given experiment would get the data it did given a theory
  - location of the likelihood maximum in the theory parameters space
  - width of the likelihood gives error estimation on parameters
- **example:** Find out the weight of an object i.e. find the best value out of a number of measurements and its uncertainty

# Generalities

- The probability that a given experiment would get the data it did given a theory
  - location of the likelihood maximum in the theory parameters space
  - width of the likelihood gives error estimation on parameters
- **example:** Find out the weight of an object i.e. find the best value out of a number of measurements and its uncertainty

# Definitions

- Model: $d_i = w + n_i$ where $n$ is drawn from a Gaussian distribution with zero mean and variance $\sigma_w^2$ $\rightarrow$ two free parameters: $w$ and $\sigma_w$
- With $N$ measurements, the probability $P[\{d_i\}|w, \sigma_w]$ is

$$\mathcal{L}(d; w, \sigma_w) = \frac{1}{(2\pi\sigma_w^2)^{N/2}} \exp\left\{-\frac{\sum_i (d_i - w)^2}{2\sigma_w^2}\right\}$$

- Bayes Theorem states

$$P[w, \sigma_w|\{d_i\}] = \frac{P[\{d_i\}|w, \sigma_w] P[w, \sigma_w]}{P[\{d_i\}]}$$

where $P[w, \sigma_w]$ is the *p*rior probability i.e. any prior knowledge on the two parameters, and $P[\{d_i\}]$ normalize the probability to 1.

# Definitions

- Model: $d_i = w + n_i$ where $n$ is drawn from a Gaussian distribution with zero mean and variance $\sigma_w^2 \rightarrow$ two free parameters: $w$ and $\sigma_w$
- With $N$ measurements, the probability $P[\{d_i\}|w, \sigma_w]$ is

$$\mathcal{L}(d; w, \sigma_w) = \frac{1}{(2\pi\sigma_w^2)^{N/2}} \exp\left\{ -\frac{\sum_i (d_i - w)^2}{2\sigma_w^2} \right\}$$

- Bayes Theorem states

$$P[w, \sigma_w|\{d_i\}] = \frac{P[\{d_i\}|w, \sigma_w] P[w, \sigma_w]}{P[\{d_i\}]}$$

where $P[w, \sigma_w]$ is the *p*rior probability i.e. any prior knowledge on the two parameters, and $P[\{d_i\}]$ normalize the probability to 1.

# Definitions

- Model: $d_i = w + n_i$ where $n$ is drawn from a Gaussian distribution with zero mean and variance $\sigma_w^2$ → two free parameters: $w$ and $\sigma_w$
- With $N$ measurements, the probability $P[\{d_i\}|w, \sigma_w]$ is

$$\mathcal{L}(d; w, \sigma_w) = \frac{1}{(2\pi\sigma_w^2)^{N/2}} \exp\left\{-\frac{\sum_i (d_i - w)^2}{2\sigma_w^2}\right\}$$

- Bayes Theorem states

$$P[w, \sigma_w|\{d_i\}] = \frac{P[\{d_i\}|w, \sigma_w] P[w, \sigma_w]}{P[\{d_i\}]}$$

where $P[w, \sigma_w]$ is the *p*rior probability i.e. any prior knowledge on the two parameters, and $P[\{d_i\}]$ normalize the probability to 1.

# Parameter Best Values

- The maximum of probability is given by the maximum of $\mathcal{L}$ w.r.t. the parameters $w$ and $\sigma_w$

$$\frac{\partial \mathcal{L}}{\partial w} \propto \sum_j (d_j - w) \exp\left\{ -\frac{\sum_i (d_i - w)^2}{2\sigma_w^2} \right\} = 0$$

- $\sum_i (d_i - w) = 0 \rightarrow w = \hat{w} = \frac{1}{N} \sum_i d_i$ as expected
- the most probable value for $\sigma_w^2 = \frac{1}{N} \sum_i (d_i - \hat{w})^2$ again as expected

# Parameter Best Values

- The maximum of probability is given by the maximum of $\mathcal{L}$ w.r.t. the parameters $w$ and $\sigma_w$

$$\frac{\partial \mathcal{L}}{\partial w} \propto \sum_j (d_j - w) \exp\left\{ -\frac{\sum_i (d_i - w)^2}{2\sigma_w^2} \right\} = 0$$

- $\sum_i (d_i - w) = 0 \rightarrow w = \hat{w} = \frac{1}{N} \sum_i d_i$ as expected
- the most probable value for $\sigma_w^2 = \frac{1}{N} \sum_i (d_i - \hat{w})^2$ again as expected

# Parameter Best Values

- The maximum of probability is given by the maximum of $\mathcal{L}$ w.r.t. the parameters $w$ and $\sigma_w$

$$\frac{\partial \mathcal{L}}{\partial w} \propto \sum_j (d_j - w) \exp\left\{ -\frac{\sum_i (d_i - w)^2}{2\sigma_w^2} \right\} = 0$$

- $\sum_i (d_i - w) = 0 \rightarrow w = \hat{w} = \frac{1}{N} \sum_i d_i$ as expected
- the most probable value for $\sigma_w^2 = \frac{1}{N} \sum_i (d_i - \hat{w})^2$ again as expected

# Parameter Best Values

- The error on the best value is simply the width of $\mathcal{L}$
- Assuming $\mathcal{L}$ to be Gaussian in the parameters we know the variance of a Gaussian distribution $\rightarrow \sigma_W/N^{1/2}$
- Only two numbers are needed to describe the $N$ measurements: the best estimate $\hat{w}$ and the associated error $\sigma/N^{1/2}$.

$$\mathcal{L} = \frac{1}{\sqrt{2\pi C_N}} \exp\left\{-\frac{(w-\hat{w})^2}{2C_N}\right\}$$

# Parameter Best Values

- The error on the best value is simply the width of $\mathcal{L}$
- Assuming $\mathcal{L}$ to be Gaussian in the parameters we know the variance of a Gaussian distribution $\rightarrow \sigma_w/N^{1/2}$
- Only two numbers are needed to describe the $N$ measurements: the best estimate $\hat{w}$ and the associated error $\sigma/N^{1/2}$.

$$\mathcal{L} = \frac{1}{\sqrt{2\pi C_N}} \exp\left\{-\frac{(w-\hat{w})^2}{2C_N}\right\}$$

# Parameter Best Values

- The error on the best value is simply the width of $\mathcal{L}$
- Assuming $\mathcal{L}$ to be Gaussian in the parameters we know the variance of a Gaussian distribution $\rightarrow \sigma_w/N^{1/2}$
- Only two numbers are needed to describe the $N$ measurements: the best estimate $\hat{w}$ and the associated error $\sigma/N^{1/2}$.

$$\mathcal{L} = \frac{1}{\sqrt{2\pi C_N}} \exp\left\{-\frac{(w-\hat{w})^2}{2C_N}\right\}$$

# Outline

From TOD to maps

The Likelihood function

**Application to CMB experiments**

CMB Power Spectrum extraction

- MASTER approach

Extracting and Forecasting cosmological parameters

# CMB Likelihood

Is the estimated map with map-making a ML solution?

