Euclid France Galaxy Clustering

Covariance effects on cosmological parameter inference : non-Gaussianity and sampling noise

Sylvain Gouyou Beauchamps

Philippe Baratta, Julien Bel, Stéphanie Escoffier, William Gillard,
Julian Bautista, Carmelita Carbone

3rd year PhD under the supervision of Stéphanie Escoffier and William Gillard

CPPM, RENOIR

April 20th 2021

Outline

I. Introduction

II. Simulations and covariance

- The DEMNUni-Cov simulations
- The covmos method
- The NERCOME estimator

III. Results from MCMC on the DEMNUni power spectrum

Inference of cosmological parameters θ from data set D

Likelihood → Prior →

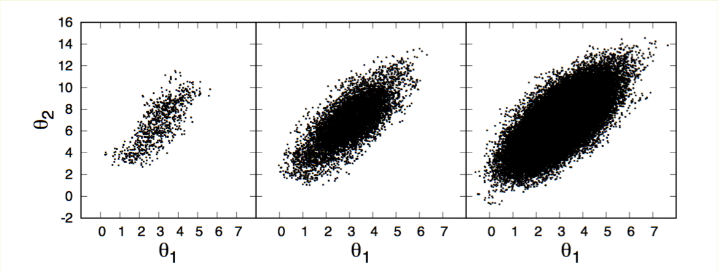
$$P(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{P(\boldsymbol{D}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\boldsymbol{D})}$$

← Normalization

Posterior

Bayes theorem

Sample the posterior through statistical processes such as Monte Carlo Markov Chains



Inference of cosmological parameters θ from data set D

Likelihood → Prior →

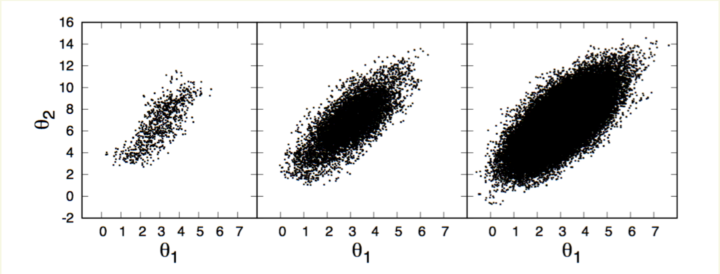
$$P(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{P(\boldsymbol{D}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\boldsymbol{D})}$$

← Normalization

↑ Posterior

Bayes theorem

Sample the posterior through statistical processes such as Monte Carlo Markov Chains



Assumption of a Gaussian Likelihood

Covariance independant of cosmology

$$\chi^2 \equiv -2\ln P(\boldsymbol{\theta}|\boldsymbol{D}) = \left[[\mathbf{D} - \mathbf{m}(\boldsymbol{\theta})]^T \boldsymbol{C}^{-1}[\mathbf{D} - \mathbf{m}(\boldsymbol{\theta})]\right] + \text{const}$$

Data

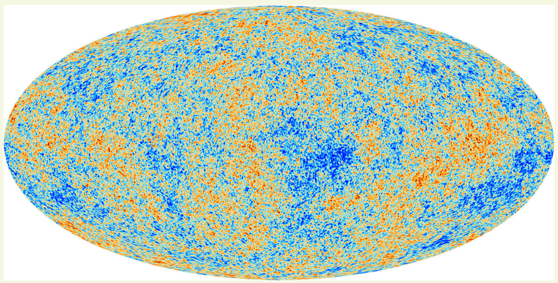
Model

Covariance : Errors and correlations

The covariance is a key ingredient for parameter inference

Covariance of the power spectrum P(k)

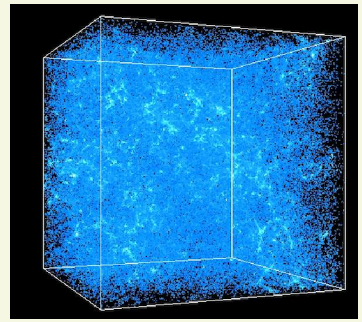
z = 1100



CMB ➔ Gaussian initial conditions

Linear clustering →

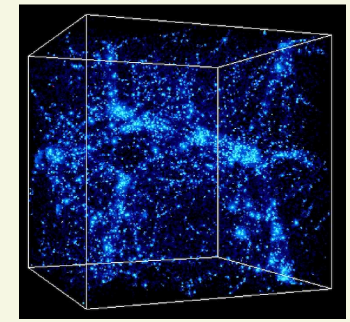
z = 10



Stays Gaussian

Non-linear clustering on small scales →

z = 0



Becomes Non-Gaussian

$k \gtrsim 0.2\ h/\mathrm{Mpc}$

Bin averaged Trispectrum

$$\mathbf{C}(k_i, k_j) = \frac{P^2(k_i)}{N_{k_i}}\delta_{ij} + \bar{T}(k_i, k_j) = \mathbf{C}^{\mathrm{G}}(k_i) + \mathbf{C}^{\mathrm{NG}}(k_i, k_j),$$

Non-Gaussianities are contained in higher order statistics : Difficult to predict analytically

[Wadekar & Scoccimaro 2019]

Estimate the full covariance from simulations

$$\hat{C}_{ij} = \frac{1}{N_m - 1}\left[\sum_{n}^{N_m}[P^{(n)}(k_i) - \bar{P}(k_i)][P^{(n)}(k_j) - \bar{P}(k_j)]\right]$$

Sensitive to the number of simulations and bins N_m and N_k

Estimate the full covariance from simulations

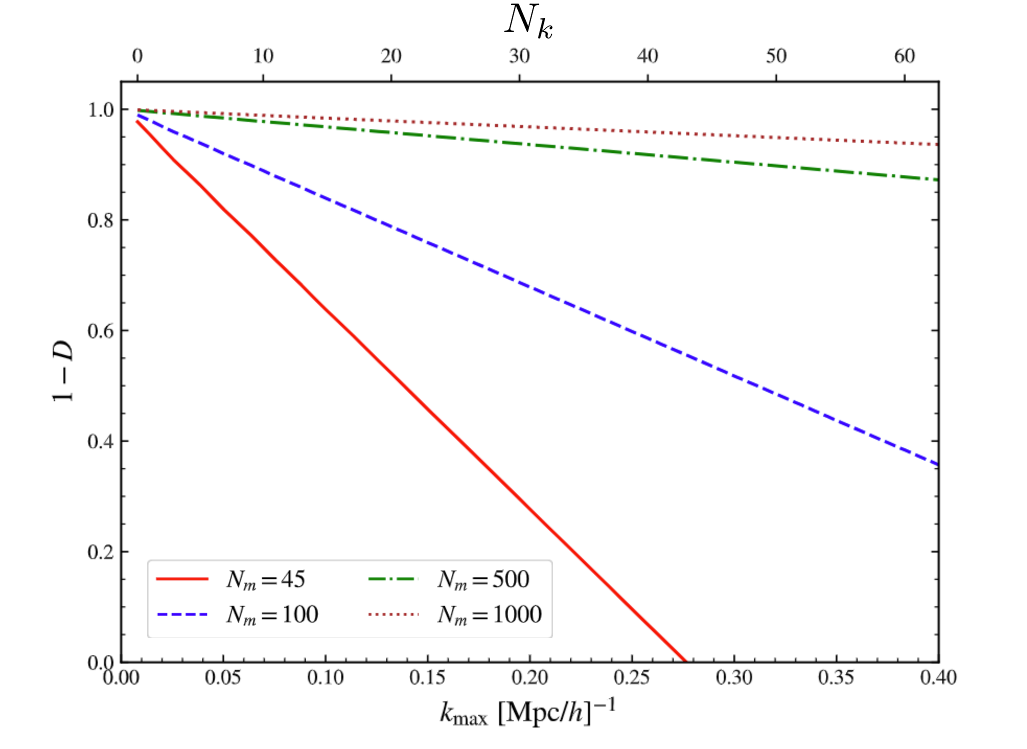
$$\hat{C}_{ij} = \frac{1}{N_m - 1}\left[\sum_{n}^{N_m}[P^{(n)}(k_i) - \bar{P}(k_i)][P^{(n)}(k_j) - \bar{P}(k_j)]\right]$$

Sensitive to the number of simulations and bins N_m and N_k

- Hartlap bias in the precision matrix $\boldsymbol{\Psi}$ [Wishart 1928, Hartlap 2007]

$$\hat{\boldsymbol{\Psi}} = \hat{\mathbf{C}}^{-1}$$

Biased estimator of $\boldsymbol{\Psi}$: $E[\hat{\boldsymbol{\Psi}}] = \frac{1}{1-D}\boldsymbol{\Psi},\ D = \frac{N_k + 1}{N_m - 1}$



Hartlap factor to multiply to $\boldsymbol{\Psi}$, before the minimization.

