

CLASS

Cosmological Linear Anisotropy Solving System

Markus Mosbech
Institute for Theoretical Particle Physics and Cosmology, RWTH Aachen
University

Les Karellis, France, 17-30Aug 2025

Visit https://lesgourg.github.io/class_public/class.html for more
info!

class in Les Karellis

What to expect in this first lecture:

- Basics: Why use `class`?
- Usage: Installation
- Usage: Python Interface
- Usage: Samplers
- Basics: Existing Species
- Basics: Module Overview

We will learn how to use `class` and which models can be run with it.

What is an Einstein-Boltzmann solver?

Often just called a *Boltzmann code* for brevity, a typical Boltzmann code will:

- Solve coupled Einstein and Boltzmann equations.
- Generally work at linear level in perturbation theory.
- Compute global (Background+Thermodynamic) quantities *and* perturbations.

$$\underbrace{G_{\mu\nu} = 8\pi T_{\mu\nu}}_{\text{Einstein-equation}} \quad \underbrace{\frac{df}{d\lambda} = C[f]}_{\text{Boltzmann-equation}} \quad (1)$$

Why use a Boltzmann code?

Modern Boltzmann codes offer:

- History of the universe at the global level ($H(z)$, $\rho_i(z)$, etc.)

Why use a Boltzmann code?

Modern Boltzmann codes offer:

- History of the universe at the global level ($H(z)$, $\rho_i(z)$, etc.)
- Thermal history of the universe ($T_b(z)$, x_e , τ , etc.)

Why use a Boltzmann code?

Modern Boltzmann codes offer:

- History of the universe at the global level ($H(z)$, $\rho_i(z)$, etc.)
- Thermal history of the universe ($T_b(z)$, x_e , τ , etc.)
- Evolution of (linear) perturbations (δ_i , θ_i , ψ , ϕ , etc.)

Why use a Boltzmann code?

Modern Boltzmann codes offer:

- History of the universe at the global level ($H(z)$, $\rho_i(z)$, etc.)
- Thermal history of the universe ($T_b(z)$, x_e , τ , etc.)
- Evolution of (linear) perturbations (δ_i , θ_i , ψ , ϕ , etc.)
- Fourier space transfer functions ($T(k)$)

Why use a Boltzmann code?

Modern Boltzmann codes offer:

- History of the universe at the global level ($H(z)$, $\rho_i(z)$, etc.)
- Thermal history of the universe ($T_b(z)$, x_e , τ , etc.)
- Evolution of (linear) perturbations (δ_i , θ_i , ψ , ϕ , etc.)
- Fourier space transfer functions ($T(k)$)
- CMB spectra, both lensed and unlensed (C_ℓ^{TT} , C_ℓ^{TE} , C_ℓ^{EE} , C_ℓ^{BB})

Why use a Boltzmann code?

Modern Boltzmann codes offer:

- History of the universe at the global level ($H(z)$, $\rho_i(z)$, etc.)
- Thermal history of the universe ($T_b(z)$, x_e , τ , etc.)
- Evolution of (linear) perturbations (δ_i , θ_i , ψ , ϕ , etc.)
- Fourier space transfer functions ($T(k)$)
- CMB spectra, both lensed and unlensed ($C_\ell^{TT}, C_\ell^{TE}, C_\ell^{EE}, C_\ell^{BB}$)
- Linear matter power spectrum, galaxy counts, cosmic shear ($\xi^\pm, C_\ell^{dd}, P_{\text{lin}}(k)$)

Why use a Boltzmann code?

Modern Boltzmann codes offer:

- History of the universe at the global level ($H(z)$, $\rho_i(z)$, etc.)
- Thermal history of the universe ($T_b(z)$, x_e , τ , etc.)
- Evolution of (linear) perturbations (δ_i , θ_i , ψ , ϕ , etc.)
- Fourier space transfer functions ($T(k)$)
- CMB spectra, both lensed and unlensed ($C_\ell^{TT}, C_\ell^{TE}, C_\ell^{EE}, C_\ell^{BB}$)
- Linear matter power spectrum, galaxy counts, cosmic shear ($\xi^\pm, C_\ell^{dd}, P_{\text{lin}}(k)$)
- Emulated non-linear power spectra

Why use a Boltzmann code?