- the model is $d_t = P_{ti} m_i + n_t$ where $n$ is Gaussian distributed
- the likelihood for this model is

$$\mathcal{L} = \frac{1}{(2\pi)^{N_t/2}\sqrt{|\mathcal{N}|}} \exp\left\{ -\frac{1}{2}(d_t - P_{ti} m_i)\mathcal{N}^{-1}_{tt'}(d_{t'} - P_{t'j} m_j) \right\}$$

and the ML solution is obtained minimizing the quantity in the exponential that is exactly the $\chi^2$ previously defined.

# CMB Likelihood

Is the estimated map with map-making a ML solution?

- the model is $d_t = P_{ti} m_i + n_t$ where $n$ is Gaussian distributed
- the likelihood for this model is

$$\mathcal{L} = \frac{1}{(2\pi)^{N_t/2}\sqrt{|\mathcal{N}|}} \exp\left\{ -\frac{1}{2}(d_t - P_{ti} m_i)\mathcal{N}^{-1}_{tt'}(d_{t'} - P_{t'j} m_j) \right\}$$

  and the ML solution is obtained minimizing the quantity in the exponential that is exactly the $\chi^2$ previously defined.

# CMB Likelihood

Is the estimated map with map-making a ML solution?

- the model is $d_t = P_{ti} m_i + n_t$ where $n$ is Gaussian distributed
- the likelihood for this model is

$$\mathcal{L} = \frac{1}{(2\pi)^{N_t/2}\sqrt{|\mathcal{N}|}} \exp\left\{ -\frac{1}{2}(d_t - P_{ti} m_i) \mathcal{N}^{-1}_{tt'} (d_{t'} - P_{t'j} m_j) \right\}$$

and the ML solution is obtained minimizing the quantity in the exponential that is exactly the $\chi^2$ previously defined.

# CMB Likelihood

**Theorem: Cramer-Rao Inequality**

No estimator can measure $m_i$ with an error smaller than the diagonal elements of $\mathbf{F}^{-1}$ where $\mathbf{F}$ is the Fisher or Curvature matrix defined as

$$F_{m,ij} = -\frac{\partial^2 \mathcal{L}}{\partial m_i \partial m_j}$$

# CMB Likelihood

Is the estimated map with map-making a ML solution?

- Computing second derivatives of $\mathcal{L}$ w.r.t. $m$ we found that $\mathbf{F} = C_N^{-1}$: the map-making solution is a ML solution with minimum variance
- No information is lost passing from TOD to ML map but a large ($\simeq 10^3$) compression factor is obtained. Further analysis can be made directly on maps

# CMB Likelihood

Is the estimated map with map-making a ML solution?

- Computing second derivatives of $\mathcal{L}$ w.r.t. $m$ we found that $\mathbf{F} = C_N^{-1}$: the map-making solution is a ML solution with minimum variance
- No information is lost passing from TOD to ML map but a large ($\simeq 10^3$) compression factor is obtained. Further analysis can be made directly on maps

# Outline

# Band Power Estimation

- Once you have a CMB map, the next step in data compression is the extraction of (band) Angular Power spectrum from the map
- Let us construct a new likelihood where the sky signal $m_i$ are the data
- we know only the statistical properties of the set of $m_i$: no theory will give you the CMB value in a given pixel in the sky!

# Band Power Estimation

- Once you have a CMB map, the next step in data compression is the extraction of (band) Angular Power spectrum from the map
- Let us construct a new likelihood where the sky signal $m_i$ are the data
- we know only the statistical properties of the set of $m_i$: no theory will give you the CMB value in a given pixel in the sky!

# Band Power Estimation

- Once you have a CMB map, the next step in data compression is the extraction of (band) Angular Power spectrum from the map
- Let us construct a new likelihood where the sky signal $m_i$ are the data
- we know only the statistical properties of the set of $m_i$: no theory will give you the CMB value in a given pixel in the sky!

# Band Power Estimation

Consider only the diagonal element of the signal covariance matrix

$$C_{S,ii} = \langle m_i m_i \rangle$$

where $\langle \cdots \rangle$ is an ensemble average and

$$m_i = \int d\hat{n} \Theta(\hat{n}) B_i(\hat{n})$$

where $B_i$ is the antenna beam pattern at pixel $i$ and $\Theta(\hat{n})$ is the "true" sky signal (CMB anisotrpies) in the direction $\hat{n}$.
Let us square the last equation and expand anisotropies in spherical harmonic $\Theta(\hat{n}) = \sum_{\ell m} a_{\ell m} Y_{\ell m}(\hat{n})$

$$C_{S,ii} = \int d\hat{n} \int d\hat{n}' B_i(\hat{n}) B_i(\hat{n}') \sum_{\ell m} Y_{\ell m}(\hat{n}) \sum_{\ell' m'} Y^{\star}_{\ell' m'}(\hat{n}') \langle a_{\ell m} a^{\star}_{\ell' m'} \rangle$$

# Band Power Estimation

But $\langle a_{\ell m} a^{\star}_{\ell' m'} \rangle = \delta_{\ell\ell'}\delta_{mm'} C_\ell$ where $C_\ell$ is called angular power spectrum and we find

$$C_{S,ii} = \int d\hat{n} \int d\hat{n}' B_i(\hat{n}) B_i(\hat{n}') \sum_\ell C_\ell \sum_{\ell m} Y_{\ell m}(\hat{n}) Y^{\star}_{\ell m}(\hat{n}')$$

Since $\sum_m Y_{\ell m}(\hat{n}) Y_{\ell m}(\hat{n}') = 2(\ell + 1) P_\ell(\hat{n} \cdot \hat{n}')/4\pi$ and hence

$$C_{S,ii} = \sum_\ell \frac{2\ell + 1}{4\pi} C_\ell W_{\ell,ii}$$

where $W_{\ell,ii}$ is called Window Function. For an experiment like *WMAP*, or PLANCK we may approximate the Window Function as a Gaussian

$$W_{\ell,ii} = \exp(-\ell^2 \sigma^2)$$

with $\sigma = \mathrm{FWHM}/\sqrt{8\ln(2)}$

# Band Power Estimation

- For Gaussian theories (inflation) the likelihood is

$$\mathcal{L}(m_i) = \frac{1}{(2\pi)^{N_p/2}\sqrt{|C|}}\exp(-\frac{1}{2}m_i C_{ij}^{-1} m_j)$$

where $C = C_S + C_N$ and $N_p$ is the total number of pixels in the map

- It is clear that $\mathcal{L}$ is Gaussian in $m_i$ but our variables of interest are band power $\rightarrow$ likelihood not Gaussian and there is no easy way to compute this.

