Berkeley, 29.04.2025



J. Lesgourgues

Institut für Theoretische Teilchenphysik und Kosmologie (TTK), RWTH Aachen University





















EUCLID photometric probe (3x2pt): sensitivity to non-standard DM

Euclid preparation. Sensitivity to non-standard particle dark matter models,[2406.18274]by Euclid collaboration: JL, J. Schwagereit, J. Bucko, G. Parimbelli et al., accepted in A&A



Linear level: all models implemented in CLASS

Non-linear level: assume $P_{obs}(k, z) = P_L^{\Lambda CDM}(k, z) \times T^{DM}(k, z) \times T^{BF}(k, z)$ Emulator from N-body Emulator from hydro sims

Parimbelli et al, Bucko et al.

Giri & Schneider



EUCLID photometric probe (3x2pt): sensitivity to non-standard DM

MCMC forecasts for Euclid 3x2pt:

- Great sensitivity increase (
- e.g. for CWDM and $m_{\rm wdm}^{\rm thermal}$

, ty to 1% warm, 99% cold !



Extended DM & photometric probe (3x2pt): theoretical challenges

Non-linear level: assume
$$P_{obs}(k, z) = P_L^{\Lambda CDM}(k, z) \times T^{DM}(k, z) \times T^{BF}(k, z)$$

Emulator from N-body Emulator from hydro sims

To be improved for weak lensing and photometric galaxy clustering:

- Beyond separability assumption of cosmo/DM/baryonic parameters
- Accuracy of N-body and baryonic feedback simulations (from micro to macro processes)
- Intrinsic alignement modelling
- Replace simulations by semi-analytic EFT of Cosmic Shear (Chen & Kokron 23; Chen, DeRose et al. 24; ...)

For spectroscopic galaxy surveys:

• Semi-analytic methods like EFTofLSS are mature and necessary!



spectroscopic probe: EFTofLSS

- One-loop EFT galaxy power spectrum in redshift space = linear + 42 loop integrals + 11 bias and EFT parameters per sample / redshift bin
 - Implemented in various codes:
 - CLASS-PT [Chudaykin, Ivanov, Philcox, Simonovic 20],
 - velocilaptors [Chen et al. 20],
 - PyBird [D'Amico et al. 20],
 - FOLPS [Noriega et al. 24],
 - cosmosis_gclust [Moradinezhad et al. 24],
 - CLASS-OneLoop [Linde et al. 24]; (full integration in CLASS, calculation in O(10) ms, parallelised C, new method for Fourier log-transform)
 - Uncertainty on counter-terms
 critical to measure extended DM
 parameters



9 / 28 DM properties from stage-IV surveys - J. Lesgourgues

Extended DM & EFTofLSS: theoretical challenges

- Models with suppression of structures imprinted during radiation-dominated (RD) era. Examples:
 - warm DM (WDM), hot DM (HDM), mixed C+WDM, mixed C+HDM (mass in 10 eV 10 keV range) when neglecting thermal velocity effects during MD
 - DM scattering off dark radiation (DR). DR can be free-streaming or self-interacting. Either dark decoupling (ETHOS n>0) or Γ/H = constant during RD (ETHOS n=0)
 - DM scattering off neutrinos; some models of DM scattering off baryons
- Models with suppression of structures imprinted during matter-dominated (MD) era. Examples:
 - WDM, HDM, C+WDM, C+HDM (mass in 1 eV 100 eV range) if high precision required (thermal velocity effects during MD)
 - Decaying dark matter. Simplest: relativistic daughters. Next level: relativistic + lighter daughter (CDM—> DR+WDM)
 - DM with self-interactions (SIDM): short-range; long-range, with interacting DM-dark energy as possible limit
 - Some models of DM scattering off baryons