Modern Boltzmann codes offer:

- History of the universe at the global level ($H(z)$, $\rho_i(z)$, etc.)
- Thermal history of the universe ($T_b(z)$, x_e , τ , etc.)
- Evolution of (linear) perturbations (δ_i , θ_i , ψ , ϕ , etc.)
- Fourier space transfer functions ($T(k)$)
- CMB spectra, both lensed and unlensed ($C_\ell^{TT}, C_\ell^{TE}, C_\ell^{EE}, C_\ell^{BB}$)
- Linear matter power spectrum, galaxy counts, cosmic shear ($\xi^\pm, C_\ell^{dd}, P_{\text{lin}}(k)$)
- Emulated non-linear power spectra
- CMB spectral distortions

Why use a Boltzmann code?

Modern Boltzmann codes offer:

- History of the universe at the global level ($H(z)$, $\rho_i(z)$, etc.)
- Thermal history of the universe ($T_b(z)$, x_e , τ , etc.)
- Evolution of (linear) perturbations (δ_i , θ_i , ψ , ϕ , etc.)
- Fourier space transfer functions ($T(k)$)
- CMB spectra, both lensed and unlensed ($C_\ell^{TT}, C_\ell^{TE}, C_\ell^{EE}, C_\ell^{BB}$)
- Linear matter power spectrum, galaxy counts, cosmic shear ($\xi^\pm, C_\ell^{dd}, P_{\text{lin}}(k)$)
- Emulated non-linear power spectra
- CMB spectral distortions

All computed in a matter of seconds!

Why use a Boltzmann code?

This has several use cases:

- Analysis of CMB experiments
- Analysis of LSS experiments
- Initial conditions for non-linear simulations (N -body, etc.)
- Consistent treatment of background/thermodynamic evolution

All easy to do with `class`!

Fast execution \Rightarrow ideal for use in an MCMC pipeline.

Why use class?

class is:

- Accurate: class & camb cross-check each other
- Versatile: Interfaces with MontePython, Cobaya, Cosmosis, Procoli, CosmoPower, OLÉ, CONNECT, and others!
- Comprehensive: Computes a wide range of cosmological observables for a large selection of models beyond Λ CDM.
- Modular and well-documented: ReadTheDocs page and Doxygen documentation, thoroughly commented source code, easy to modify

All strong arguments to use **class**!

Installing class

Using class

If you have no intention of modifying source code:

```
> pip install classy
```

And the class wrapper will be ready to use in your Python environment.

This is the easiest way to install.

Modifying class

If you wish to modify source code:

```
> git clone git@github.com:
    lesgourg/class_public.
    git class
> cd class/
> make clean; make -j
```

The wrapper can be used in your Python environment, and the binary executable can be called from the terminal.

Documentation

1 Basic information and links:

Documentation

- 1 Basic information and links:
 - in the historical `class` webpage <http://class-code.net>

Documentation

- 1 Basic information and links:
 - in the historical `class` webpage <http://class-code.net>
 - in the pdf manual in `doc/manual/CLASS_MANUAL.pdf`

Documentation

1 Basic information and links:

- in the historical `class` webpage <http://class-code.net>
- in the pdf manual in `doc/manual/CLASS_MANUAL.pdf`
- the online documentation page (from the previous page, or from https://github.com/lesgourg/class_public/wiki, click on the link `online html documentation`)
- First three subsections:
 - Installation instructions
 - References to many papers for the physics
 - General overview (architecture, input/output, general principles)

Documentation

1 Basic information and links:

- in the historical `class` webpage <http://class-code.net>
- in the pdf manual in `doc/manual/CLASS_MANUAL.pdf`
- the online documentation page (from the previous page, or from https://github.com/lesgourg/class_public/wiki, click on the link `online html documentation`)
- First three subsections:
 - Installation instructions
 - References to many papers for the physics
 - General overview (architecture, input/output, general principles)

2 More advanced:

- Old course notes from previous years on <https://schoeneberg.github.io/> under “Resources”
- several detailed courses on Julien’s course webpage <https://lesgourg.github.io/courses.html>, especially the courses from Tokyo and NYC
- Full auto-generated documentation with dependence tree.

The code structure