

Tutorial for numerical cosmology

CAMB, CosmoMC, GetDist/Parallel processing

Daniel Boriero

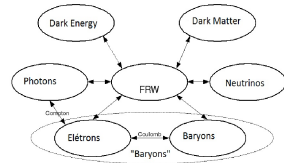
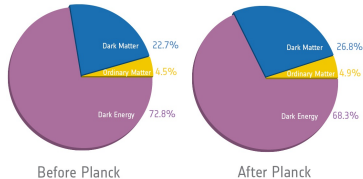
danielb@ifi.unicamp.br



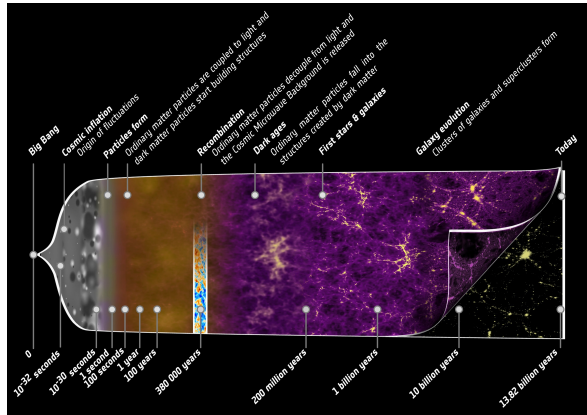
Instituto de Física Gleb Wataghin – UNICAMP, 13083-859, Campinas SP, Brazil

The standard model of cosmology

- Content: Baryons/electrons, photons, neutrinos, dark matter and dark energy



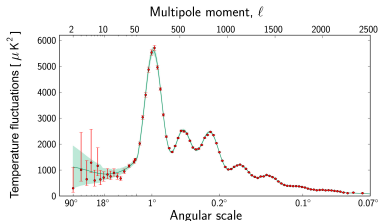
- Evolution: Inflation, particles creation, matter domination, recombination, dark ages, first stars, galaxies evolution, late accelerated expansion



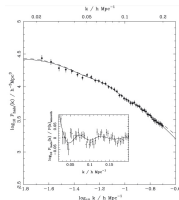
The standard model of cosmology, ...but how do we know?

The model predicts structures that can be tested by a variety of observations.

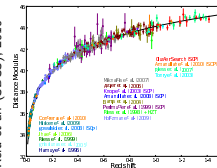
- ▶ Cosmic microwave background (temperature and polarization)
- ▶ Matter power spectrum (galactic, cluster, weak lensing, lyman- α , etc.)
- ▶ Standard markers (candles, rulers, clocks, etc.)



Ade et al. (Planck), 2013



Reid et al. (SDSS), 2010



Suzuki et al. (UNION-2), 2011

The new age of observational cosmology requires precision **numerical tools for theoretical forecast.**

This combination allowed the emergence of precision cosmology as a laboratory for particle physics.

Content of the tutorial of numerical cosmology

► **CAMB**

Boltzmann code for theoretical forecast. Integrates the Boltzmann equations and generates predictions for observables. There are others codes: CMBFast, CMBEasy, CLASS, etc.

► **CosmoMC**

Markov-Chain Monte-Carlo (MCMC) engine for exploring cosmological parameter space. Uses the likelihoods and data made available by the experimental collaborations to find the model that better represents the data.

► **GetDist**

Software that does the statistical analysis of the chains generated by the CosmoMC.

► **Parallel processing**

Independent, but communicating, MCMC going by different paths to reach the same best fit region with velocity and precision.

The choice for this tutorial is the package developed by Antony Lewis and Anthony Challinor.

CAMB: Code for Anisotropies in the Microwave Background

- ▶ Boltzmann code (linear perturbation theory) to solve a realization for a given cosmological model and generate observables to be fitted with data, e.g., the expansion rate of the universe, anisotropy power spectrum of perturbations of the cosmic microwave background and the matter power spectrum.
- ▶ Written in Fortran and based on the CMBFast, which is based on the COSMICS. The most recent version uses Fortran 90 specifications (ex. gfortran, g95).
- ▶ It has a large community and several extensions made by different people.
- ▶ Has online version, useful for fast tests.
- ▶ Contains the standard model (Ω_b , Ω_c , $H_0[\theta]^1$, n_s , A_s , $z_{re}[\tau]^2$) and minimal extensions:
 1. Extra and massive neutrinos: m_n , N_{eff} and mass splitting.
 2. Simple dark energy models: w_0 , w_a , $w(a)$, cs_Λ .
 3. Open, flat and closed models: Ω_k .
 4. Running scalar index, tensor spectral index and tensor to scalar ratio: n_t , n_{run} , $r = A_t/A_s$.
- ▶ Includes: Halofit; scalar, vector and tensor modes; polarization; lensed CMB and lensing potential; internal parallelization of loops; support for adiabatic or isocurvature initial conditions; estimates bispectrum; controllable accuracy