First effect of sampling noise :

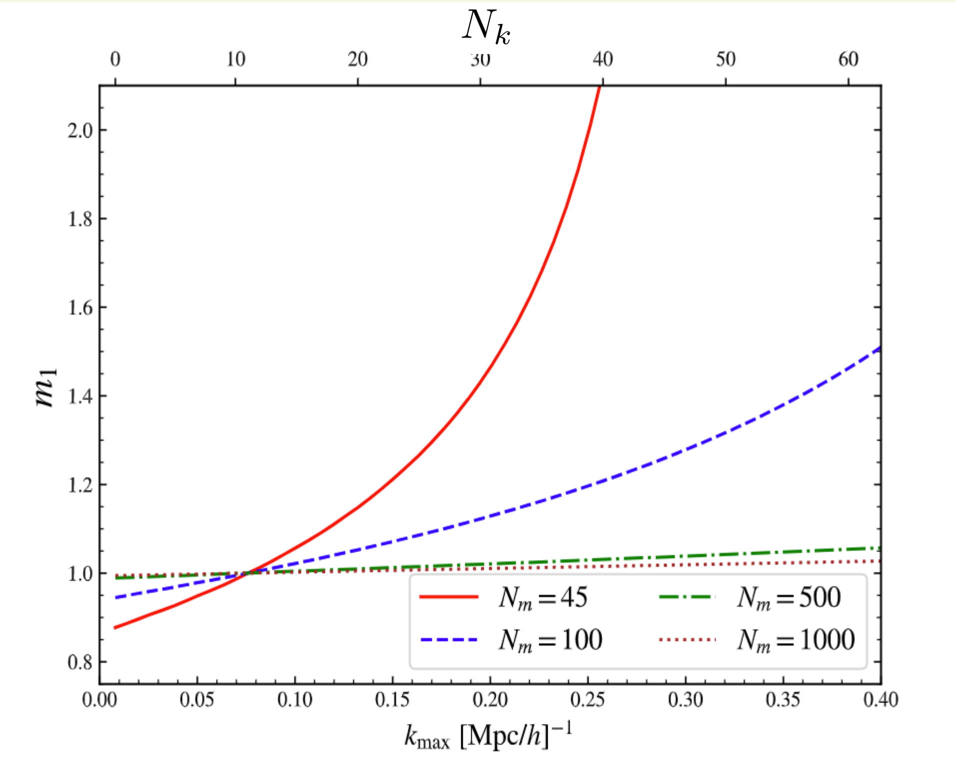
Overestimate $\boldsymbol{\Psi}$ ➔ underestimate parameter's errors

Estimate the full covariance from simulations

$$\hat{C}_{ij} = \frac{1}{N_m - 1}\left[\sum_{n}^{N_m}[P^{(n)}(k_i) - \bar{P}(k_i)][P^{(n)}(k_j) - \bar{P}(k_j)]\right]$$

Sensitive to the number of simulations and bins N_m and N_k

- Sampling noise in the precision matrix propagates to the posteriors : Additional errors on best-fit and errors [Taylor et al. 2013 ; Dodelson & Schneider 2013 ; Percival et al. 2014 ;Taylor & Joachimi 2014]



Correct by inflating the parameter's covariance matrix Φ
[Percival et al. 2014]

$$\hat{\Phi} \rightarrow m_1 \times \hat{\Phi}$$

$$m_1 = \frac{1 + B(N_b - N_p)}{1 + A + B(N_p + 1)}$$

Dispersion on the best-fit

Bias on the estimation of the error

A and B are fonctions of (N_m, N_k)

Estimate the full covariance from simulations

$$\hat{C}_{ij} = \frac{1}{N_m - 1}\left[\sum_{n}^{N_m}[P^{(n)}(k_i) - \bar{P}(k_i)][P^{(n)}(k_j) - \bar{P}(k_j)]\right]$$

Sensitive to the number of simulations and bins N_m and N_k

- Hartlap bias in the precision matrix $\mathbf{\Psi}$ [Hartlap 2007]
- Sampling noise in the precision matrix propagates to the posteriors : Additional errors on best-fit and errors [Taylor et al. 2013 ; Dodelson & Schneider 2013 ; Percival et al. 2014 ;Taylor & Joachimi 2014]

Estimate the full covariance from simulations

$$\hat{C}_{ij} = \frac{1}{N_m - 1}\left[\sum_{n}^{N_m}[P^{(n)}(k_i) - \bar{P}(k_i)][P^{(n)}(k_j) - \bar{P}(k_j)]\right]$$

Sensitive to the number of simulations and bins N_m and N_k

- Hartlap bias in the precision matrix $\mathbf{\Psi}$ [Hartlap 2007]
- Sampling noise in the precision matrix propagates to the posteriors : Additional errors on best-fit and errors [Taylor et al. 2013 ; Dodelson & Schneider 2013 ; Percival et al. 2014 ;Taylor & Joachimi 2014]

To deal with sampling noise we can :

- Create very large sets of mocks : N-body is slow

➔ Approximate method : *covmos* by Philippe Baratta [Baratta et al. 2019, Baratta et al. In prep]

Estimate the full covariance from simulations

$$\hat{C}_{ij} = \frac{1}{N_m - 1}\left[\sum_{n}^{N_m}[P^{(n)}(k_i) - \bar{P}(k_i)][P^{(n)}(k_j) - \bar{P}(k_j)]\right]$$

Sensitive to the number of simulations and bins N_m and N_k

- Hartlap bias in the precision matrix $\mathbf{\Psi}$ [Hartlap 2007]
- Sampling noise in the precision matrix propagates to the posteriors : Additional errors on best-fit and errors [Taylor et al. 2013 ; Dodelson & Schneider 2013 ; Percival et al. 2014 ;Taylor & Joachimi 2014]

To deal with sampling noise we can :

- Create very large sets of mocks : N-body is slow

➔ Approximate method : *covmos* by Philippe Baratta [Baratta et al. 2019, Baratta et al. In prep]

- Use alternative estimators

➔ Non linear shrinkage : *NERCOME* [Joachimi 2016]

Estimate the full covariance from simulations

$$\hat{C}_{ij} = \frac{1}{N_m - 1}\left[\sum_{n}^{N_m}[P^{(n)}(k_i) - \bar{P}(k_i)][P^{(n)}(k_j) - \bar{P}(k_j)]\right]$$

Sensitive to the number of simulations and bins N_m and N_k

- Hartlap bias in the precision matrix Ψ [Hartlap 2007]
- Sampling noise in the precision matrix propagates to the posteriors : Additional errors on best-fit and errors [Taylor et al. 2013 ; Dodelson & Schneider 2013 ; Percival et al. 2014 ;Taylor & Joachimi 2014]

To deal with sampling noise we can :

- Create very large sets of mocks : N-body is slow

➔ Approximate method : *covmos* by Philippe Baratta [Baratta et al. 2019, Baratta et al. in prep.]

- Use alternative estimators

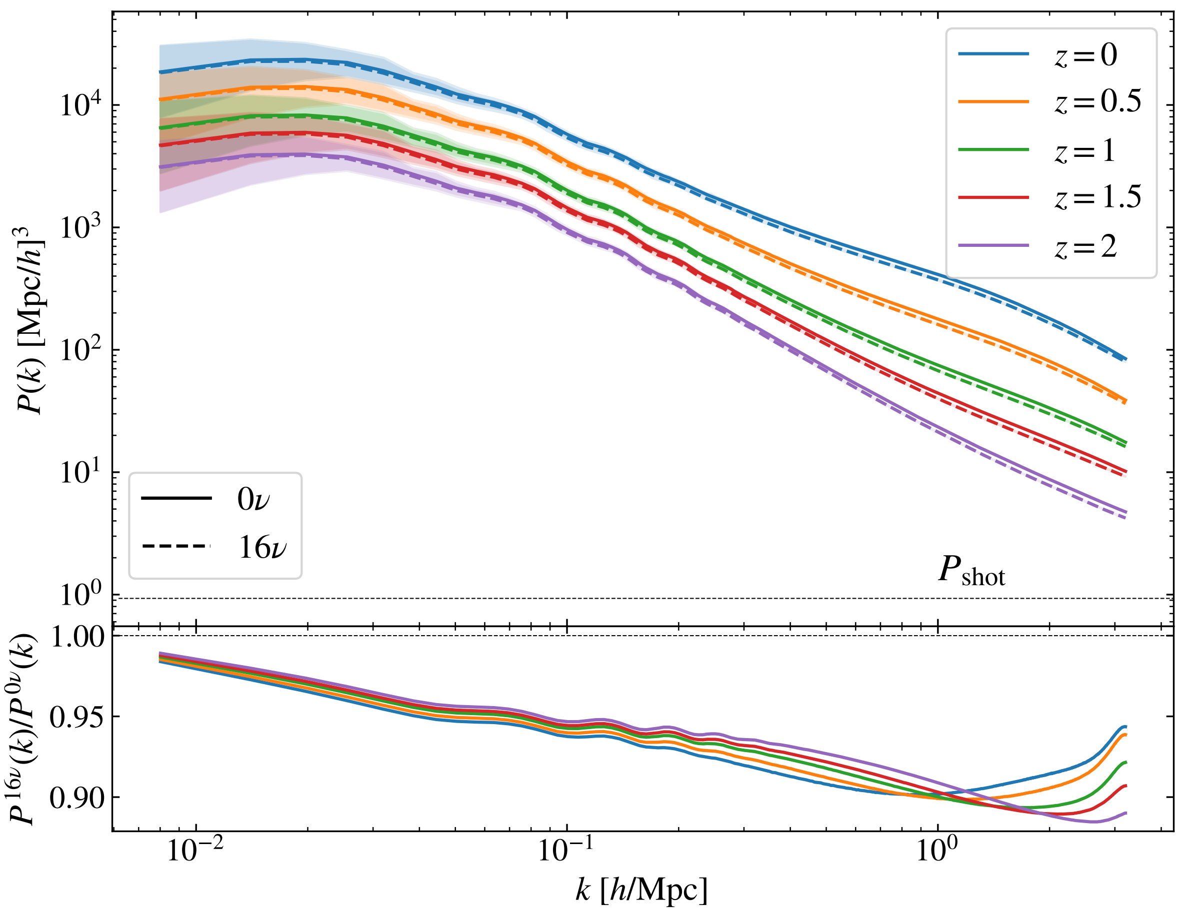
➔ Non linear shrinkage : *NERCOME* [Joachimi 2016]

