Numerical Codes for Dark Matter research

Sabarish

RWTH Aachen University

sabarish.venkataramani@rwth-aachen.de

2022-01-06

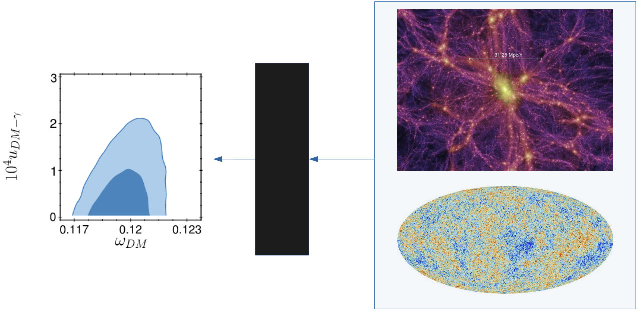
Computational Methods

- Analytic solutions almost never exist for realistic models.
- Translation of theories into observables.
- Evolution of complex systems
    1. N-Body simulations are literally playing god, e.g. you can observe phase transitions in gases.
    2. Genetic algorithms that simulate evolution, spread of diseases, etc.
- Pedagogical reasons.
- Cellular automata.

Motivation

Two Scenarios :

1. Constraining DM parameters



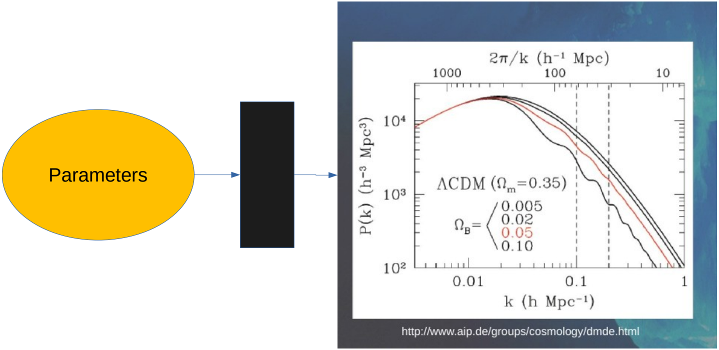
[1]

---

[1] Niklas Becker et al. (2020) [1]

Motivation

2. A plot from the course on DM,



- To understand the black box, let us start with a short introduction to Linear Perturbation theory !

Linear Perturbation Theory

Perturbation theory

- Universe deviates from ideal homogeneous, isotropic FLRW universe.
- Introduce perturbations in the formalism.
- Decompose : $\text{Universe} = \underbrace{\text{Background}}_{\text{Friedmannian}} + \underbrace{\text{Perturbations}}_{\text{Deviations}}$
- Decompose : Both Metric and the source

Example : CMB Anisotropies

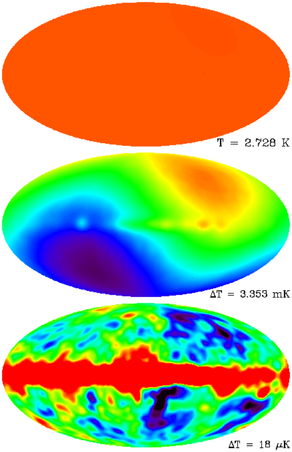


Figure: Credits : aether.lbl.gov

Metric Perturbation

- FLRW or Background metric,

$$g_{\alpha\beta}^{FLRW} = a^2(\tau)\eta_{\alpha\beta}$$

- The perturbed FLRW metric in Newtonian Gauge is,

$$g_{\alpha\beta} = a^2(\tau)(\eta_{\alpha\beta} + h_{\alpha\beta})$$

$h_{00}(\mathbf{x}, t) = 2\psi(\mathbf{x}, t)$	$h_{0i} = 0$	$h_{ij} = 2\phi(\mathbf{x}, t)\delta_{ij}$

Matter Perturbation

- Energy density : $\rho_i(\mathbf{x}, t) = \bar{\rho}_i(t) + \delta\rho_i(\mathbf{x}, t)$

- $\bar{\rho}_i(t)$ obeys the usual Friedmannian evolution, that is,

$$\bar{\rho}_m(t) \propto a^{-3}$$

- Density contrast : $\delta_i(\mathbf{x}, t) := \delta\rho_i(\mathbf{x}, t)/\bar{\rho}_i(t)$