# Band Power Estimation

- For Gaussian theories (inflation) the likelihood is

$$\mathcal{L}(m_i) = \frac{1}{(2\pi)^{N_p/2}\sqrt{|C|}}\exp(-\frac{1}{2}m_i C_{ij}^{-1} m_j)$$

where $C = C_S + C_N$ and $N_p$ is the total number of pixels in the map

- It is clear that $\mathcal{L}$ is Gaussian in $m_i$ but our variables of interest are band power $\rightarrow$ likelihood not Gaussian and there is no easy way to compute this.

# Band Power Estimation

- Brute force? $C = C_S + C_N$. We know that
  - in pixel space with un-correlated noise $C_N$ is diagonal
  - in harmonic space $C_S$ is diagonal with power spectrum on the diagonal
- Computing the determinant and inverse of $C$ is not trivial and scales as $N_p^3$
- Brute force worked for Boomerang 98 with “only” $N_p = 57,000$

# Band Power Estimation

- Brute force? $C = C_S + C_N$. We know that
  - in pixel space with un-correlated noise $C_N$ is diagonal
  - in harmonic space $C_S$ is diagonal with power spectrum on the diagonal
- Computing the determinant and inverse of $C$ is not trivial and scales as $N_p^3$
- Brute force worked for Boomerang 98 with “only” $N_p = 57,000$

# Band Power Estimation

- Brute force? $C = C_S + C_N$. We know that
  - in pixel space with un-correlated noise $C_N$ is diagonal
  - in harmonic space $C_S$ is diagonal with power spectrum on the diagonal
- Computing the determinant and inverse of $C$ is not trivial and scales as $N_p^3$
- Brute force worked for Boomerang 98 with "only" $N_p = 57,000$

# Band Power Estimation

- Brute force? $C = C_S + C_N$. We know that
  - in pixel space with un-correlated noise $C_N$ is diagonal
  - in harmonic space $C_S$ is diagonal with power spectrum on the diagonal
- Computing the determinant and inverse of $C$ is not trivial and scales as $N_p^3$
- Brute force worked for Boomerang 98 with "only" $N_p = 57,000$

# Band Power Estimation

- Brute force? $C = C_S + C_N$. We know that
  - in pixel space with un-correlated noise $C_N$ is diagonal
  - in harmonic space $C_S$ is diagonal with power spectrum on the diagonal
- Computing the determinant and inverse of $C$ is not trivial and scales as $N_p^3$
- Brute force worked for Boomerang 98 with "only" $N_p = 57,000$

# Band Power Estimation

- How to solve the issue? Different approaches:
  1. Transform data to signal-to-noise basis throwing away contaminated modes (requires at least one $N_p^3$ operation)
  2. Use specific observing symmetries i.e. for azimuthal scanning strategy likelihood computation scales as $N_p^{3/2}$ but works only in ideal cases
  3. Compute real space 2-pt correlation function and then get power spectrm
- or $\cdots$

# Band Power Estimation

- How to solve the issue? Different approaches:
  1. Transform data to signal-to-noise basis throwing away contaminated modes (requires at least one $N_p^3$ operation)
  2. Use specific observing symmetries i.e. for azimuthal scanning strategy likelihood computation scales as $N_p^{3/2}$ but works only in ideal cases
  3. Compute real space 2-pt correlation function and then get power spectrm
- or $\cdots$

# Band Power Estimation

- How to solve the issue? Different approaches:
  1. Transform data to signal-to-noise basis throwing away contaminated modes (requires at least one $N_p^3$ operation)
  2. Use specific observing symmetries i.e. for azimuthal scanning strategy likelihood computation scales as $N_p^{3/2}$ but works only in ideal cases
  3. Compute real space 2-pt correlation function and then get power spectrm
- or $\cdots$

# Band Power Estimation

- How to solve the issue? Different approaches:
  1. Transform data to signal-to-noise basis throwing away contaminated modes (requires at least one $N_p^3$ operation)
  2. Use specific observing symmetries i.e. for azimuthal scanning strategy likelihood computation scales as $N_p^{3/2}$ but works only in ideal cases
  3. Compute real space 2-pt correlation function and then get power spectrm
- or $\cdots$

# Band Power Estimation

- How to solve the issue? Different approaches:
  1. Transform data to signal-to-noise basis throwing away contaminated modes (requires at least one $N_p^3$ operation)
  2. Use specific observing symmetries i.e. for azimuthal scanning strategy likelihood computation scales as $N_p^{3/2}$ but works only in ideal cases
  3. Compute real space 2-pt correlation function and then get power spectrm
- or $\cdots$

# Monte Carlo Apodized Spherical Transform EstimatoR - MASTER

- Direct spherical harmonic transform of the observed map
- Easy to include properties specific of a given CMB experiment (survey geometry, scanning strategy, instrumental noise, glitches, spikes, etc)
- Calibrate un-wanted effects by means of Monte Carlo (MC) simulations of modeled observations and data analysis
- mode-mode coupling due to incomplete sky coverage described as a sky window

# Monte Carlo Apodized Spherical Transform EstimatoR - MASTER

- Direct spherical harmonic transform of the observed map
- Easy to include properties specific of a given CMB experiment (survey geometry, scanning strategy, instrumental noise, glitches, spikes, etc)
- Calibrate un-wanted effects by means of Monte Carlo (MC) simulations of modeled observations and data analysis
- mode-mode coupling due to incomplete sky coverage described as a sky window

