A brief Introduction to CAMB and CosmoMC

Georgios Zacharegkas

University of Chicago

Monday, November 26 2018

Introduction

- In the current era of Cosmology, the very good agreement between theoretical predictions and observations has led to the need for high-precision tools for data analysis to look for new physics
- With a huge volume of data at hand and a powerful cosmological model we need code that is capable of fast and accurate analysis that leads to constraints of the model parameters
- There are a two main steps involved in this:
  (I) We need to solve the equations of the model numerically and from that get the predictions for the observables of interest
  (II) Compare the predictions of the model at various points in parameter space to data and, based on the likelihood, find the best-fit parameters

List of some Numerical Tools

- **Code that deals with the physics:**
  CAMB (Code for Anisotropies in the Microwave Background), CMBFast,...
- **Code for parameter-space exploration**
  CosmoMC (COSMOlogical Monte-Carlo), MontePython, Cosmosis,...
- **Analysis of the final chains**
  GetDist (comes with a convenient GUI),...

CAMB

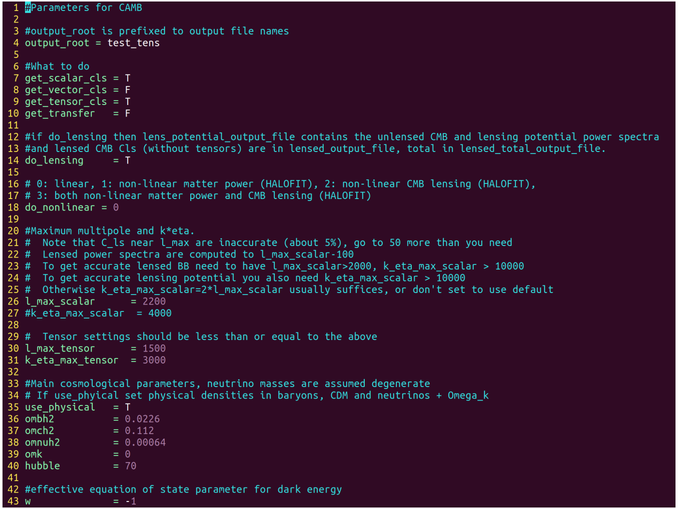
Overview and Installation

- It is based on CMBFast and is written in Fortran
- Is organized in modules, making it easy to modify, e.g. to include new physics
- It is used by many for papers, e.g. Planck
- Python wrapper exists (pycamb): Python CAMB documentation
- **Installation**:
  - (I) Download source code: clone git repository
  - (II) Unzip file and cd to extracted directory
  - (III) Compile the code (need gfortran): run ”make clean” and ”make”
  - (IV) To install pycamb, just run ”python setup.py install - -user” in the pycamb directory

  For more see: CAMB ReadMe

Initialize a run

- Input parameters are defined in a ”params.ini” file that looks like this:



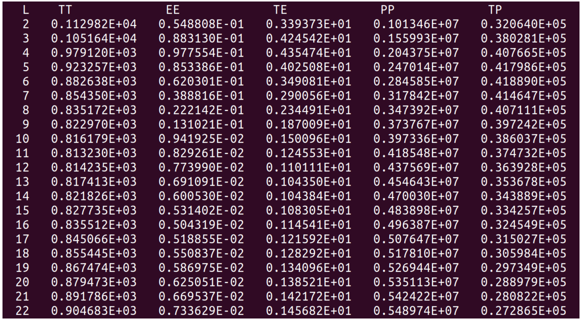
```
#Parameters for CAMB

#output_root is prefixed to output file names
output_root = test_tens

#What to do
get_scalar_cls = T
get_vector_cls = F
get_tensor_cls = T
get_transfer   = F

#if do_lensing then lens_potential_output_file contains the unlensed CMB and lensing potential power spectra
#and lensed CMB Cls (without tensors) are in lensed_output_file, total in lensed_total_output_file.
do_lensing     = T

# 0: linear, 1: non-linear matter power (HALOFIT), 2: non-linear CMB lensing (HALOFIT),
# 3: both non-linear matter power and CMB lensing (HALOFIT)
do_nonlinear = 0

#Maximum multipole and k*eta.
#  Note that C_ls near l_max are inaccurate (about 5%), go to 50 more than you need
#  Lensed power spectra are computed to l_max_scalar-100
#  To get accurate lensed BB need to have l_max_scalar>2000, k_eta_max_scalar > 10000
#  To get accurate lensing potential you also need k_eta_max_scalar > 10000
#  Otherwise k_eta_max_scalar=2*l_max_scalar usually suffices, or don't set to use default
l_max_scalar      = 2200
#k_eta_max_scalar  = 4000

#  Tensor settings should be less than or equal to the above
l_max_tensor      = 1500
k_eta_max_tensor  = 3000

#Main cosmological parameters, neutrino masses are assumed degenerate
# If use_phyical set physical densities in baryons, CDM and neutrinos + Omega_k
use_physical   = T
ombh2          = 0.0226
omch2          = 0.112
omnuh2         = 0.00064
omk            = 0
hubble         = 70

#effective equation of state parameter for dark energy
w              = -1
```

Run the code and Output files

- The ”params.ini” file contains all information about the cosmology (cosmological parameter values, etc) and also this is where we ask CAMB to output the observables of interest
- Run ”./camb params.ini”
- Output files are named in format ”name_given_x.dat”; for example:
  - (I) x=”scalCls” for Scalar power spectrum C_ℓ’s (TT, EE, TE)
  - (II) x=”tensCls” for Tensor power spectrum C_ℓ’s (TT, EE, BB, TE) - if ”get_tensor_cls = T”
  - (III) If ”do_lensing = T” the lensed CMB power spectrum and the lensing potential are also calculated ... there are more options
  - (IV) To use the python version, in your python code just ”import camb”

What does an output file look like?