Evolution

Basic Equations

Boltzmann equation

- It gives the evolution of phase space distribution of particles.
- Collisionless Boltzmann equation :

$$N \propto \int \mathrm{d}^3\mathbf{x}\mathrm{d}^3\mathbf{p} \;\; f(\mathbf{x}, \mathbf{p}, t) \implies \frac{\mathrm{d}f}{\mathrm{d}t} = 0$$

- With interactions :

$$\frac{\mathrm{d}f}{\mathrm{d}t} = C[f]$$

Einstein equation

- Relation between Matter and Metric

$$G_{\mu\nu} = 8\pi G T_{\mu\nu}$$

Einstein + Boltzmann system

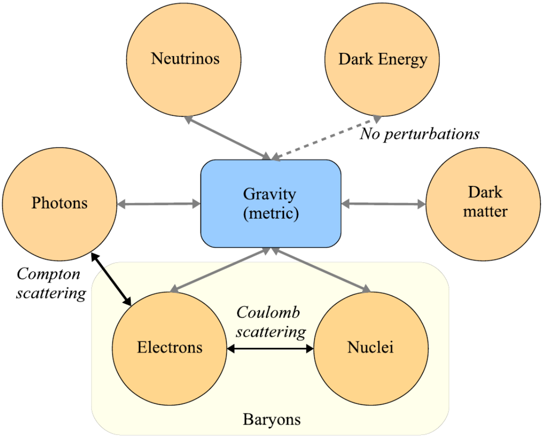


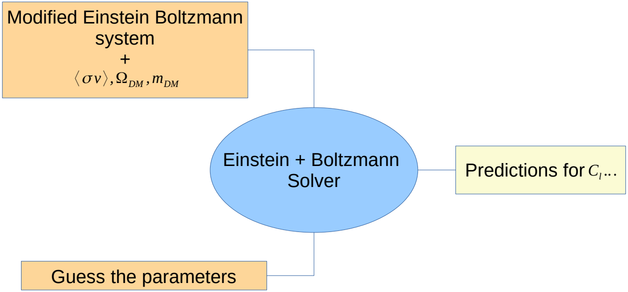
Figure: Credits : Modern Cosmology, Dodelson

- Black box solves this system of DE's.

The Black Box

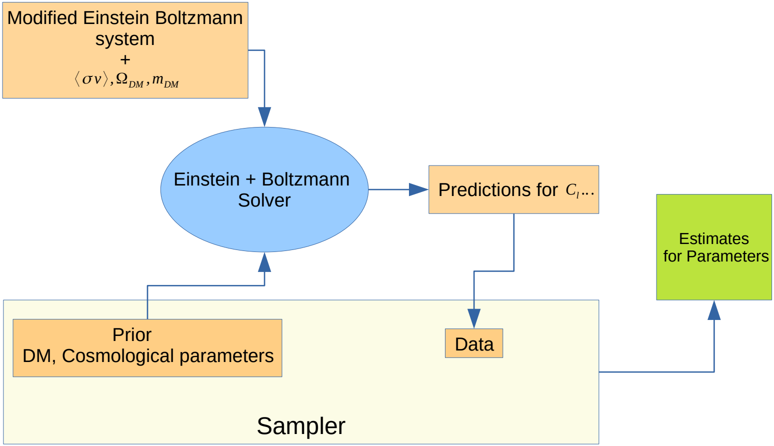
Einstein Boltzmann Solver

- Adding a new species into the game ?



- Compare with data, and change guess parameters.
- Can we keep guessing ? Yes.
- Elegant solution : Interface EBS with a Sampler

Samplers



- E.g. Montypython, Cobaya.
- They explore the parameter space using MCMC methods.

Einstein Boltzmann Solvers

CLASS

History of Boltzmann solvers

There have been a lot of Boltzmann solvers in the past, here is a small history.

Package	Language	Year	Authors
COSMICS	f77	1995	Bertschinger
CMBFAST	f77	1996	Seljak & Zaldarriaga
CAMB	f90	1999	Lewis & Challinor
CMBEASY	C++	2003	Doran
CLASS	C	2011	J.Lesgourgues & Tram
PyCosmo	Py + C	2018	Alexandre Refregier et al.

- CAMB, and CLASS are the most widely used ones.
- CLASS is a well structured, and it is *relatively* easy to edit the code to add new species + Python wrapper.