# Monte Carlo Apodized Spherical Transform EstimatoR - MASTER

- Direct spherical harmonic transform of the observed map
- Easy to include properties specific of a given CMB experiment (survey geometry, scanning strategy, instrumental noise, glitches, spikes, etc)
- Calibrate un-wanted effects by means of Monte Carlo (MC) simulations of modeled observations and data analysis
- mode-mode coupling due to incomplete sky coverage described as a sky window

# Monte Carlo Apodized Spherical Transform EstimatoR - MASTER

- Direct spherical harmonic transform of the observed map
- Easy to include properties specific of a given CMB experiment (survey geometry, scanning strategy, instrumental noise, glitches, spikes, etc)
- Calibrate un-wanted effects by means of Monte Carlo (MC) simulations of modeled observations and data analysis
- mode-mode coupling due to incomplete sky coverage described as a sky window

# MASTER

- Given a map $m_i$ decompose in spherical harmonics to get

$$a_{\ell m} = \int d\hat{n}\, m(\hat{n})\, Y^{\star}_{\ell m}(\hat{n})$$

- Without instrumental noise an un-biased estimator of $C_\ell^{\mathrm{th}}$ is

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

- For incomplete sky coverage we weight pixels with $W(\hat{n})$ and

$$\begin{aligned} \tilde{a}_{\ell m} &= \int d\hat{n}\, m(\hat{n}) W(\hat{n}) Y^{\star}_{\ell m}(\hat{n}) \\ &\approx \Omega_p \sum_p m(p) W(p) Y^{\star}_{\ell m}(p) \end{aligned}$$

# MASTER

- Given a map $m_i$ decompose in spherical harmonics to get

$$a_{\ell m} = \int d\hat{n}\, m(\hat{n})\, Y^{\star}_{\ell m}(\hat{n})$$

- Without instrumental noise an un-biased estimator of $C_\ell^{\mathrm{th}}$ is

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

- For incomplete sky coverage we weight pixels with $W(\hat{n})$ and

$$\begin{aligned} \tilde{a}_{\ell m} &= \int d\hat{n}\, m(\hat{n})\, W(\hat{n})\, Y^{\star}_{\ell m}(\hat{n}) \\ &\approx \Omega_p \sum_p m(p)\, W(p)\, Y^{\star}_{\ell m}(p) \end{aligned}$$

# MASTER

- Given a map $m_i$ decompose in spherical harmonics to get

$$a_{\ell m} = \int d\hat{n}\, m(\hat{n})\, Y^{\star}_{\ell m}(\hat{n})$$

- Without instrumental noise an un-biased estimator of $C_\ell^{\mathrm{th}}$ is

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

- For incomplete sky coverage we weight pixels with $W(\hat{n})$ and

$$\begin{aligned} \tilde{a}_{\ell m} &= \int d\hat{n}\, m(\hat{n}) W(\hat{n})\, Y^{\star}_{\ell m}(\hat{n}) \\ &\approx \Omega_p \sum_p m(p) W(p)\, Y^{\star}_{\ell m}(p) \end{aligned}$$

# MASTER

- The pseudo-power spectrum $\tilde{C}_\ell$ is then defined as

$$\tilde{C}_\ell = \frac{1}{2\ell + 1} \sum_{m=-\ell}^{\ell} |\tilde{a}_{\ell m}|^2$$

- computing $\tilde{a}_{\ell m}$ with a suitable pixelization (HEALPix) scales as $N_p^{1/2} \ell_{\max}$
- $\tilde{C}_\ell$ differs from the full-sky power spectrum but their ensemble average are related

$$\langle \tilde{C}_\ell \rangle = \sum_{\ell'} M_{\ell\ell'} \langle C_{\ell'} \rangle$$

where $M_{\ell\ell'}$ accounts for mode-mode coupling due to incomplete sky

# MASTER

- The pseudo-power spectrum $\tilde{C}_\ell$ is then defined as

$$\tilde{C}_\ell = \frac{1}{2\ell+1}\sum_{m=-\ell}^{\ell} |\tilde{a}_{\ell m}|^2$$

- computing $\tilde{a}_{\ell m}$ with a suitable pixelization (HEALPix) scales as $N_p^{1/2}\ell_{\max}$
- $\tilde{C}_\ell$ differs from the full-sky power spectrum but their ensemble average are related

$$\langle \tilde{C}_\ell \rangle = \sum_{\ell'} M_{\ell\ell'} \langle C_{\ell'} \rangle$$

where $M_{\ell\ell'}$ accounts for mode-mode coupling due to incomplete sky

# MASTER

- The pseudo-power spectrum $\tilde{C}_\ell$ is then defined as

$$\tilde{C}_\ell = \frac{1}{2\ell+1} \sum_{m=-\ell}^{\ell} |\tilde{a}_{\ell m}|^2$$

- computing $\tilde{a}_{\ell m}$ with a suitable pixelization (HEALPix) scales as $N_p^{1/2} \ell_{\mathrm{max}}$
- $\tilde{C}_\ell$ differs from the full-sky power spectrum but their ensemble average are related

$$\langle \tilde{C}_\ell \rangle = \sum_{\ell'} M_{\ell\ell'} \langle C_{\ell'} \rangle$$

where $M_{\ell\ell'}$ accounts for mode-mode coupling due to incomplete sky

# MASTER

- Beam, instrumental noise and data processing of TOD can be included in the power spectrum computation

$$\langle \tilde{C}_\ell \rangle = \sum_{\ell'} M_{\ell\ell'} F_{\ell'} B_{\ell'}^2 \langle C_{\ell'} \rangle + \langle \tilde{N}_\ell \rangle$$

where $B_\ell^2$ is the Window Function describing both beam and pixel smoothing, $\langle \tilde{N}_\ell \rangle$ is the averaged noise power spectrum and $F_\ell$ is the transfer function that models data analysis (filtering, map-making, etc.)