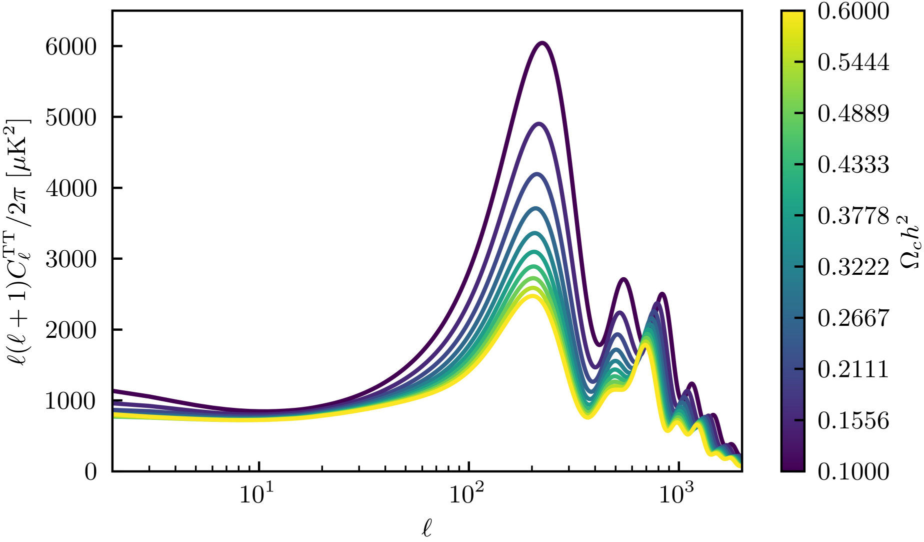
- This is an example of a ”name_given_scalCls.dat” file:



```
  L    TT           EE           TE           PP           TP
  2   0.112982E+04 0.548808E-01 0.339373E+01 0.101346E+07 0.320640E+05
  3   0.105164E+04 0.883130E-01 0.424542E+01 0.155993E+07 0.380281E+05
  4   0.979120E+03 0.977554E-01 0.435474E+01 0.204375E+07 0.407665E+05
  5   0.923257E+03 0.853386E-01 0.402508E+01 0.247014E+07 0.417986E+05
  6   0.882638E+03 0.620301E-01 0.349081E+01 0.284585E+07 0.418890E+05
  7   0.854350E+03 0.388816E-01 0.290056E+01 0.317842E+07 0.414647E+05
  8   0.835172E+03 0.222142E-01 0.234491E+01 0.347392E+07 0.407111E+05
  9   0.822970E+03 0.131021E-01 0.187009E+01 0.373767E+07 0.397242E+05
 10   0.816179E+03 0.941925E-02 0.150096E+01 0.397336E+07 0.386037E+05
 11   0.813230E+03 0.829261E-02 0.124553E+01 0.418548E+07 0.374732E+05
 12   0.814235E+03 0.773990E-02 0.110111E+01 0.437569E+07 0.363928E+05
 13   0.817413E+03 0.691091E-02 0.104350E+01 0.454643E+07 0.353678E+05
 14   0.821826E+03 0.600530E-02 0.104384E+01 0.470030E+07 0.343889E+05
 15   0.827735E+03 0.531402E-02 0.108305E+01 0.483898E+07 0.334257E+05
 16   0.835512E+03 0.504319E-02 0.114541E+01 0.496387E+07 0.324549E+05
 17   0.845066E+03 0.518855E-02 0.121592E+01 0.507647E+07 0.315027E+05
 18   0.855445E+03 0.550837E-02 0.128292E+01 0.517810E+07 0.305984E+05
 19   0.867474E+03 0.586975E-02 0.134096E+01 0.526944E+07 0.297349E+05
 20   0.879473E+03 0.625051E-02 0.138521E+01 0.535113E+07 0.288979E+05
 21   0.891786E+03 0.669537E-02 0.142172E+01 0.542422E+07 0.280822E+05
 22   0.904683E+03 0.733629E-02 0.145682E+01 0.548974E+07 0.272865E+05
```