Extended DM & EFTofLSS: theoretical challenges

- Models with suppression of structures imprinted during radiation-dominated (RD) era. Examples:
 - warm DM (\ range) whe
 DM scatteri dark decou
 different ICs but same growth as ΛCDM ⇒ separability in k, z
 ⇒ standard EFTofLSS with EdS kernels
 - DM scattering off neutrinos; some models of DM scattering off baryons

Models with suppression of structures imprinted during matter-dominated (MD) era.
 Examples:

۲	WDM, HDM (thermal ve	scale-dependent growth
۲	Decaving d	\Rightarrow no separability in k, z
	daughter (\Rightarrow beyond EdS kernels

- DM with self-interactions (SIDM): short-range; long-range, with interacting DM-dark energy as possible limit
- Some models of DM scattering off baryons



Extended DM & EFTofLSS: degeneracy with nuisance parameters

Example in Mosbech et al. 2410.08163: DM interacting with neutrinos, scattering rate Γ

DESI ELG forecasts with 10% of DM interacting DM, unknown scattering rate Γ

- Γ degenerate with c_i at each z
- Information in z-dependance (tomography)
- If $c_i(z)$ known up to overall coefficient. \Rightarrow DESI very sensitive to $\Gamma \downarrow_{=}$
- If unknown $c_i(z_j)$ (with Gaussian priors) \Rightarrow sensitivity completely lost \P
- Bispectrum? Higher-order?



DESI-like MCMC forecasts with CLASS-OneLoop + MontePython

Incertainty on counter-terms critical to measure extended DM parameters



SPT, EFTofLSS, ... with scale-dependent growth

- Full integral over time- and scale-dependent kernels (Garny & Taule 20,22)
 - doublet: $\Psi_a = (\delta_{cb}, -\theta_{cb}/\mathscr{H}f)$

• e.o.m:
$$\partial_{\eta}\psi_a(\mathbf{k},\eta) + \Omega_{ab}$$
 $\psi_b(\mathbf{k},\eta) = \int_{\mathbf{k}_1,\mathbf{k}_2} \delta_D(\mathbf{k}-\mathbf{k}_{12})\gamma_{abc}(\mathbf{k},\mathbf{k}_1,\mathbf{k}_2)\psi_b(\mathbf{k}_1,\eta)\psi_c(\mathbf{k}_2,\eta)$

time-dependent kernels:

$$(\partial_{\eta} + n) F_a^{(n)}(\mathbf{q}_1, \dots, \mathbf{q}_n; \eta) + \Omega_{ab}(k, \eta) F_b^{(n)}(\mathbf{q}_1, \dots, \mathbf{q}_n; \eta)$$

$$= \sum_{m=1}^{n-1} \left[\gamma_{abc}(\mathbf{k}, \mathbf{q}_{1\cdots m}, \mathbf{q}_{m+1\cdots n}) F_b^{(m)}(\mathbf{q}_1, \dots, \mathbf{q}_m; \eta) F_c^{(n-m)}(\mathbf{q}_{m+1}, \dots, \mathbf{q}_n; \eta) \right]_{\text{sym.}}$$

- FOLPS: taking into account k-dependence of δ_{cb} and θ_{cb} at each time (Aviles et al. 21, 22..)
 - At given eta, kernels get multiplied by $D(k, \eta)$ or $f(k, \eta)$
- Potentially much quicker and still nearly exact method inspired from N-body: Newtonian Motion gauges





Newtonian Motion gauges

3 gauges:

- Gauge of Boltzmann code (Synchronous gauge, Poisson/Newtonian/longitudinal gauge)
- Gauge matching gauge-independent observables (Comoving gauge, $V_{\text{tot}} = B$, with $H_{\text{T}} = 0$)
- Gauge designed to get effectively Newtonian variables:
 - N-body, N-boisson, Newtonian Motion gauge (C. Fidler, C. Rampf, T. Tram, R. Crittenden, K. Koyama, D. Wands 2015, 2016, 2017)
 - particles follow same trajectories as if governed by Newtonian equations for single selfgravitating fluid
 - absorbs effects of GR, radiation, even massive neutrinos, modified gravity...