¹ $\theta = 100 \cdot$ (the ratio of the sound horizon to the angular diameter distance)

² τ = the reionization optical depth

CAMB: Code for Anisotropies in the Microwave Background

1. Online version
(http://lambda.gsfc.nasa.gov/toolbox/tb_camb_form.cfm).
2. Download (<http://camb.info/>).
3. After reading the "ReadMe", any further doubts? Visit <http://cosmocoffee.info/>.
4. Installing and compiling. The file "Makefile".
5. The input file "params.ini" and running: `./camb params.ini`
6. The files "modules.f90" and "equations.f90" contain the model parameters and the evolution equations.

CAMB: Code for Anisotropies in the Microwave Background

7. The outputs:

7.1 Spectrum of the cosmic microwave background anisotropies:

7.1.1 root_scalCls.dat: {l CTT CEE CTE [C_{phiphi} C_{phiT}]}

7.1.2 root_lensedCls.dat: {l CTT CEE CBB CTE}

7.1.3 root_lenspotentialCls.dat: {l CTT CEE CBB CTE C_{dd} C_{dT} C_{dE}}

7.1.4 root_tensCls.dat: {l CTT CEE CBB CTE}

7.2 Matter power spectrum: root_matterpower.dat {k P_k}

7.3 Transfer function for all the particle perturbations: root_transfer_out.dat {k/h Delta_CDM/k² Delta_b/k² Delta_g/k² Delta_r/k² Delta_nu/k² Delta_tot/k²}

7.4 Copy of the input file: root_params.ini

7.5 σ_8 , t_0 , θ_A , z_{eq} , $H(z)$: screen and internal variables

8. Modifying for a new physics: modules.f90(type CAMBparams), equations.f90(subs output,derivs), inidriver.F90(main), params.ini (parameters). The case of bulk viscosity for neutrinos.

CosmoMC: Cosmological Monte-Carlo

- ▶ Markov-Chain Monte-Carlo engine for exploring cosmological parameter space.
- ▶ Uses Bayesian inference to select the best model that represents the data.

$$P(\Theta | \mathbf{D}) = P(\mathbf{D} | \Theta) \times P(\Theta)/P(\mathbf{D})$$
$$-2\text{Ln}[P(\mathbf{D} | \Theta)] = \chi^2 = \sum (\Theta - \mathbf{D})^2 / \sigma^2 .$$

- ▶ Fully integrated with CAMB, which is called to give the theoretical prediction for the data being tested.
- ▶ Contains likelihoods and data for the most simple datasets (SNIa, HST, SDSS-BAO, 6dF-BAO, SDSS-DR4).
- ▶ More complex datasets (WMAP, Planck, SPT/ACT) must be installed separately. However, most collaborations make available CosmoMC extensions with the likelihood functions, their data and full descriptions of how to implement the extension.
- ▶ Compiles with Fortran 2003 specifications (only IFort 2013).
- ▶ Can be used with different applications (dark matter direct search, neutrino oscillations, etc.).
- ▶ Contains: covariance matrix for the best fit multi-dimensional region; post-processing capability for theory/likelihood correction or data inclusion; special technique to find the best fit (least χ^2); check-point feature; parallelization implemented

CosmoMC: Cosmological Monte-Carlo

1. Download (<http://cosmologist.info/cosmomc/>).
2. After reading the "ReadMe", any further doubts? Visit <http://cosmocoffee.info/>.
3. Installing and compiling. The file "Makefile". Two modes of operation: serial and parallel.
4. The input file "test.ini" and running: `./cosmomc test.ini`
The ranges are defined as: "param[paramname] = center min max start_width propose_width".
5. Modes of execution: action = 0 (MCMC), action=1 (postprocess), action=2 (find best fit point only)
6. "cmbdata.f90", "mpk.f90", etc. reads the data. The file "MCMC.f90" sort the parameters and calls the "calc.f90", which calls "params_CMB.f90" to read the parameters sorted for the CAMB, then calls "CMB_Cls.simple.f90" to do the parsing for CAMB and obtain the result, and then finally calculates the likelihood writing it in files. The files "cmbtypes.f90" contains the model parameters. The file "calclike.f90" calls different files to calculate each likelihoods selected ("BAO.f90", "HST.f90", etc.).