- For a description of name conventions and output format see: CAMB ReadMe

Example of TT power spectra



The Physics of CAMB

Initial conditions

- Start from initial conditions in the early Universe:
  "Quantum fluctuations + Inflation ⇒ Primordial perturbations"
- In CAMB:

$$\ln P_s = \ln A_s + (n_s - 1)\ln\left(\frac{k}{k_s}\right) + \frac{n_{\rm run}}{2}\left[\ln\left(\frac{k}{k_s}\right)\right]^2 + \frac{n_{\rm run,run}}{6}\left[\ln\left(\frac{k}{k_s}\right)\right]^3$$

  and

$$\ln P_t = \ln A_t + n_t \ln\left(\frac{k}{k_t}\right) + \frac{n_{t,\rm run}}{2}\left[\ln\left(\frac{k}{k_t}\right)\right]^2$$

- CAMB Input: A_s ("scalar_power_amp"), k_s ("pivot_scalar"), n_s ("scalar_spectral_index"), $n_{\rm run}$ ("scalar_nrun"), $n_{\rm run,run}$ ("scalar_nrunrun"), plus parameters for tensor

The Gauge

- Small fluctuations of $\sim 10^{-5}$ allow the use linear perturbation theory, i.e. evolve each mode independently of all others
- Two popular gauges are:
  (I) Synchronous/CDM gauge:

$$ds^2 = a^2(\tau)\left[-d\tau^2 + (\delta_{ij} + h_{ij})dx^i dx^j\right]$$

  (II) Conformal Newtonian/longitudinal gauge:

$$ds^2 = a^2(\tau)\left[-(1+2\Psi)d\tau^2 + (1-2\Phi)dx^i dx_i\right]$$

- CAMB uses the Synchronous gauge to work in, although the equations in the Newtonian gauge are also mentioned
- The formalism for these can be found here: Ma and Bertschinger (1995)
- For more notes on CAMB equations see: CAMB Notes

Background and metric perturbations evolution

- Background is fixed to homogeneous FRW:

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3}a^2\bar{\rho} - K$$

$$\frac{d}{d\eta}\left(\frac{\dot{a}}{a}\right) = -\frac{4\pi G}{3}a^2\left(\bar{\rho} + 3\bar{P}\right)$$

- Metric perturbations in synchronous gauge (the default):

$$k^2\eta - \frac{\dot{a}}{a}\dot{h} = 4\pi G a^2 \delta T^0_0$$

$$k^2\dot{\eta} = 4\pi G a^2\left(\bar{\rho} + \bar{P}\right) ik^j \delta T^0_j$$

$$\ddot{h} + 2\frac{\dot{a}}{a}\dot{h} - 2k^2\eta = -8\pi G a^2 \delta T^i_i$$

$$\ddot{h} + 6\ddot{\eta} + 2\frac{\dot{a}}{a}\left(\dot{h} + 6\dot{\eta}\right) - 2k^2\eta = 24\pi G a^2(\bar{\rho} + \bar{P})\left(\hat{k}_i\hat{k}^j - \frac{1}{3}\delta^j_i\right)\left(T^i_j - \frac{1}{3}\delta^i_j T^k_k\right)$$

The covariant approach: PSTF decomposition

- For more on the formalism that follows see: Challinor (2000)
- Typically, one would expand in spherical harmonics, $Y_{\ell m}(\theta, \phi)$
- The coordinate-free version of this is the Projected-Symmetric Trace-Free (PSTF) multipole expansion
- PSTF multipoles are indexed $A_\ell = a_1 \cdots a_\ell$ in the general basis $e_{A_\ell} = e_1 \cdots e_\ell$
- The PSTF part of the expansion reads:

$$S_{\langle ab \rangle} = \left(h^c_{(a} h^d_{b)} - \frac{1}{3} h_{ab} h^{cd} \right) S_{cd}$$

where the projector tensor is $h^{\alpha\beta} = g^{\alpha\beta} - u^\alpha u^\beta$ and the 4-velocity is u^α

- For example:

$$X(E, e^c) = \sum_{\ell=0}^{\infty} X_{A_\ell}(E) e^{A_\ell} \ ; \ \ X_{A_\ell}(E) = \frac{(2\ell+1)!}{4\pi(-2)^\ell (\ell!)^2} \int d\Omega X(E, e^c) e_{\langle A_\ell \rangle}$$

Decomposition of the polarization matrix

- The CMB radiation can be described by the Stokes parameters $\{I, Q, U, V\}$
- Intensity: $I(E, e^a)$ for photon energy E in direction e^a
- At observation point form the orthonormal tetrad $\{u^a,\, (e_1)^\alpha,\, (e_2)^\alpha,\, e^a\}$
- Polarization tensor:

$$P_{ab}(e_i)^a(e_j)^b = \frac{1}{2}\left(\begin{array}{cc} I+Q & U+V \\ U-V & I-Q \end{array}\right)$$

- If define the projector perpendicular to both u^α and e^α, $H_{ab} = h_{ab} + e_a e_b$:

$$P_{ab} = -\frac{1}{2} I(E, e^d) H_{ab} + \mathcal{P}_{ab}(E, e^d) + \frac{1}{2} V(E, e^d) \epsilon_{abc} e^c$$

where ϵ_{abc} the projected-alternating tensor and the linear polarization tensor is

$$\mathcal{P}_{ab}(e_i)^a(e_j)^b = \frac{1}{2}\left(\begin{array}{cc} Q & U \\ U & -Q \end{array}\right)$$