➡ Test and comparaison of analytic corrections, covmos and NERCOME

The DEMNUni-Cov simulations

50 realizations for each cosmology : 5 as data vector (1 for each redshift) **+ 45 for the covariance**

CDM power spectrum : $P_{cb}(k)$



- Cosmologies : 0ν and 16ν

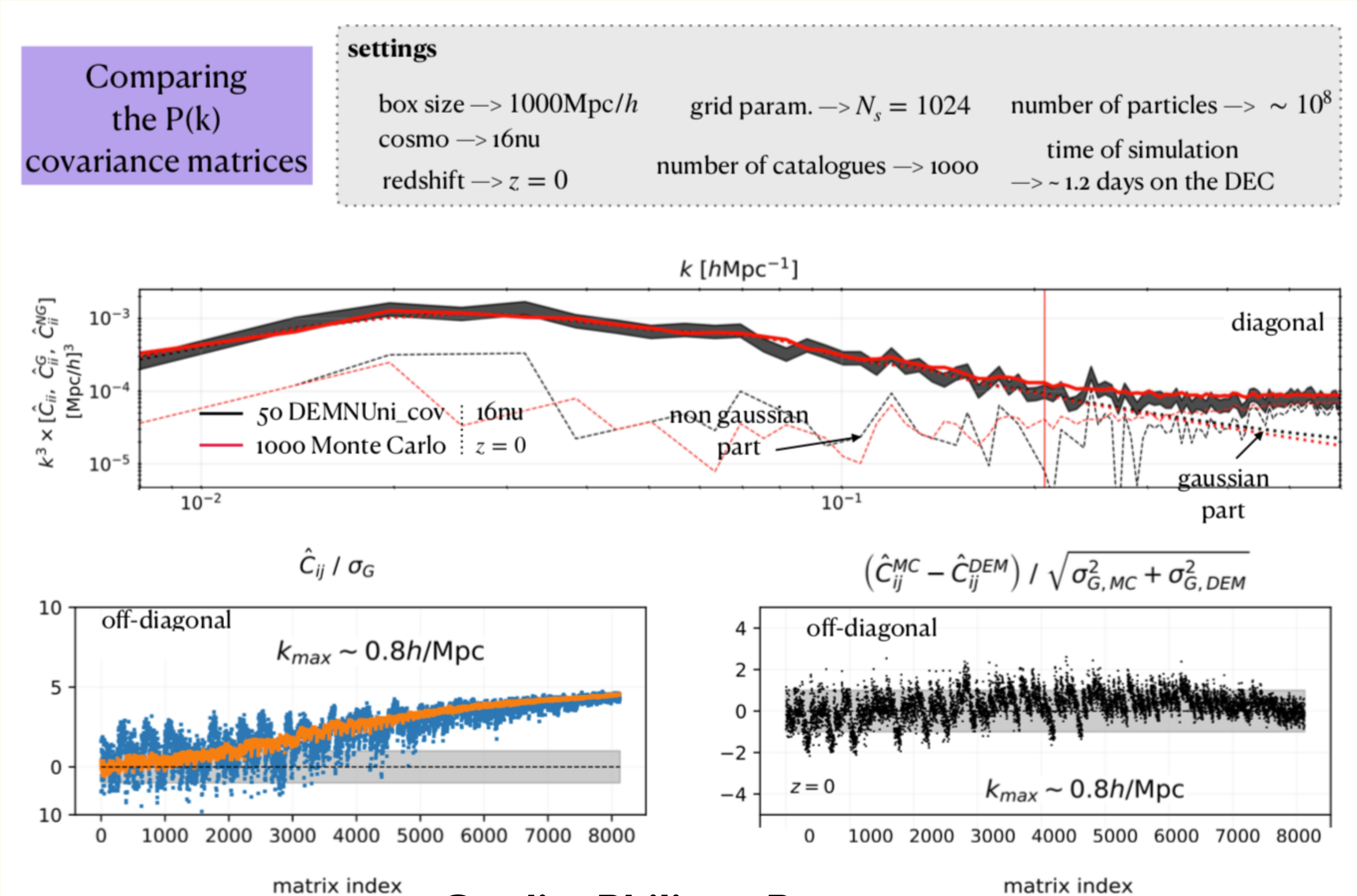
$$\begin{cases} \Omega_{\rm m} = 0.32 \\ \Omega_{\rm b} = 0.05 \\ h = 0.67 \\ n_s = 0.96 \\ A_s = 2.1265 \times 10^9 \\ \mathbf{3m_\nu} = \mathbf{0} \text{ or } \mathbf{0.16} \text{ eV} \\ \mathbf{\Omega_{\rm cdm}} = \mathbf{0.27} \text{ or } \mathbf{0.2663} \end{cases}$$

- V = 1 Gpc/h^3
- $N_p = 1024^3$ (+ 1024^3)

Managed by Carmelita Carbone in Milan

Fast non-Gaussian mock generation with *covmos*

Cloning the DEMNUni-Cov covariance matrix by targeting the P(k) and PDF(δ)