- the rms of our estimates are given by

$$\Delta C_\ell \approx \left[ C_\ell + \frac{N_\ell}{B_\ell^2} \right] \sqrt{\frac{2}{(2\ell+1)\Delta\ell f_{\rm sky}}}$$

# MASTER

- Beam, instrumental noise and data processing of TOD can be included in the power spectrum computation

$$\langle \tilde{C}_\ell \rangle = \sum_{\ell'} M_{\ell\ell'} F_{\ell'} B^2_{\ell'} \langle C_{\ell'} \rangle + \langle \tilde{N}_\ell \rangle$$

  where $B^2_\ell$ is the Window Function describing both beam and pixel smoothing, $\langle \tilde{N}_\ell \rangle$ is the averaged noise power spectrum and $F_\ell$ is the transfer function that models data analysis (filtering, map-making, etc.)
- the rms of our estimates are given by

$$\Delta C_\ell \approx \left[ C_\ell + \frac{N_\ell}{B^2_\ell} \right] \sqrt{\frac{2}{(2\ell + 1)\Delta\ell f_{\rm sky}}}$$

# MASTER

- Mode-mode kernel $M_{\ell\ell'}$ is purely geometric and needs to be computed only once
- Filter function $F_\ell$ is estimated via signal-only MC with known input theoretical model and inverting the main equation
- Number of MC depends on the required accuracy and scales as $\sqrt{N_{MC}^{(s)}}$
- From real data extract $N_{tt'}$ and its frequency power spectrum $\longrightarrow$ construct noise only Gaussian TOD and re-do the actual analysis with noise-only simulations to get $\langle N_\ell \rangle$

# MASTER

- Mode-mode kernel $M_{\ell\ell'}$ is purely geometric and needs to be computed only once
- Filter function $F_\ell$ is estimated via signal-only MC with known input theoretical model and inverting the main equation
- Number of MC depends on the required accuracy and scales as $\sqrt{N_{MC}^{(s)}}$
- From real data extract $N_{tt'}$ and its frequency power spectrum $\rightarrow$ construct noise only Gaussian TOD and re-do the actual analysis with noise-only simulations to get $\langle N_\ell \rangle$

# MASTER

- Mode-mode kernel $M_{\ell\ell'}$ is purely geometric and needs to be computed only once
- Filter function $F_\ell$ is estimated via signal-only MC with known input theoretical model and inverting the main equation
- Number of MC depends on the required accuracy and scales as $\sqrt{N_{MC}^{(s)}}$
- From real data extract $N_{tt'}$ and its frequency power spectrum $\rightarrow$ construct noise only Gaussian TOD and re-do the actual analysis with noise-only simulations to get $\langle N_\ell \rangle$

# MASTER

- Mode-mode kernel $M_{\ell\ell'}$ is purely geometric and needs to be computed only once
- Filter function $F_\ell$ is estimated via signal-only MC with known input theoretical model and inverting the main equation
- Number of MC depends on the required accuracy and scales as $\sqrt{N_{MC}^{(s)}}$
- From real data extract $N_{tt'}$ and its frequency power spectrum $\rightarrow$ construct noise only Gaussian TOD and re-do the actual analysis with noise-only simulations to get $\langle N_\ell \rangle$

# MASTER

- Reduce correlation between $C_\ell$s due to sky-cut and errors on single $C_\ell$ with binning together data in $\ell$ space: create binning, $P_{b\ell}$, and un-binning, $Q_{\ell b}$, operators: $\mathcal{C}_b = P_{b\ell} C_\ell$
- Main equation becomes

$$\langle \tilde{C}_\ell \rangle = K_{\ell\ell'} \langle C_{\ell'} \rangle + \langle \tilde{N}_\ell \rangle$$

and we want a solution in the form of

$$P_{b\ell} K_{\ell\ell'} \langle C_{\ell'} \rangle = P_{b\ell} (\langle \tilde{C}_\ell \rangle - \langle \tilde{N}_\ell \rangle)$$

# MASTER

- Reduce correlation between $C_\ell$s due to sky-cut and errors on single $C_\ell$ with binning together data in $\ell$ space: create binning, $P_{b\ell}$, and un-binning, $Q_{\ell b}$, operators: $\mathcal{C}_b = P_{b\ell} C_\ell$
- Main equation becomes

$$\langle \tilde{C}_\ell \rangle = K_{\ell\ell'} \langle C_{\ell'} \rangle + \langle \tilde{N}_\ell \rangle$$

and we want a solution in the form of

$$P_{b\ell} K_{\ell\ell'} \langle C_{\ell'} \rangle = P_{b\ell} (\langle \tilde{C}_\ell \rangle - \langle \tilde{N}_\ell \rangle)$$

# MASTER

- Working a bit with the algebra we get

$$\langle \mathcal{C}_b \rangle = K^{-1}_{bb'} P_{b'\ell'} (\langle \tilde{C}_{\ell'} \rangle - \langle \tilde{N}_{\ell'} \rangle)$$

  where $K_{bb'} = P_{b\ell} M_{\ell\ell'} F_{\ell'} B^2_{\ell'} Q_{\ell' b'}$

- Best un-biased estimator of the whole sky noise and signal power spectrum is

$$\begin{aligned} \hat{\mathcal{C}}_b &= K^{-1}_{bb'} P_{b'\ell} (\tilde{C}_\ell - \langle \tilde{N}_\ell \rangle_{MC}) \\ \hat{\mathcal{N}}_b &= K^{-1}_{bb'} P_{b'\ell} \langle \tilde{N}_\ell \rangle_{MC} \end{aligned}$$

# MASTER

- Working a bit with the algebra we get

$$\langle \mathcal{C}_b \rangle = K^{-1}_{bb'} P_{b'\ell'} (\langle \tilde{C}_{\ell'} \rangle - \langle \tilde{N}_{\ell'} \rangle)$$

  where $K_{bb'} = P_{b\ell} M_{\ell\ell'} F_{\ell'} B^2_{\ell'} Q_{\ell' b'}$

- Best un-biased estimator of the whole sky noise and signal power spectrum is

$$\begin{aligned} \hat{\mathcal{C}}_b &= K^{-1}_{bb'} P_{b'\ell} (\tilde{C}_\ell - \langle \tilde{N}_\ell \rangle_{MC}) \\ \hat{\mathcal{N}}_b &= K^{-1}_{bb'} P_{b'\ell} \langle \tilde{N}_\ell \rangle_{MC} \end{aligned}$$

# MASTER

- Finally to get the covariance of $\hat{\mathcal{C}}_b$ useful for extracting cosmological parameters we
  1. smooth interpolate $\hat{\mathcal{C}}_b$ to get a fiducial CMB model
  2. perform a set of signal+noise MC simulations to get $\{\hat{\mathcal{C}}_b\}$
  3. the covariance is given by

$$\mathbf{C}_{bb'} = \langle(\hat{\mathcal{C}}_b - \langle\hat{\mathcal{C}}_b\rangle_{MC})(\hat{\mathcal{C}}_{b'} - \langle\hat{\mathcal{C}}_{b'}\rangle_{MC})\rangle_{MC}$$