CLASS - Overview

- It is written in C, it has a python wrapper.
- Environment : Terminal or Python scripts/notebooks.

What can CLASS compute ?

- Name it, you get it !
- Age, Background evolution, $d_L(z)$, Perturbation functions, Power Spectrum, Harmonic spectra, etc.

What can CLASS already deal with ? Examples :

- Photons, Baryons, Massless/Massive neutrinos, Λ.
- CDM, WDM, Dark Radiation, Decaying DM.
- Dark Energy with different equation of state parameter.

Structure of CLASS !

Non-Perturbative calculation

Input parameters : $(H_0, \Omega_i, \ldots)$

Step 1 : Background

- Solve Friedmann equation $\implies a(t) \implies H(t)$
- $z \leftrightarrow a \leftrightarrow t \leftrightarrow \tau$
- Other quantities : Age, Size, $\bar{\rho}_i(t)$,Luminosity distance, etc.

Structure of CLASS !

Non-Perturbative calculation

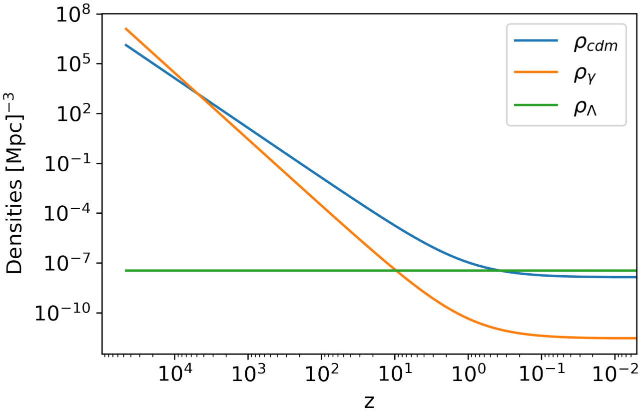


Figure: Evolution of Background densities

- Observe : Transitions from RD $\rightarrow$ MD and etc.

Structure of CLASS !

Non-Perturbative calculation

Step 2 : Thermodynamics

Thermodynamic quantities like $x_e(t), \bar{T}_b(t), t_{dec} \leftrightarrow z_{dec}$

Example : Photon decoupling

- Decoupling starts $\Gamma \sim H$
- $\Gamma \propto x_e$, and x_e is obtained by solving Boltzmann equation using external codes like HyRec2020 or RecfastClass.
- $\Gamma \sim H$ is solved to get $t_{dec} \leftrightarrow z_{dec}$

Structure of CLASS !

Non-Perturbative calculation

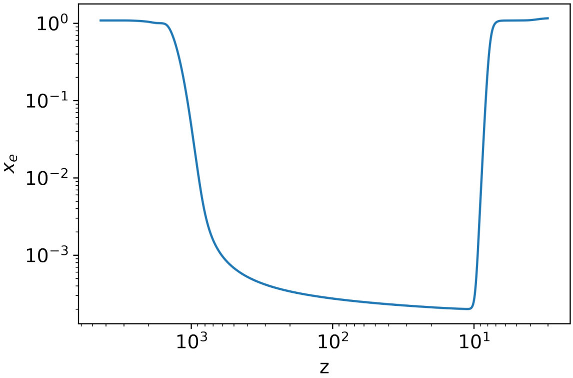


Figure: Free electron fraction $x_e(z)$

- Observe : Decoupling, and reionization.

Structure of CLASS !

Perturbative calculation

Step 3 : Inflation

Generate primordial power spectrum, in other words initial conditions for perturbations.

Step 4 : Perturbations

- Obtain the evolution of $\boxed{\psi, \phi, \delta_b, \delta_c, \delta_\gamma \ldots}$ as functions of k, τ from the initial conditions from previous step.
- Equations can be found $\rightarrow$ Ma, Bertschinger (1995) [2]
- Visualization tool : Perturbations in 2D real-space.

Structure of CLASS !