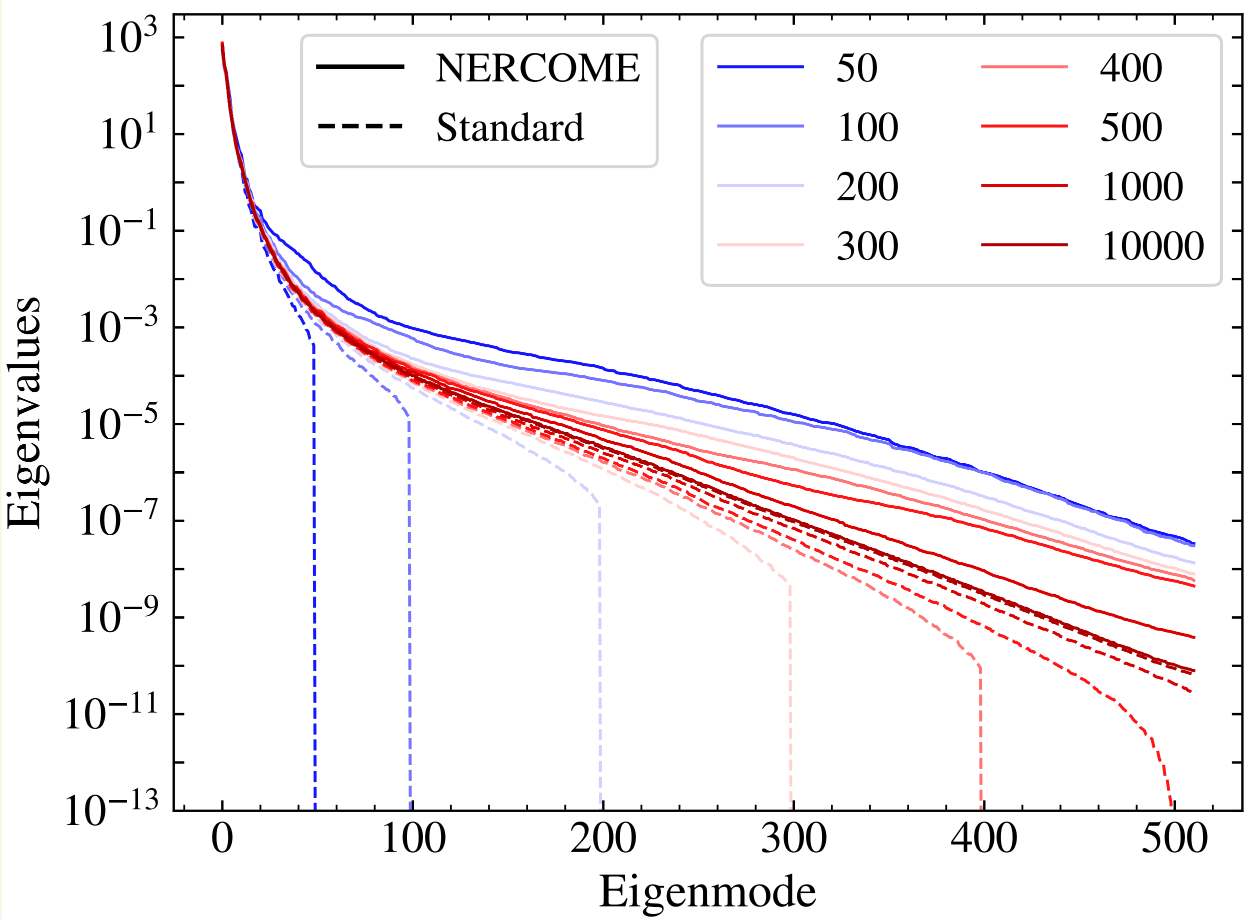


Credit : Philippe Baratta

Estimating the covariance matrix with NERCOME

NERCOME = Non-parametric eigenvalue-regularised covariance matrix estimator

Spectral decomposition of the covariance matrix



Shrinks low eigenvalues to make it invertible

See Joachimi 2016 for details

NERCOME covariances are estimated with Julian's code

Set-up for the MCMC : Fit of the CDM P(k) in real space

$$\chi^2(\hat{\boldsymbol{P}}(k)|\boldsymbol{\theta}) = \left[[\hat{\boldsymbol{P}}(k) - \boldsymbol{P}(k;\boldsymbol{\theta})]^T \mathbf{C}^{-1} [\hat{\boldsymbol{P}}(k) - \boldsymbol{P}(k;\boldsymbol{\theta})]\right]$$

- DEMNUni ← $\hat{\boldsymbol{P}}(k)$
- Halofit (with neutrinos « TakaBird ») ← $\boldsymbol{P}(k;\boldsymbol{\theta})$
- [Gaussian, DEMNUni-Cov, covmos, NERCOME] ← $\mathbf{C}$

Set-up for the MCMC : Fit of the CDM P(k) in real space

$$\chi^2(\hat{\boldsymbol{P}}(k)|\boldsymbol{\theta}) = \left[[\hat{\boldsymbol{P}}(k) - \boldsymbol{P}(k;\boldsymbol{\theta})]^T \mathbf{C}^{-1} [\hat{\boldsymbol{P}}(k) - \boldsymbol{P}(k;\boldsymbol{\theta})]\right]$$

DEMNUni ← $\hat{\boldsymbol{P}}(k)$

Halofit (with neutrinos « TakaBird ») ← $\boldsymbol{P}(k;\boldsymbol{\theta})$

[Gaussian, DEMNUni-Cov, covmos, NERCOME] ← $\mathbf{C}$

$k = [0.01, k_{\rm max}],\ \Delta k = 0.01\ h/{\rm Mpc}$ with $k_{\rm max} = [0.1,\ 0.3]$ ➔ $N_k = [16,\ 44]$

Set-up for the MCMC : Fit of the CDM P(k) in real space

$$\chi^2(\hat{\boldsymbol{P}}(k)|\boldsymbol{\theta}) = \left[[\hat{\boldsymbol{P}}(k) - \boldsymbol{P}(k;\boldsymbol{\theta})]^T \mathbf{C}^{-1} [\hat{\boldsymbol{P}}(k) - \boldsymbol{P}(k;\boldsymbol{\theta})]\right]$$

- DEMNUni ← $\hat{\boldsymbol{P}}(k)$
- Halofit (with neutrinos « TakaBird ») ← $\boldsymbol{P}(k;\boldsymbol{\theta})$
- [Gaussian, DEMNUni-Cov, covmos, NERCOME] ← $\mathbf{C}^{-1}$

$k = [0.01, k_{\max}],\ \Delta k = 0.01\ h/\mathrm{Mpc}$ with $k_{\max} = [0.1,\ 0.3]$ ➔ $N_k = [16,\ 44]$

Fitting 5 *uncorrelated* redshifts $\chi^2 = \sum_{i=0}^{n_z=5} \chi^2(z_i)$ with $z_i = [0,\ 0.5,\ 1,\ 1.5,\ 2]$

Set-up for the MCMC : Fit of the CDM P(k) in real space

$$\chi^2(\hat{\boldsymbol{P}}(k)|\boldsymbol{\theta}) = \left[[\hat{\boldsymbol{P}}(k) - \boldsymbol{P}(k;\boldsymbol{\theta})]^T \mathbf{C}^{-1} [\hat{\boldsymbol{P}}(k) - \boldsymbol{P}(k;\boldsymbol{\theta})]\right]$$

- $\hat{\boldsymbol{P}}(k)$: DEMNUni
- $\boldsymbol{P}(k;\boldsymbol{\theta})$: Halofit (with neutrinos « TakaBird »)
- $\mathbf{C}^{-1}$: [Gaussian, DEMNUni-Cov, covmos, NERCOME]

$k = [0.01, k_{\rm max}],\ \Delta k = 0.01\ h/{\rm Mpc}$ with $k_{\rm max} = [0.1,\ 0.3]$ ➔ $N_k = [16,\ 44]$

Fitting 5 *uncorrelated* redshifts $\chi^2 = \sum_{i=0}^{n_z=5} \chi^2(z_i)$ with $z_i = [0,\ 0.5,\ 1,\ 1.5,\ 2]$

4 parameters

$\boldsymbol{\theta}$	Priors
$\Omega_{\rm b}$	$[0.01,\ 0.1]$
$\Omega_{\rm cdm}$	$[0.01,\ 0.8]$
h	$[0.3,\ 1.5]$
m_ν [eV]	$[0,\ 1]$

Set-up for the MCMC : Fit of the CDM P(k) in real space

$$\chi^2(\hat{\boldsymbol{P}}(k)|\boldsymbol{\theta}) = \left[[\hat{\boldsymbol{P}}(k) - \boldsymbol{P}(k;\boldsymbol{\theta})]^T \mathbf{C}^{-1} [\hat{\boldsymbol{P}}(k) - \boldsymbol{P}(k;\boldsymbol{\theta})]\right]$$

- $\hat{\boldsymbol{P}}(k)$ ← DEMNUni
- $\boldsymbol{P}(k;\boldsymbol{\theta})$ ← Halofit (with neutrinos « TakaBird »)
- $\mathbf{C}$ ← [Gaussian, DEMNUni-Cov, covmos, NERCOME]

$k = [0.01, k_{\max}],\ \Delta k = 0.01\ h/\mathrm{Mpc}$ with $k_{\max} = [0.1,\ 0.3]$ ➔ $N_k = [16,\ 44]$

Fitting 5 *uncorrelated* redshifts $\chi^2 = \sum_{i=0}^{n_z=5} \chi^2(z_i)$ with $z_i = [0,\ 0.5,\ 1,\ 1.5,\ 2]$