Self-consistency: perturbatively small coordinate transformation -> weak-field approximation holds also in NM gauge ($|H_T^{(NM)}| \ll 1$)



- Application to N-body simulations in 3 steps:
 - 1. Boltzmann code in S, P gauge: finds relation between gauges; checks self-consistency $(|H_T^{(NM)}| \ll 1)$; finds ICs for N-body at $z \sim O(100)$ in NM gauge
 - 2. N-body simulation: non-linear evolution in NM gauge (with special ICs, true background evolution)
 - 3. Gauge transformation back to S or C gauge (displacing particles); power spectrum, bispectrum from snapshots
- Final result from standard N-body code (cold particles only) matches relativistic sims with GR effects, w/o radiation, massive neutrinos, modified gravity (Fidler et al. 17; Adamek et al. 17, 22; Partmann et al. 20; Heuschling et al. 22; ...)



NM gauges and SPT, EFTofLSS, ...

- Application to SPT/EFTofLSS inb 3 steps: (Fidler, JL, A. Moradinezhad, S. Neuland, in prep.)
 - 1. Boltzmann code in S or P gauge: finds relation between gauges; checks self-consistency $(|H_T^{(NM)}| \ll 1)$; finds $P_{cb,L}^{(NM)}(k, z)$ in NM gauge
 - 2. Standard SPT/EFTofLSS with time-independent kernels: non-linear evolution in NM gauge; input: $P_{cb,L}^{(NM)}(k, z)$, with true background evolution; output: $P_{cb,NL}^{(NM)}(k, z)$
 - 3. Linear gauge transformation back to S or C gauge gives $P_{cb,NL}^{(C)}(k,z)$



• Deviations from EdS model accounted for in these 3 steps.



• C. Fidler, JL, A. Moradinezhad, S. Neuland, in prep.: application to $\Lambda \text{CDM} + M_{\nu}$





• C. Fidler, JL, A. Moradinezhad, in prep.: Redshift-space $P(k, \mu, z)$ or $P_{\ell}(k, z)$ in $\Lambda \text{CDM} + M_{\nu}$:



- Large scales: discrepancy with respect to standard EFTofLSS with pure EdS kernels, f(z)
- Much better agreement when restoring f(k, z) in tree-level (Kaiser term) in standard method
- Small scale: excellent agreement: f(k, z) effectively scale-independent on such scales
- Comparison with Garny et al. or FLOPS: expecting ~0.4% different growing at large k



- Back to SPT/EFTofLSS in Λ CDM (with massless neutrinos)
 - doublet: $\Psi_a = (\delta_{cb}, -\theta_{cb}/\mathscr{H}f)$
 - . non-linear e.o.m: $\partial_{\eta}\Psi_{a} + \Omega_{ab}\Psi_{b} = \int K_{abc} \Psi_{a}\Psi_{c}$
 - $\cdot \text{ evolution matrix: } \quad \Omega_{ab}(\eta) = \begin{pmatrix} 0 & -1 \\ -\frac{3}{2}\frac{\Omega_m}{f^2} & \frac{3}{2}\frac{\Omega_m}{f^2} 1 \end{pmatrix}$
 - second raw: approximately (-3/2, 1/2)
 - separability in k and η , time-independent EdS kernels K_{abc}



- Our case: in Newtonian Motion gauge:
 - scale-independent $f(\eta)$ and

$$\Omega_{ab}(\eta) = \begin{pmatrix} 0 & -1 \\ -\frac{3}{2} \frac{\Omega_m}{f^2} & \frac{3}{2} \frac{\Omega_m}{f^2} - 1 \end{pmatrix}$$

$$\times (1 - f_{\nu}) \qquad \text{with } \frac{\Omega_m}{f^2} \simeq \left(1 - \frac{3}{5} f_{\nu}\right)^{-2}$$