Perturbative calculation

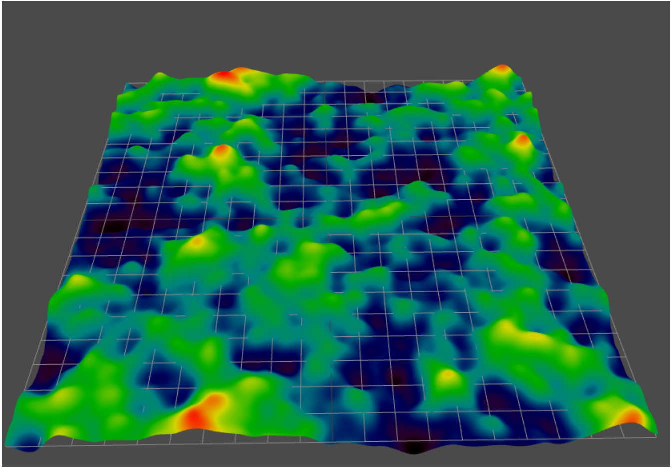


Figure: Real Space visualization of fluctuations of δ_γ

Structure of CLASS !

Perturbative calculation

Step 5 : Matter Power spectrum

- From previous step, $\delta_m = \delta_b + \delta_c$, therefore CLASS computes $P_m(k, z)$

- Matter power spectrum :

$$\left\langle \delta_m(z, \vec{k})\delta_m(z, \vec{k'}) \right\rangle \stackrel{!}{=} 2\pi\delta_D^3(\vec{k} - \vec{k'}){\color{red}P_m(z, k)}$$

- CLASS can access external codes like halofit, HMCODE, to get non-linear Matter power spectrum.

Structure of CLASS !

Perturbative calculation

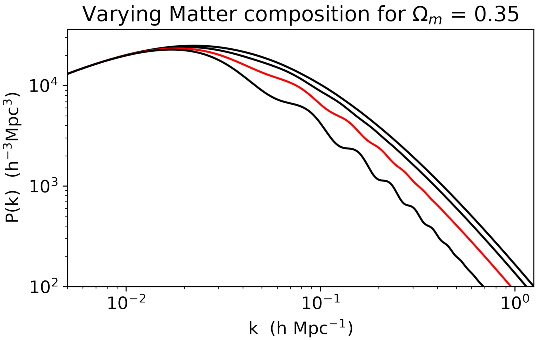


Figure: Linear matter power spectrum $P_m(k)$

Structure of CLASS !

Perturbative calculation

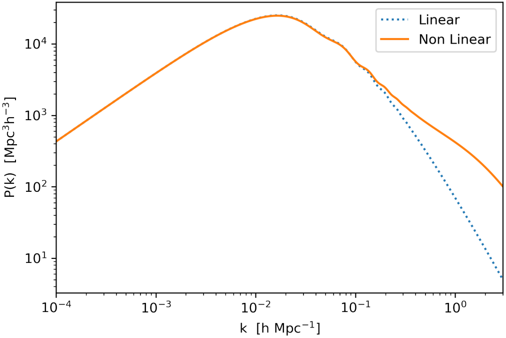


Figure: Linear, Non-linear matter power spectrum

Structure of CLASS !

Perturbative calculation

Step 5 : Harmonic Power spectrum

- Using results obtained so far,
- CMB C_ℓ spectrum :

$$\frac{\delta T}{\bar{T}}(\eta_0, \hat{n}) = \sum_{\ell,m} a_{\ell m} Y_{\ell m}(\hat{n})$$

$$\langle a_{\ell m} a^*_{\ell' m'} \rangle \stackrel{!}{=} \delta_{\ell\ell'} \delta_{mm'} C_\ell$$

- CLASS uses metric perturbations to calculate gravitational lensing corrections.

Structure of CLASS !
Perturbative calculation

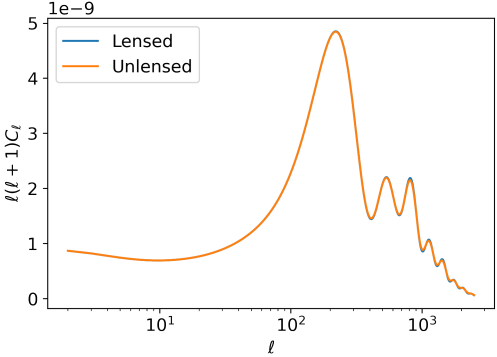


Figure: Lensed and Unlensed C_ℓ^{TT} spectrum

Usage : Example

CLASS - Background

ΛCDM Cosmology

```
from classy import Class
lcdm = Class()
lcdm.compute()
```

- `.compute()` $\longrightarrow$ Background, Thermodynamic

How to obtain a different cosmology ?

```
from classy import Class
diff = Class()
diff.set({'Omega_cdm' : 0.35, 'Omega_b' : 0.10 })
diff.compute()
```

Input Parameters

- Either use Omega_x i.e. Ω_x or omega_x i.e. ω_x, for different species where x $\in$ {cdm, b}
- Either T_cmb or Omega_g or omega_g
- Also specify Omega_k, Omega_Lambda, n_s, A_s
- Cannot set T_cmb, Omega_b to zero.

