

Kavli IPMU Workshop on Tools for Cosmology: The CLASS and Monte Python codes

The topics covered by this workshop will be:

Topic I: the simulation of cosmological perturbations in the universe, and the computation of the CMB anisotropy and Large Scale Structure power spectra. In 17 sessions of 45 minutes, we will present:

- a very brief summary on the underlying theory,
- its numerical implementation in the Cosmic Linear Anisotropy Solving System (CLASS, <http://class-code.net>), written in C,
- hands-on exercises dedicated to the practical use of the code,
- more advanced exercises to learn how to modify the code.

For this part, most of the lectures will be structured in the same way as the modules of the CLASS code:

- | | |
|--|--------------------------------|
| 1. input parameters | 6. non-linear corrections |
| 2. homogeneous background | 7. harmonic transfer functions |
| 3. thermodynamical evolution | 8. power spectra |
| 4. cosmological perturbations | 9. CMB lensing |
| 5. primordial spectra, initial conditions, inflation | 10. output of the code |

We will spend most of the time on the first modules, containing most of the physics, and that users may need to modify. The modules 6-10 will be quickly summarised during the last session.

In parallel, there will be lectures focused on specific topics, either numerical (plotting, error management, advanced solvers for differential equations, python wrapper, automatic testing) or physical (implementation of massive neutrinos, decaying dark matter).

Topic II: the extraction of cosmological parameters from observational data, using a Monte Carlo algorithm. In 6 sessions of 45 minutes, we will present:

- briefly, the principle of several algorithms,
- their implementation in Monte Python (<http://montepython.net>), a code written in Python, interfaced with CLASS and with many likelihoods accounting for cosmological data,
- hands-on exercises dedicated to the basic usage of the code, and how to visualise results,
- more advanced exercises to learn how to introduce one's own likelihood.

Prerequisites:

- some basic knowledge of cosmology. The lectures will be focused on the use of the codes and on numerical aspects, with little time available for theoretical aspects. You will follow better if you have some knowledge on topics like: cosmological perturbation theory, scalar and tensor modes, gauge freedom, initial conditions, acoustic oscillations, CMB temperature and polarisation power spectra, matter power spectrum, weak lensing. You can refresh your mind with the courses and textbooks of your choice. To keep close to the notations of the lecturer, you may *optionally* read the TASI lecture notes of J. Lesgourgues, arXiv:1302.4640; or section 2 and 5.1 of "Neutrino Cosmology", CUP 2013, by J. Lesgourgues, G. Mangano, G. Miele and S. Pastor.
- prior knowledge of C would help. C++ users automatically know C. Fortran users will find C easy. To check your level or refresh your mind on C, **please take a bit of time to read our mini-C-tutorial**. You can find it in the Dropbox folder, it is called `c-samples.pdf`. It contains a few example of short C codes, using pointers, structures and macros in a way similar to CLASS.

- prior knowledge of Python would help for the part related to advanced usage of Monte Python (structure of the code, creating new likelihoods), presented on day IV and V. For the basic usage of the code presented in the first three days, it is not even necessary to know Python. People used to Matlab will find Python rather similar in spirit. If you know Python, you will also be able to solve most CLASS exercises with Python commands, instead of typing commands in your terminal.

Equipment:

Please bring your laptop during the sessions. You will use it during both lecture and exercise time. Concerning your operating system:

- if your laptop works under linux, installing the codes should be totally straightforward. In the worst case, you will have to update your C or python versions, and upload a few python modules, but this is always quick and easy.
- if you have a Mac, it will be nearly as easy, with just a few subtleties (especially with OS 10.9 Mavericks). Indeed, with OSX, it is sometimes tricky to get some versions of C and python that are both recent and compatible with each other. However we know some easy fixes and we will provide help. See the next section for more details.
- while in principle one can compile CLASS on Windows, in practice it is more difficult. You would need a C-compiler, GNU Make for building, and you need to modify the Makefile to make it work in the Windows command shell. If you are running Windows, we recommend either running Virtualbox with a standard Linux distribution or simply connecting to a Linux cluster using SSH.

Installation:

- We will assume that everybody has CLASS v2.4.1. The code can be downloaded from the web page <http://class-code.net>

You can also download class from the public git repository, at

https://github.com/lesgourg/class_public/releases

by clicking v2.4.1 (from our point of view, both ways of downloading are equivalent).