- Second raw approximately constant but not exactly equal to (-3/2, 1/2)
- Fully taken into account by Boltzmann code at linear order, but not in kernels
- Remaining challenge: take into account exact neutrino impact on scale-dependent growth beyond linear order
- Different NM gauge in which kernels become exactly EdS, or use kernels that are scale-independent but slightly different from EdS



NM gauges and SPT, EFTofLSS, ... : prospects

Once validation with $\Lambda CDM + M_{\nu}$ finished:

Very cheap method to make EFTofLSS consistent with GR and include more complex cosmologies

Next steps:

- Generalise to biased tracers
- Applicable to bispectrum, two-loops, ...
- Apply to models with stronger scale-dependence (decaying DM, modified gravity...)



Boosting parameter inference with emulators

- Emulating observables, prior training: CosmoPower (Spurio Mancini 22); CONNECT (Nygaard 22)
- Emulating observables, no prior training: OLÉ (Günther 2503.13183)
- Emulating likelihood, no prior training: GPry (El Gammal 2503.21871)

OLÉ (Günther 2503.13183)





OLÉ (Günther 2503.13183)

CLASS call NOT added to cache Using emulator without testing • CLASS call added to cache Using emulator with testing Testing of emulator failed 0.710.700.690.68 Ч 0.670.66 0.650.64 6000 8000 10000 12000 2000 4000 14000 30000 32000 34000 0 Step

Timings of the emulator























CLASS AI assistant: • Large Language Model (openAI)

- Retrieval Augmented Generation (RAG)
- Multi-agent communication

<u>https://classclapp.streamlit.app</u> (currently requires openAI API key)

Share 🟠 🖍 🗘 🗄

🔐 API & Assistants	
OpenAl API Key	
••••••••	
Password to encrypt/decrypt API key	
G	
Choose LLM model 🧠	
gpt-4o	~
Response Mode	
Fast Mode Wode Mode	ood
💉 Initialize with Selected Model	
📌 CLASS Setup	and guide you th
Install CLASS to enable code execution plotting capabilities	and implementation
Install CLASS	
Test CLASS	
	Type your prompt here



Hello! I'm here to assist you with the CLASS code, a cosmological tool that specializes in solving the Einstein-Boltzmann equations. My role is to help you understand the key components of the CLASS code and guide you through specific tasks such as setting up cosmological parameters, understanding differential equation-solving related to cosmology, and interpreting results. Whether you need help with implementation details or clarifying concepts in cosmology, feel free to ask!



CLASS AI assistant: • Large Language Model (openAI)

- Retrieval Augmented Generation (RAG)
- Multi-agent communication

https://classclapp.streamlit.app

(currently requires openAI API key)



Example Questions:

- 1. Can you show me how to plot in python the temperature C_ells for CMB using classy?
- 2. Can you show me a python code to plot the ratio between the lensing C_ells when using two nonlinear models: hmcode and halofit?
- 3. Give me a python code that compares all nonlinear matter power spectrum methods available in classy and shows the ratio against halofit
- 4. Give me the growth rate as a function of z, between0 and 3 for a LCDM cosmology
- 5. Show me the energy density of dark energy as a function of z, for a DESI-like cosmology with w0=-0.7 and wa=-0.8?



Can you show me what would be the difference in the lensing C_ells if I use two different nonlinear models: hmcode and halofit? And a plot that shows this difference

To compare the lensing (C_\ell) values using two different nonlinear models, halofit and hmcode, you can follow these steps. This involves setting up two separate CLASS instances with the different nonlinear models, computing the lensing power spectrum for each, and then plotting the difference.

Step-by-Step Guide

- 1. Import Necessary Libraries: You need classy for cosmological calculations and matplotlib for plotting.
- 2. Set Cosmological Parameters: Define the parameters for the Lambda Cold Dark Matter (ACDM) model.
- 3. Compute the Lensing Power Spectrum: Use CLASS to compute the lensing power spectrum for both halofit and hmcode.
- 4. Plot the Results: Use matplotlib to visualize the difference in the (C_\ell^{\phi\phi}).