PSTF decomposition of the radiation field

- Intensity $I(E, e^c)$:

$$I_{A_\ell}(E) \propto \int d\Omega\; I(E, e^c) e_{\langle A_\ell \rangle}$$

- Polarization $V(E, e^c)$ (scalar function of the sphere):

$$V_{A_\ell}(E) \propto \int d\Omega\; V(E, e^c) e_{\langle A_\ell \rangle}$$

- Linear polarization E- and B-modes:

$$\mathcal{E}_{A_\ell}(E) \propto \int d\Omega\; e_{\langle A_{\ell-2}} \mathcal{P}_{a_{\ell-1} a_\ell \rangle}(E, e^c)$$

$$\mathcal{B}_{A_\ell}(E) \propto \int d\Omega\; e_b \epsilon^{bd}{}_{\langle a_\ell} e_{A_{\ell-2}} \mathcal{P}_{a_{\ell-1} \rangle d}(E, e^c)$$

TT Power spectrum and correlation function

- PSTF-decompose the temperature field:

$$\delta_T(e^c) \propto \sum_{\ell=0}^{\infty} I_{A_\ell} e^{A_\ell}$$

- Then, the temperature power spectrum is defined by:

$$\left\langle I_{A_\ell} I^{B_{\ell'}} \right\rangle \propto C_\ell^{TT} \delta_{\ell\ell'} h_{\langle A_\ell \rangle}^{\langle B_\ell \rangle}$$

- Calculate the correlation function:

$$\langle \delta_T(e^c) \delta_T(e'^c) \rangle = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} C_\ell^{TT} P_\ell(-e^c e'_c)$$

  where P_ℓ are the Legendre polynomials

- Similarly for polarization and cross-power spectra

Boltzmann equation for intensity

- (Phase-space evolution) = (Photon-matter interactions):

$$\mathcal{L}[E^{-3}P_{ab}(E,e^c)] = K_{ab}(E,e^c)$$

- The multipole-decomposed equations are:

$$\begin{aligned} \dot{I}_{A_\ell} + \frac{4}{3}\Theta I_{A_\ell} - \frac{\ell}{(2\ell+1)} D_{\langle a_\ell} I_{A_{\ell-1}\rangle} + \frac{4}{3} I A_{a_1}\delta_{\ell 1} - \frac{8}{15} I \sigma_{a_1 a_2}\delta_{\ell 2} \\ = -n_e\sigma_T \left(I_{A_\ell} - I\delta_{\ell 0} - \frac{4}{3} I v_{a_1}\delta_{\ell 1} - \frac{2}{15}\zeta_{a_1 a_2}\delta_{\ell 2} \right) \end{aligned}$$

- Definitions: $\Theta \equiv \nabla_\alpha u^\alpha$, $A_\alpha \equiv \dot{u}_\alpha$, $\sigma_{ab} \equiv D_{\langle a} u_{b\rangle}$, $\zeta_{ab} \equiv 3I_{ab}/4 + 9\mathcal{E}_{ab}/2$, $I(e^c) \equiv \int dE I(E,e^c)$, v_a is the electron 4-velocity, $D_a S_{b\ldots c} \equiv h_a^d h_b^e \ldots h_d^f \nabla_d S_{e\ldots f}$, $D_{a_\ell} \equiv D_{a_1} D_{a_2} \ldots D_{a_\ell}$ and $\dot{} \equiv u^a \nabla_a$

Boltzmann equations for polarization

- The multipole-decomposed, energy-integrated equations are:

$$\dot{\mathcal{E}}_{A_\ell} + \frac{4}{3}\Theta\mathcal{E}_{A_\ell} + \frac{(\ell+3)(\ell-1)}{(\ell+1)^2}D^b\mathcal{E}_{bA_\ell} - \frac{\ell}{2\ell+1}D_{\langle a_\ell}\mathcal{E}_{A_{\ell-1}\rangle} - \frac{2}{\ell+1}\mathrm{curl}\mathcal{B}_{A_\ell}$$
$$= -n_e\sigma_T\left(\mathcal{E}_{A_\ell} - \frac{2}{15}\zeta_{a_1a_2}\delta_{\ell 2}\right)$$

$$\dot{\mathcal{B}}_{A_\ell} + \frac{4}{3}\Theta\mathcal{B}_{A_\ell} + \frac{(\ell+3)(\ell-1)}{(\ell+1)^2}D^b\mathcal{B}_{bA_\ell} + \frac{\ell}{2\ell+1}D_{\langle a_\ell}\mathcal{B}_{A_{\ell-1}\rangle} - \frac{2}{\ell+1}\mathrm{curl}\mathcal{E}_{A_\ell} = 0$$

$$\dot{V}_{A_\ell} + \frac{4}{3}\Theta V_{A_\ell} - \frac{\ell}{(2\ell+1)}D_{\langle a_\ell}V_{A_{\ell-1}\rangle} + D^bV_{bA_\ell} = -n_e\sigma_T\left(V_{A_\ell} - \frac{1}{2}V_{a_1}\delta^1_\ell\right)$$