CosmoMC: Cosmological Monte-Carlo

7. The outputs for MCMC:

- 7.1 Chains of χ^2 for realizations (specific values for each parameter), along with log and check-point: root.txt, root.log, root.chk. The chains has the format: "weight like param1 param2 param3 ...", where "like" is $-\log(\text{likelihood})$ and "weight" is the of realizations in the same region.
- 7.2 For multiple chains, the above files will have a number related with each chain: root_1.txt, root_1.log, root_1.chk, root_2.txt, root_2.log, root_2.chk, etc. In this case, there will be also the file "root.converge_stat" containing a useful convergence statistic in runtime.
- 7.3 Copy of the list of parameters, input file and ranges assumed for each parameter: root.paramnames, root.inputparams, root.ranges
- 7.4 Reduced binary data (indeep_sample=10) for later fast post-processing: root.data
- 7.5 For post-processing must be taken care to not replace files with the same name.

8. The outputs for post-processing: only root.txt and root.data, as above.

9. The outputs for best fit:

- 9.1 The CI's in the best fit point, the least χ^2 (total and by dataset) and a copy of the input file: root.bestfit_cl, root.minimum, root.minimum.inputparams
- 9.2 A covariance matrix, if asked (estimate_propose_matrix=T): root.covmat

CosmoMC: Cosmological Monte-Carlo

10. In serial and MCMC mode, the chain will stop only with the number of samples specified. In parallel, it may be stopped when all the chains agree with the best fit point, under some convergence acceptance (the Gelman and Rubin R condition³).
11. The best fit runs only in serial and will stop when it reaches some radius (`max_like_radius = 0.01`) around the propose width for each parameter.
12. Sampling method (`sampling_method`): 1-Metropolis, 7-dragging algorithm (Planck). It is possible to start with slicing, for unknown parameter space. In this case, I would suggest to run in best fit mode (`estimate_propose_matrix=T`) and then restart with the conventional MCMC method with the covariance matrix.
13. Modifying for a new physics (new parameter):
 - 13.1 modify CAMB first,
 - 13.2 include parameter in "cmbtypes.f90",
 - 13.3 read from MCMC in "params_CMB.f90" and change the number of parameters when calling the function "SetTheoryParameterNumbers",
 - 13.4 do the parsing in "CMB_Cls_simple.f90" for the CAMB,
 - 13.5 include the name in "params_CMB.paramnames" and the range in "params_CMB_defaults.ini"
14. Run cosmomc as a generic MCMC. The flavour oscillation case.

³(variance of chain means/mean of chain variances)=MPI_Converge_Stop<0.02

GetDist

- ▶ Software to analyse the chains produced by the CosmoMC. However, it may be used totally independent of CosmoMC and even replaced for a different analyser.
1. The code comes together with the CosmoMC package. Its source is "GetDist.f90" and is generated with: "make getdist" or "make all".
 2. The file "distparams.ini", running: `./getdist distparams.ini`
 3. Getdist needs only the files with the chains generated by the CosmoMC, the "root.txt".
 4. If multiple chains, will stop in the chain with the least number of χ^2 computed.
 5. Remember to specify the cuts for parameters with sharp edged distribution (e.g., m_ν, r): `limits[mnu]= 0 N`

GetDist

6. Output text files (analysis):

- 6.1 root.margestats - the mean, limits and stddev from the 1D marginalized distributions
- 6.2 root.likestats - the best fit, and limits from the N-D confidence region
- 6.3 root.covmat - covariance matrix for parameters
- 6.4 root.corr - covariance matrix for parameters normalized to have unit variance
- 6.5 root.PCA - human-readable file giving details of PCA, constrained parameters
- 6.6 root.converge - Statistics to help assess chain convergence/sampling error
- 6.7 root_thin.txt - thinned merged versions of input files (thin_factor=10)