4 parameters

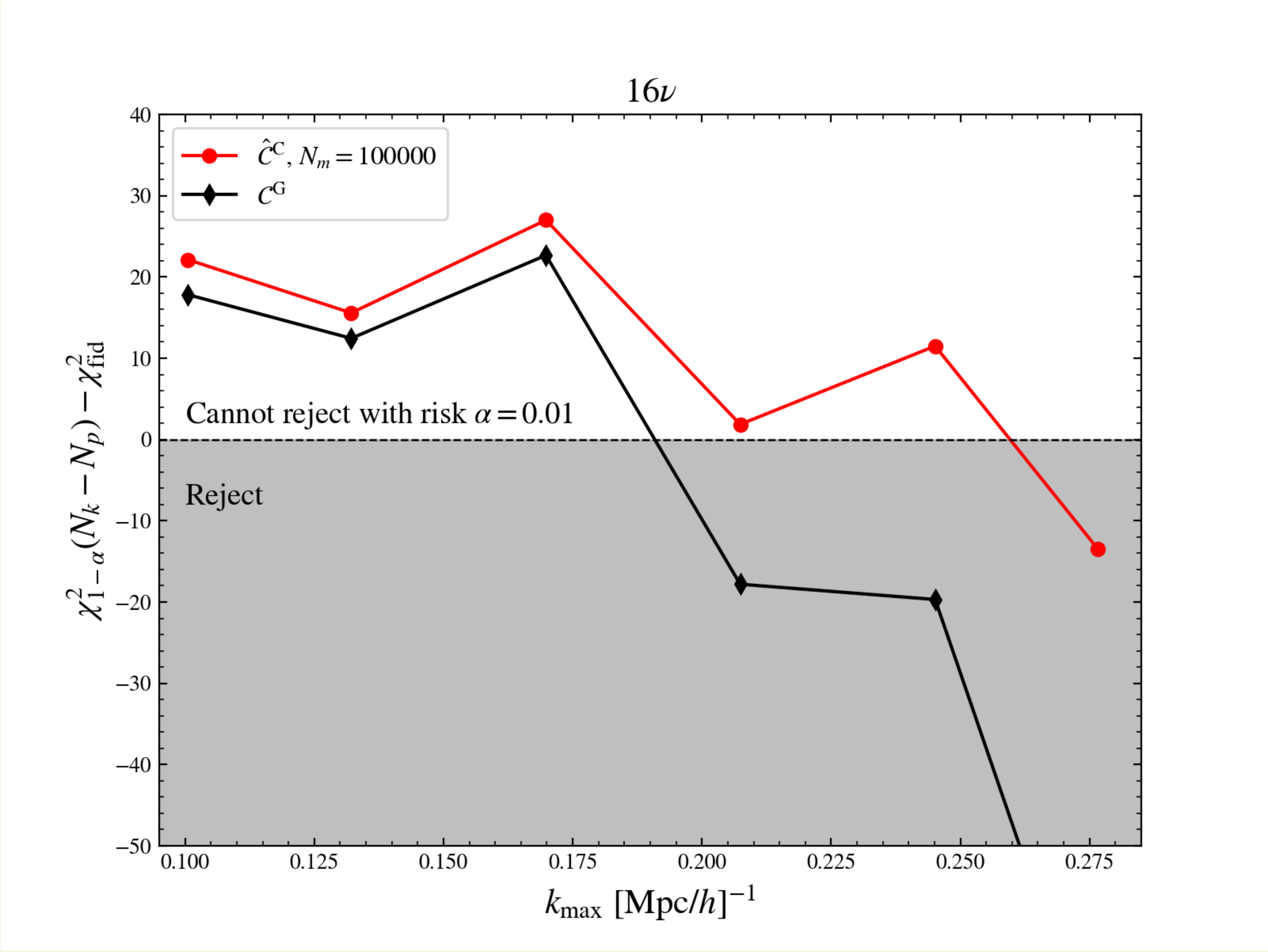
$\boldsymbol{\theta}$	Priors
Ω_{b}	$[0.01,\ 0.1]$
Ω_{cdm}	$[0.01,\ 0.8]$
h	$[0.3,\ 1.5]$
m_ν [eV]	$[0,\ 1]$

Check for convergence with Gelman-Rubin test : R-1 > 0.01

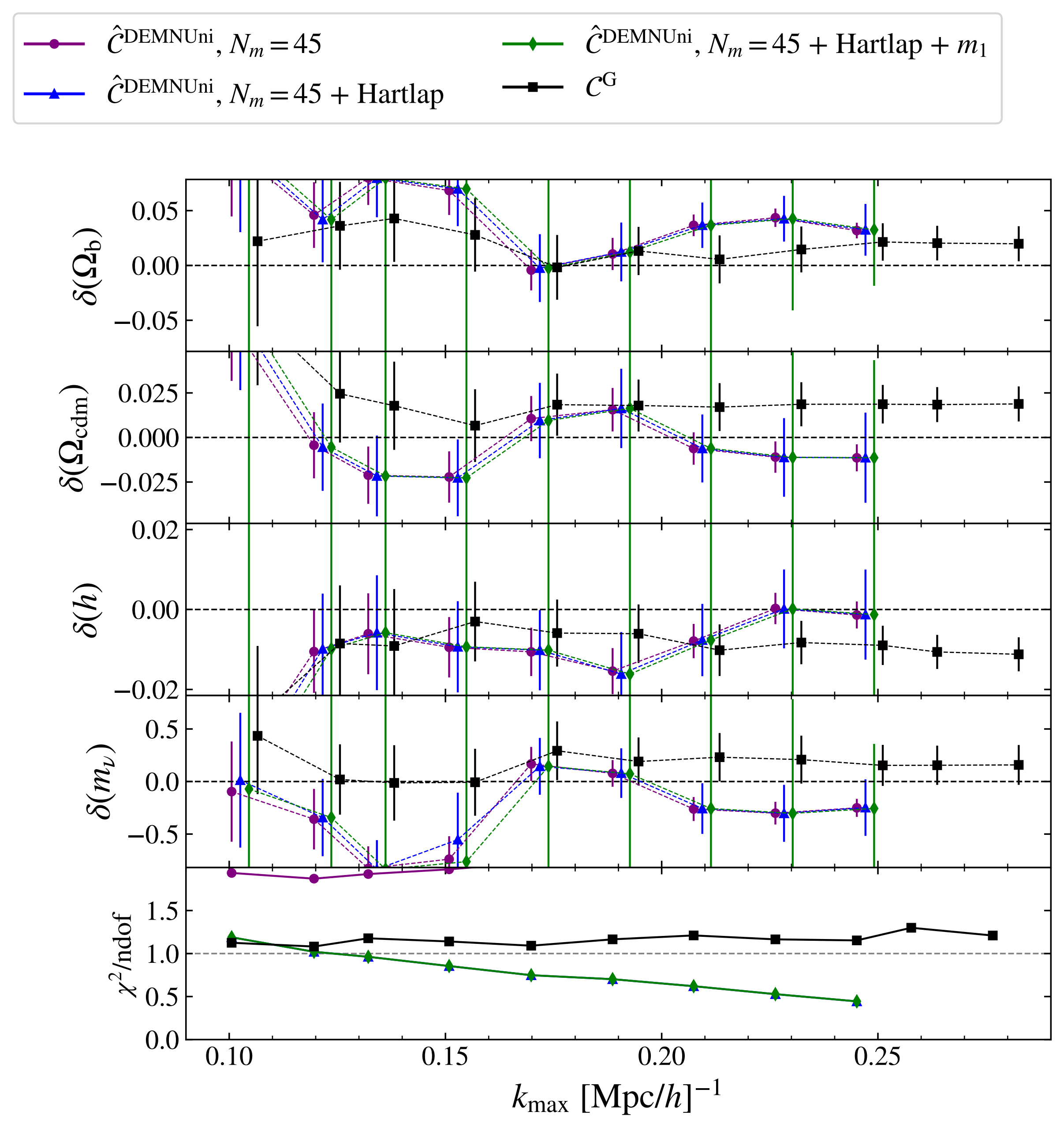
Check chains acceptance rate > 0.2

Softwares :
- MontePython for MCMC
- Class + Halofit for the model
- Nbodykit for P(k) estimation

Goodness of fit test with significance α = 0.01 : Can't trust the model for k_{max} > 0.2 h/Mpc



III. Results from MCMC on the DEMNUni power spectrum



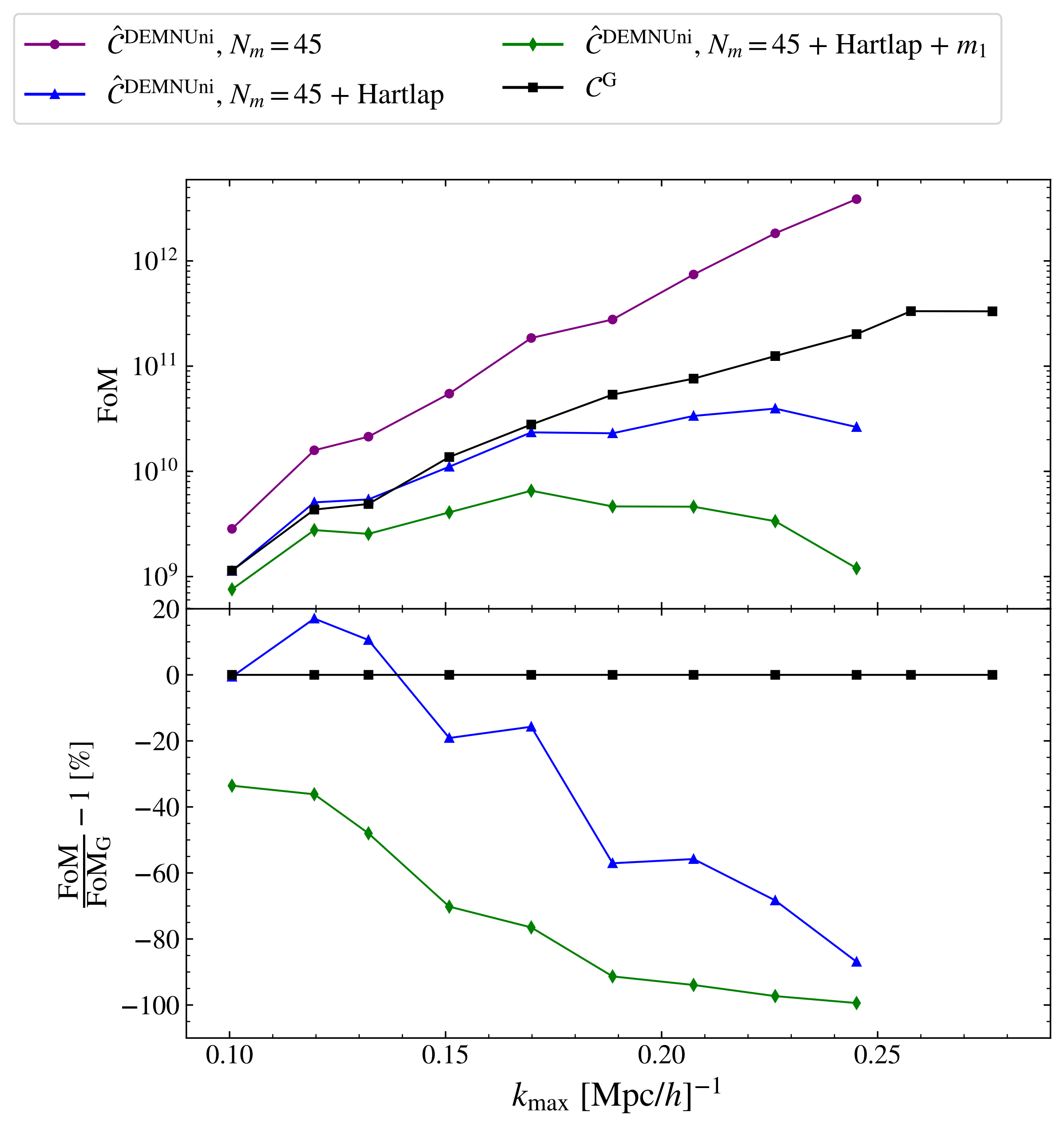
$$\delta(\theta) = \frac{\hat{\theta}}{\theta^{\rm DEMNUni}} - 1$$