  4. the error bar is $\Delta\hat{\mathcal{C}}_b = \mathbf{C}_{bb}^{1/2}$

# MASTER

- Finally to get the covariance of $\hat{\mathcal{C}}_b$ useful for extracting cosmological parameters we
  1. smooth interpolate $\hat{\mathcal{C}}_b$ to get a fiducial CMB model
  2. perform a set of signal+noise MC simulations to get $\{\hat{\mathcal{C}}_b\}$
  3. the covariance is given by

$$\mathbf{C}_{bb'} = \langle(\hat{\mathcal{C}}_b - \langle\hat{\mathcal{C}}_b\rangle_{MC})(\hat{\mathcal{C}}_{b'} - \langle\hat{\mathcal{C}}_{b'}\rangle_{MC})\rangle_{MC}$$

  4. the error bar is $\Delta\hat{\mathcal{C}}_b = \mathbf{C}_{bb}^{1/2}$

# MASTER

- Finally to get the covariance of $\hat{\mathcal{C}}_b$ useful for extracting cosmological parameters we
  1. smooth interpolate $\hat{\mathcal{C}}_b$ to get a fiducial CMB model
  2. perform a set of signal+noise MC simulations to get $\{\hat{\mathcal{C}}_b\}$
  3. the covariance is given by

$$\mathbf{C}_{bb'} = \langle (\hat{\mathcal{C}}_b - \langle \hat{\mathcal{C}}_b \rangle_{MC})(\hat{\mathcal{C}}_{b'} - \langle \hat{\mathcal{C}}_{b'} \rangle_{MC}) \rangle_{MC}$$

  4. the error bar is $\Delta\hat{\mathcal{C}}_b = \mathbf{C}_{bb}^{1/2}$

# MASTER

- Finally to get the covariance of $\hat{\mathcal{C}}_b$ useful for extracting cosmological parameters we
  1. smooth interpolate $\hat{\mathcal{C}}_b$ to get a fiducial CMB model
  2. perform a set of signal+noise MC simulations to get $\{\hat{\mathcal{C}}_b\}$
  3. the covariance is given by

$$\mathbf{C}_{bb'} = \langle(\hat{\mathcal{C}}_b - \langle\hat{\mathcal{C}}_b\rangle_{MC})(\hat{\mathcal{C}}_{b'} - \langle\hat{\mathcal{C}}_{b'}\rangle_{MC})\rangle_{MC}$$

  4. the error bar is $\Delta\hat{\mathcal{C}}_b = \mathbf{C}_{bb}^{1/2}$

# MASTER

- Finally to get the covariance of $\hat{\mathcal{C}}_b$ useful for extracting cosmological parameters we
  1. smooth interpolate $\hat{\mathcal{C}}_b$ to get a fiducial CMB model
  2. perform a set of signal+noise MC simulations to get $\{\hat{\mathcal{C}}_b\}$
  3. the covariance is given by

$$\mathbf{C}_{bb'} = \langle(\hat{\mathcal{C}}_b - \langle\hat{\mathcal{C}}_b\rangle_{MC})(\hat{\mathcal{C}}_{b'} - \langle\hat{\mathcal{C}}_{b'}\rangle_{MC})\rangle_{MC}$$

  4. the error bar is $\Delta\hat{\mathcal{C}}_b = \mathbf{C}_{bb}^{1/2}$

# MASTER

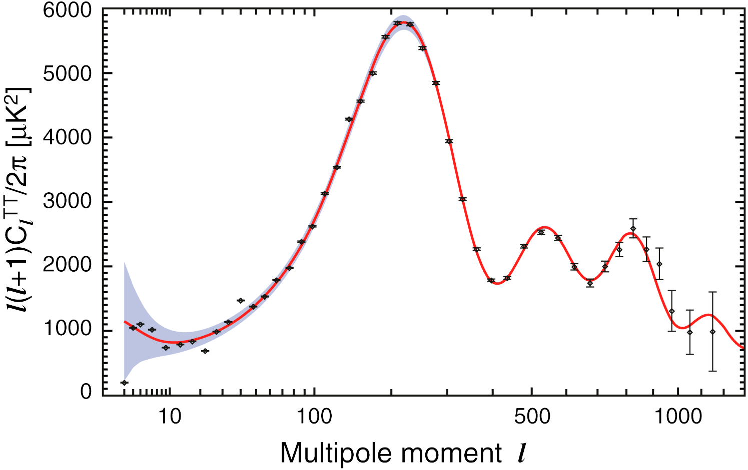



**Figure:** *WMAP*-7yr Temperature angular power spectrum

# Outline

From TOD to maps

The Likelihood function

Application to CMB experiments

CMB Power Spectrum extraction

MASTER approach

**Extracting and Forecasting cosmological parameters**

# Cosmological Parameter Estimation

- Once we have band power $\hat{\mathcal{C}}_b$ and their covariance $\mathbf{C}_{bb'}$ we can compare them to a given theory
- The likelihood of the problem is

$$\mathcal{L}(\hat{\mathcal{C}}_b) = \frac{1}{(2\pi)^{N_b/2}\sqrt{|\mathbf{C}|}} \exp\left[-\frac{1}{2}(\hat{\mathcal{C}}_b - \mathcal{C}_b^{\rm th})\mathbf{C}_{bb'}^{-1}(\hat{\mathcal{C}}_{b'} - \mathcal{C}_{b'}^{\rm th})\right]$$

  and evaluate this at many points in the parameter space
- Use a fast Boltzmann-Einstein solver like `CMBfast` or `CAMB` it is possible to evaluate $\mathcal{C}_b^{\rm th}$ as a function of $\mathbf{p} = \{\Omega_b, \Omega_0, \Omega_{\rm CDM}, H_0, n_s, \tau, r, \cdots\}$ and evaluate the likelihood.

# Cosmological Parameter Estimation

- Once we have band power $\hat{\mathcal{C}}_b$ and their covariance $\mathbf{C}_{bb'}$ we can compare them to a given theory
- The likelihood of the problem is

$$\mathcal{L}(\hat{\mathcal{C}}_b) = \frac{1}{(2\pi)^{N_b/2}\sqrt{|\mathbf{C}|}} \exp\left[-\frac{1}{2}(\hat{\mathcal{C}}_b - \mathcal{C}_b^{\mathrm{th}})\mathbf{C}_{bb'}^{-1}(\hat{\mathcal{C}}_{b'} - \mathcal{C}_{b'}^{\mathrm{th}})\right]$$

  and evaluate this at many points in the parameter space
- Use a fast Boltzmann-Einstein solver like `CMBfast` or `CAMB` it is possible to evaluate $\mathcal{C}_b^{\mathrm{th}}$ as a function of $\mathbf{p} = \{\Omega_b, \Omega_0, \Omega_{\mathrm{CDM}}, H_0, n_s, \tau, r, \cdots\}$ and evaluate the likelihood.