GetDist

7. Output plot files:

7.1 root.m - MatLab .m file to produce 1D marginalized plots

7.2 root_2D.m - MatLab .m file to produce 2D marginalized plots

7.3 root_3D.m - MatLab .m file to produce 2D sample plots colored by a third parameter (needs make_single_samples=T and single_thin=4)

7.4 root_tri.m - MatLab .m file to produce triangle plots (1D on diagonal, 2D off-diagonal)

7.5 Easy to hack and write more refined plots. The case of dataset superposition.

7.6 Sources for the plot files:

7.6.1 Distribution for each parameter: root_p_"param".dat and root_p_"param".likes

7.6.2 Correlated distribution for two parameters: root_2D_"param1"_"param2"_"*

7.6.3 Parameter names used by matlab: root.paramnames

7.6.4 Single files for 3D plots: root_single.txt

The plot files may be generated in Python, which is open source. However, the open source Octave (Linux) runs matlab scripts with minor corrections.

Parallel Processing

- ▶ Parallel processing improves the velocity and reliability in finding the best fit region.
- ▶ Takes advantage of the powerful environment available in clusters.
- ▶ Brazil has public clusters:
 1. CENAPAD-SP (<http://www.cenapad.unicamp.br/>)
 2. IAG (<https://lai.iag.usp.br/>)
 3. CENAPAD-MG (<http://www.cenapad.ufmg.br/>)
 4. ...
- ▶ A highly dimensioned computer (16 cores, e.g.) may work very well for one job at a time. However, clusters are much more stable and customized for parallel processing, besides they provide access to several jobs at the same time, technical support and external access.
- ▶ Because each environment is unique and the installation may be very cumbersome, I suggest to install in the place that you expect run the jobs. Test in simpler computers could be useless.

Parallel Processing

1. The software must be already parallelized, i.e., the intended parallel routines must be implemented using parallel parsing.
2. Find out which implementation is running in the cluster you intend to use. For Intel architecture, give preference to MPI.
3. For clusters, usually there are instructions to implement the parallelization, use it.
4. The line of compilation is very specific for each environment. The Intel help us with an advisor to be used with IFort: <http://software.intel.com/en-us/articles/intel-mkl-link-line-advisor/>
5. The Makefile for parallel compilation.
6. The PBS commands: qsub, qstat, qdel.
7. The output files contain the usual screen output of the software running, in our case, the CosmoMC
 - 7.1 The display file: job.out
 - 7.2 The error file: job.err
8. Enjoy at fullest the potential of clusters.

References

► Original papers:

- <http://arxiv.org/abs/astro-ph/9911177> (CAMB)
- <http://arxiv.org/abs/astro-ph/0205436> (COSMOMC)
- <http://arxiv.org/abs/astro-ph/9506072> (Boltzmann eqs. for cosmology)
- <http://arxiv.org/abs/astro-ph/0203507> (Boltzmann eqs. for numerical cosmology)

► Interesting pages and tutorials:

- <http://lambda.gsfc.nasa.gov/> (Nasa Repository)
- <http://cosmologist.info/> (Repository of the code's creator, contains several links, tutorials and presentations)
- <http://camb.info/readme.html> (CAMB homepage)
- <http://cosmologist.info/cosmomc/readme.html> (COSMOMC homepage)
- <http://cosmocoffee.info/> (Forum)
- <http://background.uchicago.edu/~whu/> (Wayne Hu homepage)
- <https://registrationcenter.intel.com/RegCenter/NComForm.aspx?ProductID=1523> (IFort)
- <http://software.intel.com/en-us/articles/intel-mkl-link-line-advisor/> (Advisor for MPI)

Final remarks

- ▶ The expectation of better experimental sensitivity must be followed by more reliable theoretical predictions.
- ▶ Corrections based in fiducial N-body simulations may systematically correct the predictions made with perturbative theory. However, complete and broad n-body simulations may be needed to distinguish subdominant effects from different origins and all located at non-linear scales.
- ▶ However, future long-distance measurements, such as Weak Lensing, may collect information at small scale which is still in the linear regime of perturbations. This promising data will keep perturbative theory in the frontline for, at least, the next decade.
- ▶ To work with phenomenological cosmology, it is fundamental to master tools for statistical analysis of data.
- ▶ The Planck legacy will drive the theoretical effort in the next years. The package developed by Antony Lewis is still the main platform, already integrated to take fully advantage of the Planck data.