The code compilation is very straightforward (just need to type `make` in the class directory), unless you have a prehistoric or unusual version of C (for instance, all gcc versions 4.2 or higher will work; you have this by default on your linux/windows/Mac computer, unless you use MacOS10.9 Mavericks, which switched to a very unusual C compiler! If you are on Mavericks, you either need to download a standard gcc compiler (recommended), or to switch off the `-fopenmp` option in the Makefile before compiling (but then the code will be slower)).

More details on unpacking the downloaded files and compiling the code can be found on the top of the CLASS webpage, in the brief section called Documentation, and also in the more detailed wiki page https://github.com/lesgourg/class_public/wiki/Installation

- We will also assume that everybody has Monte Python v2.1.0. The code can be downloaded from the public GitHub repository https://github.com/audren/montepython_public/releases, by clicking on the link to the .zip or tar.gz package of version 2.1.0.

Before using Monte Python, you must check that you have the correct version of python, with the necessary modules installed, and you must also compile the python wrapper of the CLASS code. All these steps are explained on the wiki https://github.com/audren/montepython_public/wiki/Installation

The most tricky step is the compilation of the CLASS wrapper, which can be delicate on MacOS10.9 Mavericks (again!). If despite of the wiki, you cannot solve some problems, do not hesitate to write to benjamin.audren@epfl.ch

Detailed schedule:

The schedule is presented on the next page. Everyday, we will have 5 sessions of 45 minutes each. Each session will be a mix of lecturing time and exercise time. The schedule shows the main speaker for each session. However, during exercises, the three lecturers will be available for help.

On top of that, everyday between 17:45 and 16:15, there will be a question/answer session, during which participants can get help for installation issues, catch up with unfinished exercises, or ask more specific questions

on the codes and on the underlying theory.

DAY I : Monday 27th October

09:30-10:15	CLASS	History, goals & philosophy.	JL
10:15-11:00	General	Bayesian parameter extraction.	BA
Coffee			
11:30-12:15	CLASS	Overall style and structure.	JL
Lunch			
13:30-14:15	CLASS	Input and output files. Basic running.	JL
14:15-15:00	CLASS	How to visualise the output.	TT
Tea			
15:45-16:30	Optional	Lecturers will answer questions and provide help on exercises	

DAY II : Tuesday 28th October

09:30-10:15	CLASS	Dynamical indexing and error management.	JL
10:15-11:00	CLASS	How the input module works.	TT
Coffee			
11:30-12:15	CLASS	The python wrapper classy.py	TT
Lunch			
13:30-14:15	MontePython	Basic runs.	BA
14:15-15:00	MontePython	Analyzing runs.	BA
Tea			
15:45-16:30	Optional	Lecturers will answer questions and provide help on exercises	

DAY III : Wednesday 29th October

09:30-10:15	CLASS	The background module.	JL
10:15-11:00	CLASS	Playing with the background module.	JL
Coffee			
11:00-11:45	General	Git repositories.	BA
Lunch			
13:30-14:15	CLASS	Introducing new physics in the background.	JL
14:15-15:00	MontePython	All running and plotting options.	BA
Tea			
15:45-16:30	Optional	Lecturers will answer questions and provide help on exercises	

DAY IV : Thursday 30th October

09:30-10:15	CLASS	The thermodynamics module.	JL
10:15-11:00	MontePython	Internal structure of the code.	BA
Coffee			
11:30-12:15	CLASS	The perturbation module.	JL
Lunch			
13:30-14:15	CLASS	Playing with perturbations.	JL
14:15-15:00	General	Advanced ODE solvers. ndf15.	TT
Tea			
15:45-16:30	Optional	Lecturers will answer questions and provide help on exercises	

DAY V : Friday 31th October

09:30-10:15	CLASS	The primordial module.	JL
10:15-11:00	CLASS	Implementation of non-cold dark matter.	TT
Coffee			
11:30-12:15	MontePython	Writing your own likelihood.	BA
Lunch			
13:30-14:15	CLASS	Last modules.	JL
14:15-15:00	CLASS	Automatic testing. Implementation of new physics: a full example.	TT
		Projects and Prospects.	JL
Tea			
15:45-16:30	Optional	Lecturers will answer questions and provide help on exercises	

Speakers : BA = Benjamin Audren, JL = Julien Lesgourgues, TT = Thomas Tram