# Cosmological Parameter Estimation

- Once we have band power $\hat{\mathcal{C}}_b$ and their covariance $\mathbf{C}_{bb'}$ we can compare them to a given theory
- The likelihood of the problem is

$$\mathcal{L}(\hat{\mathcal{C}}_b) = \frac{1}{(2\pi)^{N_b/2}\sqrt{|\mathbf{C}|}} \exp\left[-\frac{1}{2}(\hat{\mathcal{C}}_b - \mathcal{C}_b^{\mathrm{th}})\mathbf{C}_{bb'}^{-1}(\hat{\mathcal{C}}_{b'} - \mathcal{C}_{b'}^{\mathrm{th}})\right]$$

  and evaluate this at many points in the parameter space
- Use a fast Boltzmann-Einstein solver like `CMBfast` or `CAMB` it is possible to evaluate $\mathcal{C}_b^{\mathrm{th}}$ as a function of $\mathbf{p} = \{\Omega_b, \Omega_0, \Omega_{\mathrm{CDM}}, H_0, n_s, \tau, r, \cdots\}$ and evaluate the likelihood.

# Cosmological Parameter Estimation

- Due to high precision of current CMB experiments, this brute-force approach is no more possible: 10 parameters samples at 10 points each $\rightarrow 10^{10}$ runs of `CMBfast`
- Other approaches:
  - knowing the physics of CMB interpolate on a non-regular grid of parameters (slowly and fast varying parameters)
  - sample the likelihood via Monte Carlo methods

# Cosmological Parameter Estimation

- Due to high precision of current CMB experiments, this brute-force approach is no more possible: 10 parameters samples at 10 points each $\rightarrow 10^{10}$ runs of `CMBfast`
- Other approaches:
  - knowing the physics of CMB interpolate on a non-regular grid of parameters (slowly and fast varying parameters)
  - sample the likelihood via Monte Carlo methods

# Cosmological Parameter Estimation

- Due to high precision of current CMB experiments, this brute-force approach is no more possible: 10 parameters samples at 10 points each $\rightarrow 10^{10}$ runs of `CMBfast`
- Other approaches:
  - knowing the physics of CMB interpolate on a non-regular grid of parameters (slowly and fast varying parameters)
  - sample the likelihood via Monte Carlo methods

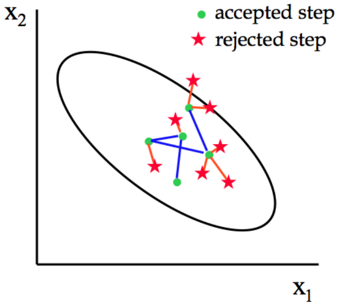
# Cosmological Parameter Estimation

- Due to high precision of current CMB experiments, this brute-force approach is no more possible: 10 parameters samples at 10 points each $\rightarrow 10^{10}$ runs of `CMBfast`
- Other approaches:
  - knowing the physics of CMB interpolate on a non-regular grid of parameters (slowly and fast varying parameters)
  - sample the likelihood via Monte Carlo methods

# Markov Chain Monte Carlo

Generates sequence of random samples from an arbitrary probability density function

- Metropolis algorithm
  - assume a gaussian distribution for parameter steps
  - accept of reject the step
  - simple and generally applicable



# Metropolis Algorithm

- select initial parameter vector $\mathbf{p}_0$
- iterate as follows: at iteration number $k$
  1. create new position $\mathbf{p}^* = \mathbf{p}_k + \Delta\mathbf{p}$ where $\Delta\mathbf{p}$ is randomly chosen from a Gaussian distribution
  2. evaluate $r = \mathcal{L}(\mathbf{p}^*)/\mathcal{L}(\mathbf{p}_k)$
  3. accept position i.e. set $\mathbf{p}_{k+1} = \mathbf{p}^*$ if $r \geq 1$, otherwise set $\mathbf{p}_{k+1} = \mathbf{p}_k$
- requires only computation of $\mathcal{L}$
- creates a Markov chain since $\mathbf{p}_{k+1}$ depends only on $\mathbf{p}_k$.

# Metropolis Algorithm

- select initial parameter vector $\mathbf{p}_0$
- iterate as follows: at iteration number $k$
  1. create new position $\mathbf{p}^\star = \mathbf{p}_k + \Delta\mathbf{p}$ where $\Delta\mathbf{p}$ is randomly chosen from a Gaussian distribution
  2. evaluate $r = \mathcal{L}(\mathbf{p}^\star)/\mathcal{L}(\mathbf{p}_k)$
  3. accept position i.e. set $\mathbf{p}_{k+1} = \mathbf{p}^\star$ if $r \geq 1$ otherwise set $\mathbf{p}_{k+1} = \mathbf{p}_k$
- requires only computation of $\mathcal{L}$
- creates a Markov chain since $\mathbf{p}_{k+1}$ depends only on $\mathbf{p}_k$.

# Metropolis Algorithm

- select initial parameter vector $\mathbf{p}_0$
- iterate as follows: at iteration number $k$
  1. create new position $\mathbf{p}^\star = \mathbf{p}_k + \Delta\mathbf{p}$ where $\Delta\mathbf{p}$ is randomly chosen from a Gaussian distribution
  2. evaluate $r = \mathcal{L}(\mathbf{p}^\star)/\mathcal{L}(\mathbf{p}_k)$
  3. accept position i.e. set $\mathbf{p}_{k+1} = \mathbf{p}^\star$ if $r \geq 1$ otherwise set $\mathbf{p}_{k+1} = \mathbf{p}_k$
- requires only computation of $\mathcal{L}$
- creates a Markov chain since $\mathbf{p}_{k+1}$ depends only on $\mathbf{p}_k$.