$$k_{\rm max} = [0.1,\ 0.3]$$

$$N_k = [16,\ 44]$$

- With Gaussian cov the best-fit is more stable because analytic (no noise)
- Hartlap bias causes an underestimation of the errors
- Even with the Hartlap correction, the chi2 is too low
- m_1 factor, should correct for sampling-noise effects by inflating the error bars ➔ increase by a large amount

III. Results from MCMC on the DEMNUni power spectrum



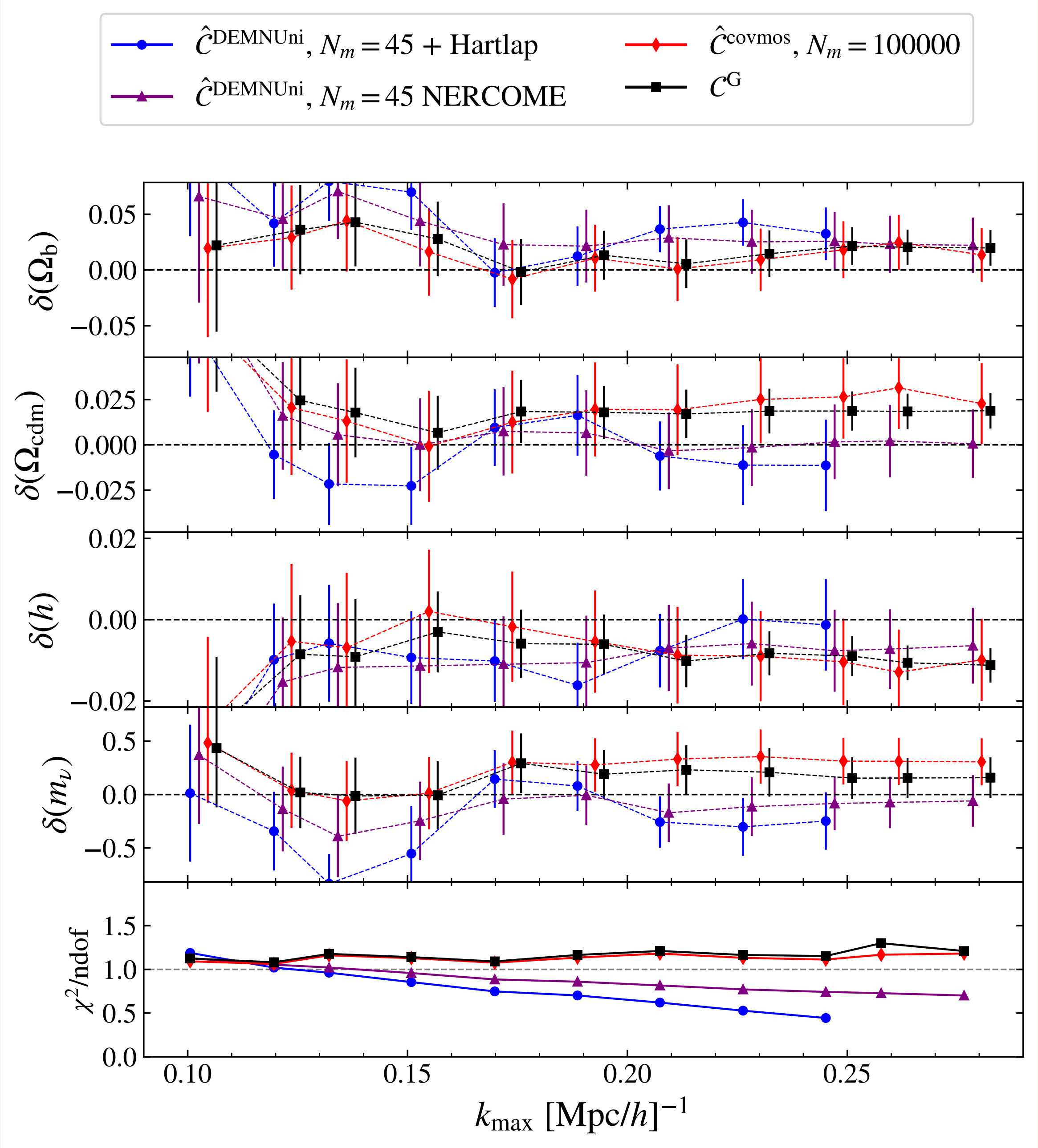
$$\mathrm{FoM} = \frac{1}{\sqrt{\det(\Phi)}}$$

$$k_{\rm max} = [0.1,\ 0.3]$$

$$N_k = [16,\ 44]$$

- Hartlap bias causes an underestimation of the errors
- m_1 facteur, should correct for sampling-noise effects by inflating the error bars
- BUT In the limit where $N_m \sim N_k$, we loose constraining power when increasing k_{max}

III. Results from MCMC on the DEMNUni power spectrum



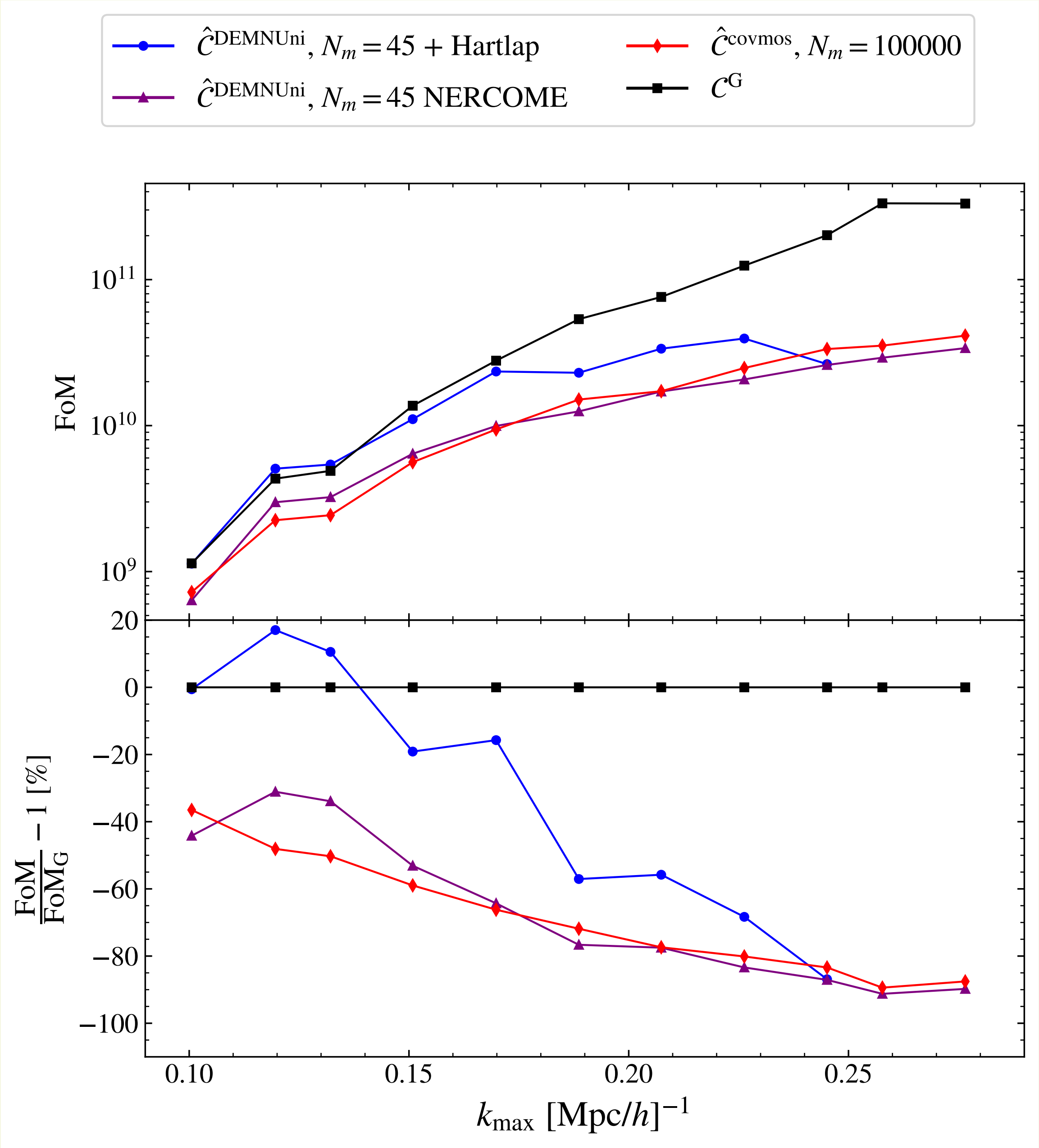
$$\delta(\theta) = \frac{\hat{\theta}}{\theta^{\mathrm{DEMNUni}}} - 1$$