Scalar-vector-tensor decomposition and Harmonic expansion

- Linearity allows to decompose general rank-ℓ PSTF tensors into constituent rank-m tensors: Scalars ($m = 0$), vectors ($m = 1$), tensors ($m = 2$), etc:

$$X_{A_\ell} = \sum_{m=0}^{\ell} X^m_{A_\ell}$$

- To solve the equations numerically the "div"'s and "curl"'s must go away, so a Harmonic expansion is required in terms of $Q^m_{A_m}$, the eigenfunctions of the comoving Laplacian operator $S^2 D_a D^a$:

$$D^2 Q^m_{A_m} \equiv D^a D_a Q^m_{A_m} = \frac{k^2}{S^2} Q^m_{A_m}$$

where the "covariantly"-defined scale factor S satisfies $\dot{S}/S = \Theta/3$

Harmonic expansion of the scalar-vector-tensor decomposed radiation field

- For the intensity:

$$I^m_{A_\ell} \propto \sum_{k,\pm} I^{m\pm}_\ell(k) Q^{m\pm}_{A_\ell}$$

- The polarizations become:

$$\mathcal{E}^m_{A_\ell} \propto \sum_{k,\pm} \mathcal{E}^{m\pm}_\ell(k) Q^{m\pm}_{A_\ell}$$

  and

$$\mathcal{B}^m_{A_\ell} \propto \sum_{k,\pm} \mathcal{B}^{m\pm}_\ell(k) Q^{m\pm}_{A_\ell}$$

- Substitute these into Boltzmann hierarchy

Final Boltzmann Hierarchy

$$I_\ell' + \frac{k}{2\ell+1}\left[\beta_{\ell+1}^m \frac{(\ell+1)^2-m^2}{\ell+1} I_{\ell+1} - \ell I_{\ell-1}\right]$$
$$= -S n_e \sigma_T \left(I_\ell - I\delta_{\ell 0} - \frac{4}{3} v \delta_{\ell 1} - \frac{2}{15}\zeta\delta_{\ell 2}\right) + \frac{8}{15} k\sigma\delta_{\ell 2} - 4h'\delta_{\ell 0} - \frac{4}{3} k A \delta_{\ell 1}$$

$$\mathcal{E}_\ell^{m\pm\prime} + k\left[\beta_{\ell+1}^m \frac{(\ell+3)(\ell-1)}{(\ell+1)^3}\frac{(\ell+1)^2-m^2}{(2\ell+1)}\mathcal{E}_{\ell+1}^{m\pm} - \frac{\ell}{2\ell+1}\mathcal{E}_{\ell-1}^{m\pm} - \frac{2m}{\ell(\ell+1)}\sqrt{\beta_0^m}\mathcal{B}_\ell^{m\mp}\right]$$
$$= -S n_e \sigma_T \left(\mathcal{E}_\ell^{m\pm} - \frac{2}{15}\zeta^{m\pm}\delta_{\ell 2}\right)$$

$$\mathcal{B}_\ell^{m\pm\prime} + k\left[\beta_{\ell+1}^m \frac{(\ell+3)(\ell-1)}{(\ell+1)^3}\frac{(\ell+1)^2-m^2}{(2\ell+1)}\mathcal{B}_{\ell+1}^{m\pm} - \frac{\ell}{2\ell+1}\mathcal{B}_{\ell-1}^{m\pm} + \frac{2m}{\ell(\ell+1)}\sqrt{\beta_0^m}\mathcal{E}_\ell^{m\mp}\right] = 0$$

How many equations are solved and how?

- CAMB solves the equations in the synchronous gauge, a.k.a the CDM gauge
- Uses the Runge-Kutta integration scheme
- In total it solves for:
  - (I) A few hundred or more equations for ℓ modes in the Boltzmann hierarchy
  - (II) A few hundred decoupled wavemodes $\mathbf{k}$
  - (III) ≤ 3 types of perturbations, i.e. scalar, vector, tensor
  - (IV) It also evolves the equations for baryons, dark matter, etc
  - (V) Plus it solves for neutrinos
  - (VI) And evolves the gravitational perturbations
  - (VII) It does the above over conformal time η steps

Transfer function and observables

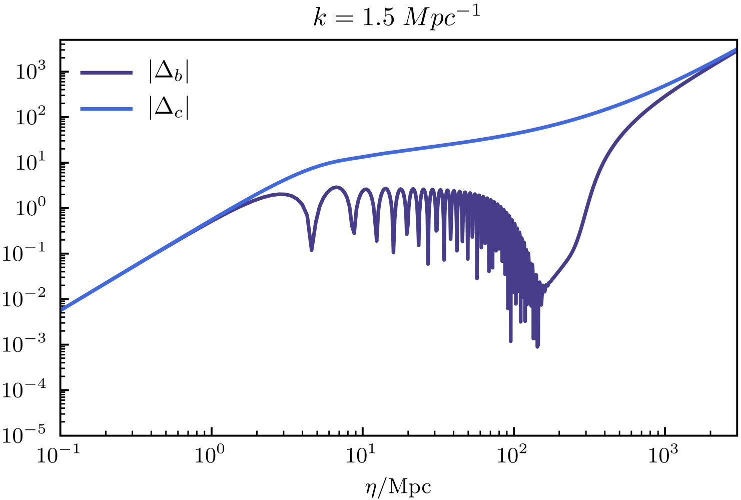
- In the linear approximation primordial quantities are related to later values via the transfer function, defined by

$$T_X^{(\ell)}(\mathbf{k}, \eta) \equiv \frac{X^{(\ell)}(\mathbf{k}, \eta)}{X^{(\ell)}(\mathbf{k}, \eta_i)}$$