# Metropolis Algorithm

- select initial parameter vector $\mathbf{p}_0$
- iterate as follows: at iteration number $k$
  1. create new position $\mathbf{p}^\star = \mathbf{p}_k + \Delta\mathbf{p}$ where $\Delta\mathbf{p}$ is randomly chosen from a Gaussian distribution
  2. evaluate $r = \mathcal{L}(\mathbf{p}^\star)/\mathcal{L}(\mathbf{p}_k)$
  3. accept position i.e. set $\mathbf{p}_{k+1} = \mathbf{p}^\star$ if $r \geq 1$ otherwise set $\mathbf{p}_{k+1} = \mathbf{p}_k$
- requires only computation of $\mathcal{L}$
- creates a Markov chain since $\mathbf{p}_{k+1}$ depends only on $\mathbf{p}_k$.

# Metropolis Algorithm

- select initial parameter vector $\mathbf{p}_0$
- iterate as follows: at iteration number $k$
  1. create new position $\mathbf{p}^\star = \mathbf{p}_k + \Delta\mathbf{p}$ where $\Delta\mathbf{p}$ is randomly chosen from a Gaussian distribution
  2. evaluate $r = \mathcal{L}(\mathbf{p}^\star)/\mathcal{L}(\mathbf{p}_k)$
  3. accept position i.e. set $\mathbf{p}_{k+1} = \mathbf{p}^\star$ if $r \geq 1$ otherwise set $\mathbf{p}_{k+1} = \mathbf{p}_k$
- requires only computation of $\mathcal{L}$
- creates a Markov chain since $\mathbf{p}_{k+1}$ depends only on $\mathbf{p}_k$.

# Metropolis Algorithm

- select initial parameter vector $\mathbf{p}_0$
- iterate as follows: at iteration number $k$
  1. create new position $\mathbf{p}^\star = \mathbf{p}_k + \Delta\mathbf{p}$ where $\Delta\mathbf{p}$ is randomly chosen from a Gaussian distribution
  2. evaluate $r = \mathcal{L}(\mathbf{p}^\star)/\mathcal{L}(\mathbf{p}_k)$
  3. accept position i.e. set $\mathbf{p}_{k+1} = \mathbf{p}^\star$ if $r \geq 1$ otherwise set $\mathbf{p}_{k+1} = \mathbf{p}_k$
- requires only computation of $\mathcal{L}$
- creates a Markov chain since $\mathbf{p}_{k+1}$ depends only on $\mathbf{p}_k$.

# Metropolis Algorithm

- select initial parameter vector $\mathbf{p}_0$
- iterate as follows: at iteration number $k$
  1. create new position $\mathbf{p}^\star = \mathbf{p}_k + \Delta\mathbf{p}$ where $\Delta\mathbf{p}$ is randomly chosen from a Gaussian distribution
  2. evaluate $r = \mathcal{L}(\mathbf{p}^\star)/\mathcal{L}(\mathbf{p}_k)$
  3. accept position i.e. set $\mathbf{p}_{k+1} = \mathbf{p}^\star$ if $r \geq 1$ otherwise set $\mathbf{p}_{k+1} = \mathbf{p}_k$
- requires only computation of $\mathcal{L}$
- creates a Markov chain since $\mathbf{p}_{k+1}$ depends only on $\mathbf{p}_k$.

# Forecasting

- Having the covariance matrix we can construct the Fisher matrix as

$$F_{ij} = \sum_{\ell} \frac{\partial C_\ell}{\partial p_i} C^{-1} \frac{\partial C_\ell}{\partial p_j}$$

- Fisher matrix formalism allows to predict how accurately a given experiment (encoded into the covariance matrix) would constrain parameters assuming to know them so we can compute derivatives around their best-fit values

# Forecasting

- Having the covariance matrix we can construct the Fisher matrix as

$$F_{ij} = \sum_{\ell} \frac{\partial C_\ell}{\partial p_i} C^{-1} \frac{\partial C_\ell}{\partial p_j}$$

- Fisher matrix formalism allows to predict how accurately a given experiment (encoded into the covariance matrix) would constrain parameters assuming to know them so we can compute derivatives around their best-fit values

# Forecasting

- $1-\sigma$ limit on $p_i$ is $1/\sqrt{F_{ii}}$. If more than one parameter is allowed to vary at the same time the error is instead $\sqrt{F_{ii}^{-1}}$
- the join probability is given

$$P(p_i, p_j) \propto \exp\left(-\frac{1}{2} p_i F_{ij} p_j\right)$$

- allowing $p_j$ to vary is equivalent to integrate the probability over all possible values for $p_j$: this is called *m*arginalization over $p_j$

$$P(p_i) = \int dp_j \, P(p_i, p_j) \propto \exp\left\{-\frac{1}{2}\left(\frac{F_{ii}F_{jj} - F_{ij}F_{ji}}{F_{jj}}\right)\right\}$$

where the exponential is exactly $1/F_{ii}^{-1}$ and the $\sigma = \sqrt{F_{ii}^{-1}}$

# Forecasting

- $1-\sigma$ limit on $p_i$ is $1/\sqrt{F_{ii}}$. If more than one parameter is allowed to vary at the same time the error is instead $\sqrt{F^{-1}_{ii}}$
- the join probability is given

$$P(p_i, p_j) \propto \exp\left(-\frac{1}{2} p_i F_{ij} p_j\right)$$

- allowing $p_j$ to vary is equivalent to integrate the probability over all possible values for $p_j$: this is called *m*arginalization over $p_j$

$$P(p_i) = \int dp_j \, P(p_i, p_j) \propto \exp\left\{-\frac{1}{2}\left(\frac{F_{ii}F_{jj} - F_{ij}F_{ji}}{F_{jj}}\right)\right\}$$

where the exponential is exactly $1/F^{-1}_{ii}$ and the $\sigma = \sqrt{F^{-1}_{ii}}$

# Forecasting

- $1-\sigma$ limit on $p_i$ is $1/\sqrt{F_{ii}}$. If more than one parameter is allowed to vary at the same time the error is instead $\sqrt{F^{-1}_{ii}}$
- the join probability is given

$$P(p_i, p_j) \propto \exp\left(-\frac{1}{2} p_i F_{ij} p_j\right)$$

- allowing $p_j$ to vary is equivalent to integrate the probability over all possible values for $p_j$: this is called *m*arginalization over $p_j$

$$P(p_i) = \int dp_j\, P(p_i, p_j) \propto \exp\left\{-\frac{1}{2}\left(\frac{F_{ii}F_{jj} - F_{ij}F_{ji}}{F_{jj}}\right)\right\}$$

where the exponential is exactly $1/F^{-1}_{ii}$ and the $\sigma = \sqrt{F^{-1}_{ii}}$