$$k_{\mathrm{max}} = [0.1,\ 0.3]$$

$$N_k = [16,\ 44]$$

- The chi2 for covmos Nm = 100000 is stable
- Non-Gaussian covariance increases error and moves best-fit
- With NERCOME and covmos the best-fit is more stable
- With NERCOME the best-fit is closer to the true cosmology ➔ could be luck or a bias in the estimator (need more tests to conclude)
- The chi2 for NERCOME is better but still biased

III. Results from MCMC on the DEMNUni power spectrum



$$\mathrm{FoM} = \frac{1}{\sqrt{\det(\Phi)}}$$

$$k_{\rm max} = [0.1,\ 0.3]$$

$$N_k = [16,\ 44]$$

- With NERCOME and covmos the errors are more stable
- NERCOME gives a good estimation of errors even with low N_m
- If we consider covmos N_m = 100000 as the « truth » ➔ Non-Gaussian covariance degrades the FoM up to 80%

Conclusions

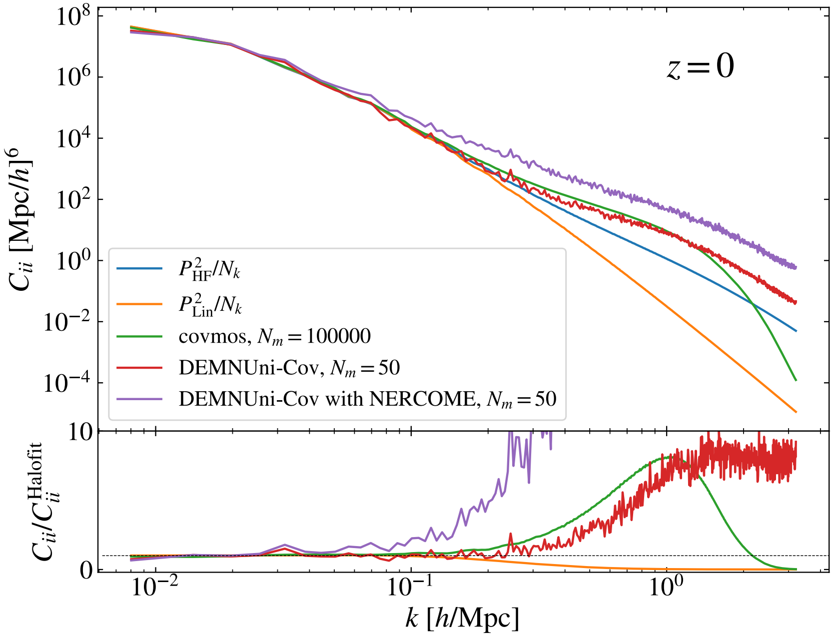
Covariance estimation will be crucial for Euclid

- **Large number of parameters and data points when combining probes**
- **Covariance has an effect on the error and the best-fit !** (Cannot see this with Fisher analysis)
  ➔ Need to understand and control these effects to achieve Euclid requirements
- Analytic prediction of sampling noise effects are useful to understand what is going on,
  but should not be applied in their limits
- The Trispectrum part of the covariance increases the errors : **here 80% decrease of the FoM**
- **Covmos is a reliable tool to produce fast and accurate covariance matrices** (can be adapted for cross-correlation of probes)
- NERCOME is handy if N_m is low, but need more testing to asses biases

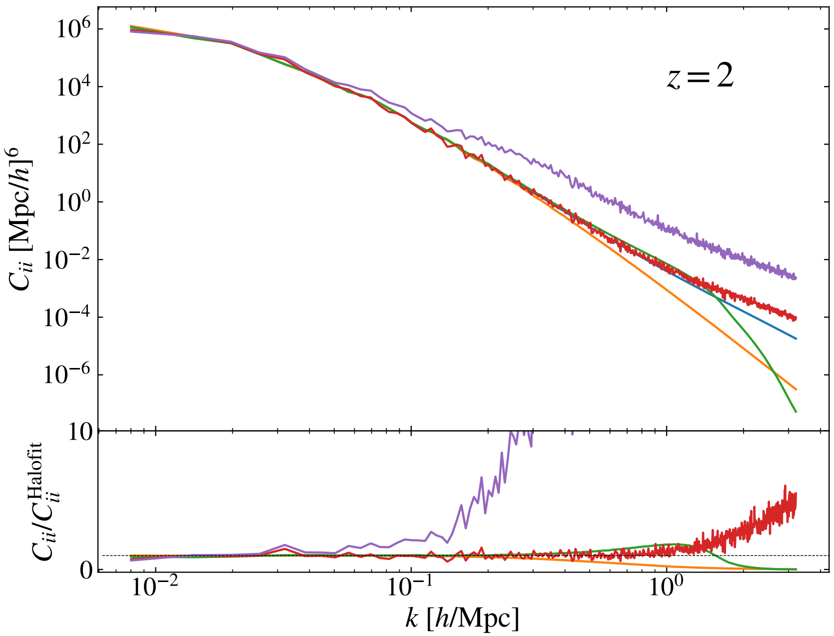
Thank you !