- Since early universe to today many things change
  $\Rightarrow$ Need to evolve several modes over time
  $\Rightarrow$ This is what CAMB does that by evolving the Boltzmann equations
- So, finally, CAMB calculates the power spectra using these transfer functions by integrating for all modes over the primordial power spectrum:

$$C_\ell^{XY} \propto \int d\ln k P_R(k) T_X^{(\ell)}(k) T_Y^{(\ell)}(k)$$

Example of time evolution of perturbations



CosmoMC

Overview and Installation

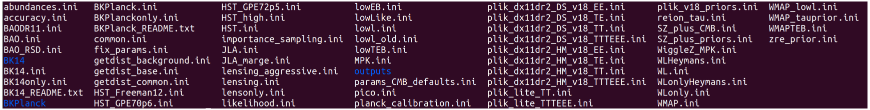
- Performs the MCMC exploration of the parameter space
- Written in Fortran and fully integrated with CAMB, i.e. in each run CosmoMC basically performs what we saw above to get the observables it needs
- Python wrapper exists (cobaya)
- **Installation**:
  - (I) Download code (e.g., clone git repository)
  - (II) Unzip file and cd to extracted directory
  - (III) Compile the code: run ”make”
  - (IV) To use CosmoMC with Planck data we also need to download and compile the Planck likelihood: Planck Legacy Archive

Initialize a run

- Like for CAMB, to initialize a CosmoMC run we need to setup the ".ini" file
- It contains:
  (I) The data we want to use and the corresponding likelihood
  (II) The options for the cosmological parameters and the general settings
  (III) Settings/options for the chains we will run
- To run the chains, we have two options:
  (I) Do "./cosmomc initfile.ini" to run a single chain
  (II) Do "mpirun -n x ./cosmomc initfile.ini" to run in parallel x chains (make sure you have mpi setup)

Lots of built-in likelihoods and data

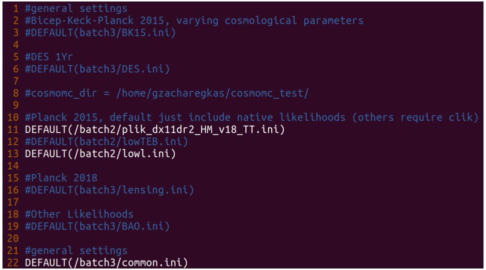
- We can find likelihoods in the "/batch2/" directory:



```
abundances.ini   BKPlanck.ini            HST_GPE72p5.ini          lowEB.ini                plik_dx11dr2_DS_v18_EE.ini      plik_v18_priors.ini  WMAP_lowl.ini
accuracy.ini     BKPlanckonly.ini        HST_high.ini             lowLike.ini              plik_dx11dr2_DS_v18_TE.ini      reion_tau.ini        WMAP_tauprior.ini
BAODR11.ini      BKPlanck_README.txt     HST.ini                  lowl.ini                 plik_dx11dr2_DS_v18_TT.ini      SZ_plus_CMB.ini      WMAPTEB.ini
BAO.ini          common.ini              importance_sampling.ini  lowl_old.ini             plik_dx11dr2_DS_v18_TTTEEE.ini  SZ_plus_priors.ini   zre_prior.ini
BAO_RSD.ini      fix_params.ini          JLA.ini                  lowTEB.ini               plik_dx11dr2_HM_v18_EE.ini      WiggleZ_MPK.ini
BK14             getdist_background.ini  JLA_marge.ini            MPK.ini                  plik_dx11dr2_HM_v18_TE.ini      WLHeymans.ini
BK14.ini         getdist_base.ini        lensing_aggressive.ini   outputs                  plik_dx11dr2_HM_v18_TT.ini      WL.ini
BK14only.ini     getdist_common.ini      lensing.ini              params_CMB_defaults.ini  plik_dx11dr2_HM_v18_TTTEEE.ini  WLonlyHeymans.ini
BK14_README.txt  HST_Freeman12.ini       lensonly.ini             pico.ini                 plik_lite_TT.ini                WLonly.ini
BKPlanck         HST_GPE70p6.ini         likelihood.ini           planck_calibration.ini   plik_lite_TTTEEE.ini            WMAP.ini
```

Note that to use Planck likelihoods we need to download, compile it, and link it appropriately (see CosmoMC and Planck)