Constraint : Budget equation $\sum_i \Omega_i + \Omega_k = 1$

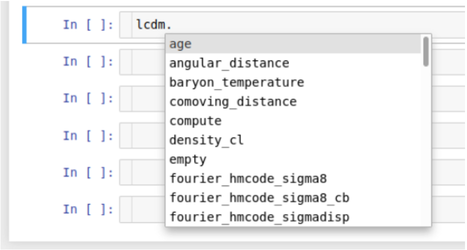
- CLASS always sets a flat universe $\Omega_k = 0$
- You increase $\Omega_b \uparrow \underset{\text{CLASS}}{\Longrightarrow} \Omega_\Lambda \downarrow$
- Inbuilt option to turn on species. See, explanatory.ini

CLASS - Background

How do you get the output now ?

⋮

```
lcdm.compute()
lcdm. Tab
```



Getting age

```
lcdm.age()
```

CLASS - Background

Getting distances ?

```
⋮
lcdm.compute()
lcdm.comoving_distance(0.2)
```

```
In [33]:  lcdm.comoving_distance(0.2)
          executed in 14ms, finished 10:06:16 2021-12-15
Out[33]:  841.8571983974737
```

- All distances are returned in units Mpc

Temperature C_ℓ spectrum

Initialize

```
lcdm = Class()
lcdm.set({'output' : 'tCl'})
lcdm.compute()
cls = lcdm.raw_cl()
```

- `.raw_cl` returns a python dictionary, to see the contents of the dictionary use `.keys()`

```
In [70]: cls.keys()
executed in 14ms, finished 13:30:12 2021-12-15
Out[70]: dict_keys(['tt', 'ell'])
```

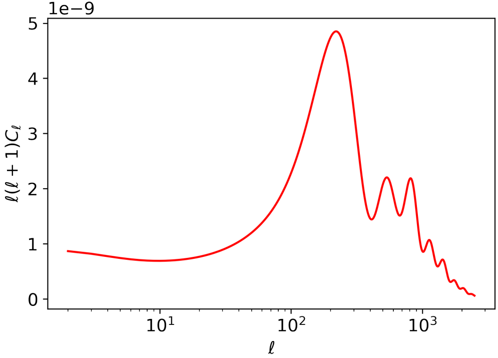
Extract the values

```
l = cls['ell'][2:]
clTT = cls['tt'][2:]
```

Temperature C_ℓ spectrum

Plot

```
plt.plot(l, l*(l+1)*clTT)
plt.xscale('log')
```



- The above procedure can be iterated over a loop to get this pretty plot.

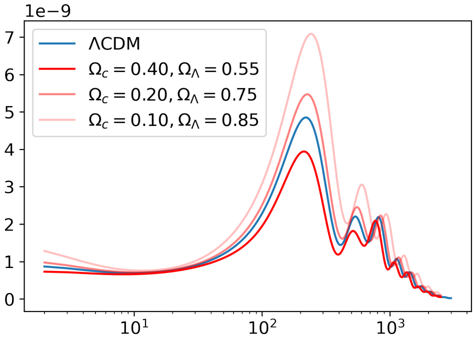


Figure: C_ℓ^{TT} for varying Ω_c, Ω_Λ

Summary

1. Black Box : $\begin{cases} \text{Parameters} \rightarrow \text{Obervables} \\ \text{Observations} \rightarrow \text{Constraints} \end{cases}$
2. Black box = Einstein Boltzmann Solver + Sampler
3. Background $\rightarrow$ Thermo. $\rightarrow$ Perturbations $\rightarrow$ Obs.
4. Usage examples

References

[1] Niklas Becker et al., *Cosmological constraints on multi-interacting dark matter*, arXiv:2010.04074v2

[2] Chung-Pei Ma, Edmund Bertschinger, *Cosmological Perturbation Theory in the Synchronous and Conformal Newtonian Gauges*, arXiv:astro-ph/9506072v1