Back up

Diagonal

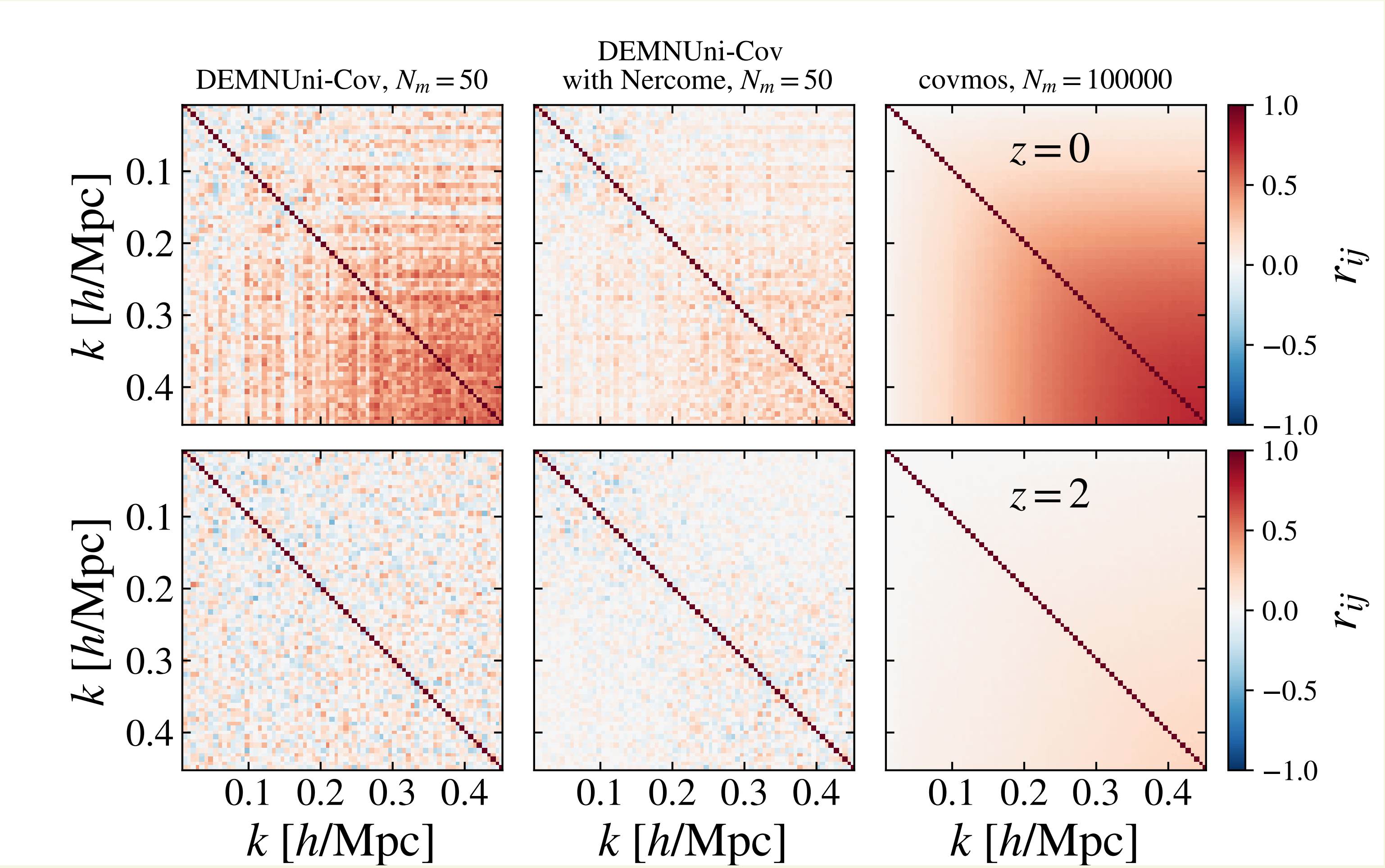


$z = 0$
C_{ii} [Mpc/h]^6
$P_{\rm HF}^2/N_k$
$P_{\rm Lin}^2/N_k$
covmos, $N_m = 100000$
DEMNUni-Cov, $N_m = 50$
DEMNUni-Cov with NERCOME, $N_m = 50$
$C_{ii}/C_{ii}^{\rm Halofit}$
k [h/Mpc]



$z = 2$
C_{ii} [Mpc/h]^6
$C_{ii}/C_{ii}^{\rm Halofit}$
k [h/Mpc]

Correlation matrix



La méthode Nercome

$$\hat{\mathbf{S}} = \frac{1}{N_S - 1}\, \mathbf{X}\, \mathbf{X}^\tau \,, \tag{1}$$

which is unbiased, $\langle \hat{\mathbf{S}} \rangle = \mathbf{\Sigma}$. A key idea of NERCOME is to divide the dataset into two subsamples, $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)$, with $\mathbf{X}_1$ an $N_D \times s$ matrix and $\mathbf{X}_2$ an $N_D \times (N_S - s)$ matrix. The sample covariance can also be measured from each subset, denoted by $\hat{\mathbf{S}}_i$ with $i = 1, 2$. The estimator uses the diagonal decomposition of these estimates, $\hat{\mathbf{S}}_i = \mathbf{U}_i\, \mathbf{D}_i\, \mathbf{U}_i^\tau$, where $\mathbf{U}$ is the matrix of eigenvectors and $\mathbf{D}$ is a diagonal matrix with entries $d_{\alpha\beta} = \delta_{\alpha\beta}\, \lambda_\alpha$, where the λ_α are the eigenvalues and δ is the Kronecker delta.

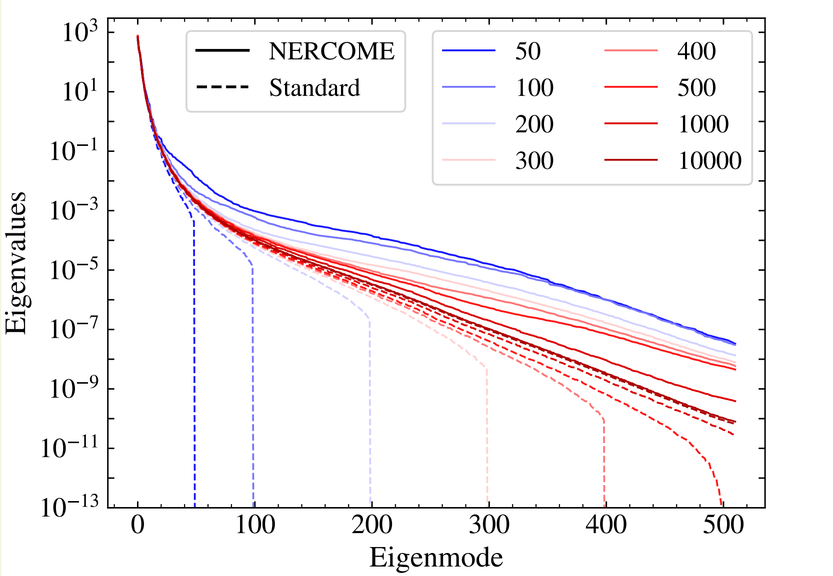
The NERCOME estimation process consists of three steps: 1. apply the basic estimator

$$\hat{\mathbf{Z}} \equiv \mathbf{U}_1\, \mathrm{diag}\left(\mathbf{U}_1^\tau\, \hat{\mathbf{S}}_2\, \mathbf{U}_1\right)\, \mathbf{U}_1^\tau \tag{2}$$

to a given subdivision of $\mathbf{X}$; 2. average over different compositions of $(\mathbf{X}_1, \mathbf{X}_2)$ for a given location s of the split, of which there are $\binom{N_S}{s}$; 3. find the optimal location of the data vector split by minimising

$$Q(s) = \left|\left|\overline{\hat{\mathbf{Z}}}(s) - \overline{\hat{\mathbf{S}}}_2(s)\right|\right|_{\mathrm{F}}^2 \,, \tag{3}$$

where the bar denotes the average of step (2.), and where $||\mathbf{A}||_{\mathrm{F}}^2 = \mathrm{Tr}(\mathbf{A}\,\mathbf{A}^\tau)$ is the Frobenius matrix norm. An estimate for the inverse covariance is then simply provided by the inverse of the covariance estimator.



I. Introduction

Estimate the full covariance from simulations

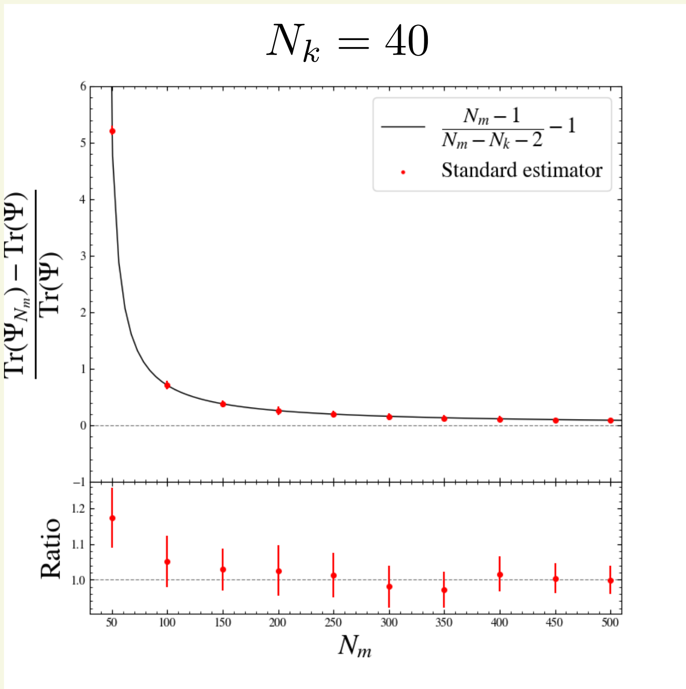
$$\hat{C}_{ij} = \frac{1}{N_m - 1}\left[\sum_{n}^{N_m}[P^{(n)}(k_i) - \bar{P}(k_i)][P^{(n)}(k_j) - \bar{P}(k_j)]\right]$$

Unbiased estimator of $\mathbf{C}$: $E[\hat{\mathbf{C}}] = \mathbf{C}$

BUT we need the precision matrix : $\mathbf{\Psi} \equiv \mathbf{C}^{-1}$ —Standard estimator→ $\hat{\mathbf{\Psi}} = \hat{\mathbf{C}}^{-1}$

Biased estimator of $\mathbf{\Psi}$: $E[\hat{\mathbf{\Psi}}] = \frac{N_m - 1}{N_m - N_k - 2}\mathbf{\Psi}$ [Hartlap 2007]

N_k → Number of bins



Estimation of the Hartlap bias :

Comparaison between biased precision matrix and the Truth

Estimated from N_m mocks

VS

Estimated from 100'000 mocks