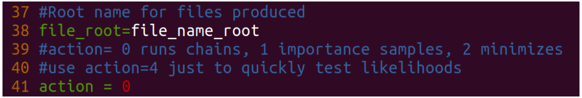
- In the ".ini" file we must include the likelihood; for example:



```
#general settings
#Bicep-Keck-Planck 2015, varying cosmological parameters
#DEFAULT(batch3/BK15.ini)

#DES 1Yr
#DEFAULT(batch3/DES.ini)

#cosmomc_dir = /home/gzacharegkas/cosmomc_test/

#Planck 2015, default just include native likelihoods (others require clik)
DEFAULT(/batch2/plik_dx11dr2_HM_v18_TT.ini)
#DEFAULT(batch2/lowTEB.ini)
DEFAULT(/batch2/lowl.ini)

#Planck 2018
#DEFAULT(batch3/lensing.ini)

#Other Likelihoods
#DEFAULT(batch3/BAO.ini)

#general settings
DEFAULT(/batch3/common.ini)
```

Choose an action

- First we need to specify the ”action”, i.e. what we want to do:



```
#Root name for files produced
file_root=file_name_root
#action= 0 runs chains, 1 importance samples, 2 minimizes
#use action=4 just to quickly test likelihoods
action = 0
```

- Available options are:

(I) ”action=0”: Run chains

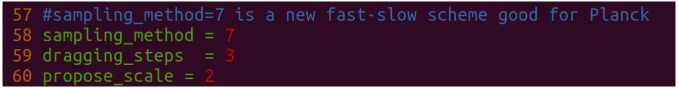
(II) ”action=1”: Post-process the chains (i.e. the ”.dat” files) to importance sample new data, quickly explore effects of new data on already calculated chains by weighting each step appropriately, add corrections to approximate theory, etc

(III) ”action=2”: After the chains are run this finds the maximum posterior

(IV) ”action=4”: Do some quick tests of the likelihoods

Many sampling methods available

- Then, choose a sampling method:



```
#sampling_method=7 is a new fast-slow scheme good for Planck
sampling_method = 7
dragging_steps  = 3
propose_scale = 2
```

- Available sampling methods are:
  - sampling_method=1: Metropolis-Hastings (default)
  - sampling_method=2: Slice sampling
  - sampling_method=3: Fast Slice sampling
  - sampling_method=4: Slow-grid
  - sampling_method=5: Multi-Canonical
  - sampling_method=6: Wang-Landau algorithm
  - sampling_method=7: Fast dragging (good for Planck)

Metropolis-Hastings Algorithm

- We start from data, $\mathbf{d}$, and a model, $M(\boldsymbol{\theta})$, that describes the data and depends on parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_N\}$
- Do a random walk, instead, where in each step from $\boldsymbol{\theta}_i$ we move to $\boldsymbol{\theta}_{i+1}$ at a random direction
- Compute the likelihood at both points (e.g. by running CAMB) and compare them, i.e. compute $r_{i+1|i} \equiv \mathcal{L}(\mathbf{d}|M, \boldsymbol{\theta}_{i+1})/\mathcal{L}(\mathbf{d}|M, \boldsymbol{\theta}_i)$; i.e. for Gaussian likelihood: $-2\ln\mathcal{L} \propto \chi^2 \equiv (\mathbf{d} - M(\boldsymbol{\theta}))^T\Sigma^{-1}(\mathbf{d} - M(\boldsymbol{\theta}))$
- If $r_{i+1|i} \geq 1$ accept $\boldsymbol{\theta}_i$, i.e. add it to the chain
- If $r_{i+1|i} < 1$ do not move to new point, but add old point to the chain again instead (i.e. the multiplicity of that point will increase)
- Choose a new direction and repeat

Convergence of the chains (Part I)

- Usually the Gelman-Rubin statistic is used
- Suppose we have M chains, each with N steps
- Each chain has mean $\bar{\theta}_i$ and variance σ_i $(i = 1, 2, ..., M)$
- The variance within the chains is quantified by:

$$W \equiv \frac{1}{M} \sum_{i=1}^{M} \sigma_i^2 \quad ; \quad \sigma_i^2 = \frac{1}{N-1} \sum_{j=1}^{N} (\theta_{ij} - \bar{\theta}_i)$$

- The variance between the chains will be:

$$B \equiv \frac{N}{M-1} \sum_{i=1}^{M} (\bar{\theta}_i - \bar{\bar{\theta}}) \quad ; \quad \bar{\bar{\theta}} = \frac{1}{M} \sum_{i=1}^{M} \bar{\theta}_i$$

Convergence of the chains (Part II)

- Then, the Gelman-Rubin (R) statistic is defined as:

$$\hat{R} \equiv \sqrt{\frac{\hat{V}}{W}} \;\; ; \;\; \hat{V} \equiv \left(1 - \frac{1}{N}\right) W + \frac{1}{N} B$$

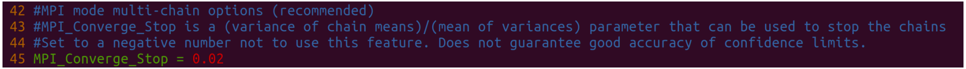
- At the limit of large N, we do not expect a lot of variations to the parameters within of between the chains and $W \approx B$, so that

$$\hat{R} \rightarrow 1$$

- Thus, the convergence criterion requires that for convergence $\hat{R} - 1$ becomes smaller than some value (typically 0.01 or so)
- That's why we need to do a mpi run to check for convergence

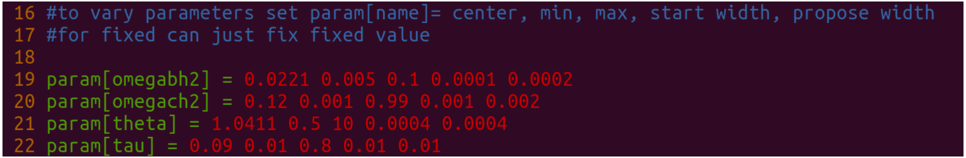
Check for convergence and set parameter values

- We can set the $\hat{R} - 1$ value in the code in "/batch1/common_batch1.ini":



```
#MPI mode multi-chain options (recommended)
#MPI_Converge_Stop is a (variance of chain means)/(mean of variances) parameter that can be used to stop the chains
#Set to a negative number not to use this feature. Does not guarantee good accuracy of confidence limits.
MPI_Converge_Stop = 0.02
```

- To set parameter ranges/values in "/batch1/params_CMB_defaults.ini":



```
#to vary parameters set param[name]= center, min, max, start width, propose width
#for fixed can just fix fixed value

param[omegabh2] = 0.0221 0.005 0.1 0.0001 0.0002
param[omegach2] = 0.12 0.001 0.99 0.001 0.002
param[theta] = 1.0411 0.5 10 0.0004 0.0004
param[tau] = 0.09 0.01 0.8 0.01 0.01
```

CosmoMC base parameters

- The usual ΛCDM parameters:

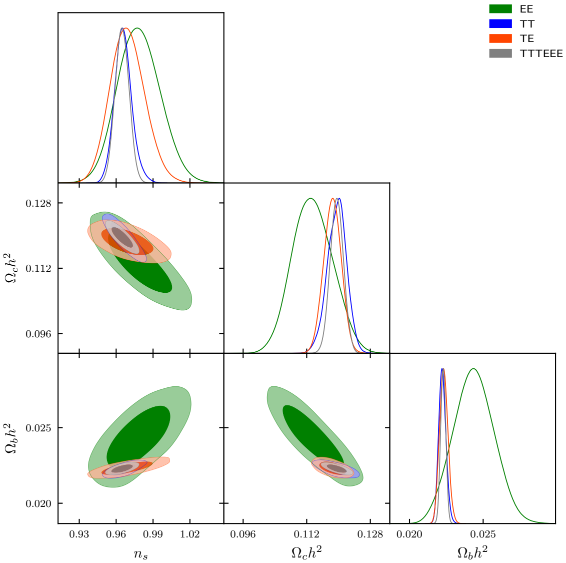
$$\{\Omega_b c^2, \Omega_c h^2, H_0, \tau_{\rm reio}, A_s, n_s\}$$

- More can be added (e.g. if want to test new physics with CAMB)
- Some minimal extensions in CosmoMC are:
  (I) Curvature parameter, Ω_K
  (II) Massive neutrinos, $\sum m_\nu$
  (III) Extra neutrino species, $N_{\rm eff}$
  (IV) Dark Energy equation of state, (w_0, w_a)
  (V) Scalar-to-tensor ratio, $r = A_t/A_s$
  (VI) Running scalar index, $n_{\rm run}$

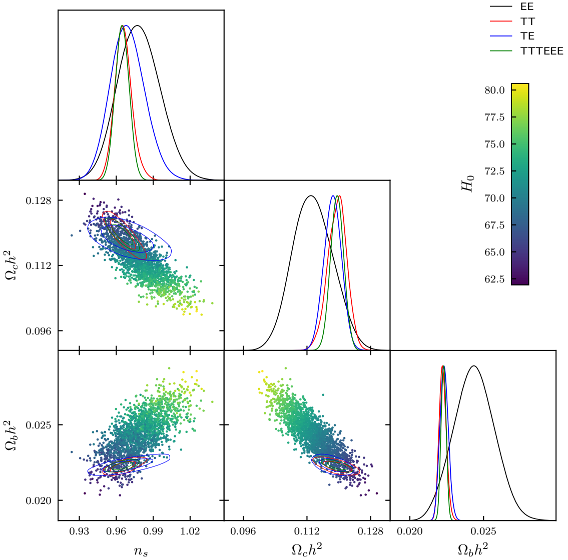
Analysis of chains: What can GetDist give us?

- Covariance matrix which can be used in future runs as input
- Convergence diagnostics
- Best-fit parameters and its likelihood
- Marginalized values of parameters with mean and limits
- Correlation matrix for parameters
- Plots of posterior distributions
  - (I) Marginalized 1D plots
  - (II) Marginalized 2D plots
  - (III) 2D plots colored based on a third parameter
  - (IV) Triangle plots with 1D plots on the diagonal and 2D plots on lower off-diagonal

Example of a triangle plot with GetDist



Example of a ”3D” triangle plot with GetDist



The End