Here's a complete code example:

```
# Import necessary modules
from classy import Class
import matplotlib.pyplot as plt
import numpy as np
from math import pi
# Define the cosmological parameters for the ACDM model
cosmo_params = {
   'h': 0.67810,
    'omega_b': 0.02238280,
    'omega_cdm': 0.1201075,
    'A_s': 2.100549e-09,
    'n_s': 0.9660499,
    'tau_reio': 0.05430842,
    'output': 'lCl',
    'lensing': 'yes',
    'l_max_scalars': 2500
```



Initialize CLASS instances for halofit and hmcode cosmo_halofit = Class() cosmo_hmcode = Class()

```
# Set parameters for halofit
cosmo_params['non linear'] = 'halofit'
cosmo_halofit.set(cosmo_params)
cosmo_halofit.compute()
```

Set parameters for hmcode cosmo_params['non linear'] = 'hmcode' cosmo_hmcode.set(cosmo_params) cosmo_hmcode.compute()

Retrieve the C_l data for both models
cls_halofit = cosmo_halofit.lensed_cl(2500)
cls_hmcode = cosmo_hmcode.lensed_cl(2500)

```
# Extract ell and C_ell^phi phi
ell = cls_halofit['ell'][2:] # Start from ell=2
clPP_halofit = cls_halofit['pp'][2:]
clPP_hmcode = cls_hmcode['pp'][2:]
```

```
# Calculate the difference
clPP_difference = clPP_hmcode - clPP_halofit
```

```
# Plotting the difference in lensing power spectrum
plt.figure(figsize=(10, 6))
plt.plot(ell, clPP_difference, 'b-', label=r'Difference in $C_\ell^{\phi\phi}$ (HMCode - Halofit)')
```

```
plt.xlabel(r'Multipole moment $\ell$')
plt.ylabel(r'Difference in $C_\ell^{\phi\phi}$')
plt.title('Difference in Lensing Power Spectrum (HMCode vs Halofit)')
plt.legend()
plt.grid(True)
```



Explanation

- Cosmological Parameters: These are standard parameters for a flat ACDM model. The non linear parameter is set to either halofit or hmcode to specify the nonlinear model.
- CLASS Initialization: Two class objects are initialized and configured with the cosmological parameters for each nonlinear model.
- Power Spectrum Calculation: The lensed_cl method retrieves the lensed CMB power spectrum up to a specified multipole (\ell).
- Plotting: The difference in the lensing power spectrum (C_\ell^{\phi\phi}) is plotted, showing how the two models differ.
- Saving the Plot: The plot is saved to a file named lensing_power_spectrum_difference.png .

This code will generate a plot showing the difference in the lensing power spectrum between the hmcode and halofit models, providing insights into the impact of different nonlinear corrections on the CMB lensing signal.

Prote: This answer contains code. If you want to execute it, type 'execute!' in the chat.

execute!



Fixing errors (attempt 1/3)...

Previous error: STDOUT: Traceback (most recent call last): File "/tmp/tmpr_jxo3ea/temp_script_2025-04-23-18-23.py", line 26, in <module> cosmo_halofit.compute() File "python/classy.pyx", line 352, in classy.Class.compute classy.CosmoSevereError:

Error in Class: input_read_from_file(L:428) :error in input_read_parameters(pfc,ppr,pba,pth,ppt,ptr,ppm,phr,pfo,ple,psd,pop, errmsg); =>input_read_parameters(L:1697) :error in input_read_parameters_lensing(L:5056) :error; you asked for lensed CMB Cls, but this requires a minimal number of options: 'modes' should include 's', 'output' should include 'tCl' and/or 'pCl', and also, importantly, 'lCl', the CMB lensing potential spectrum.

Executing corrected code...

Execution Output

✓ Plot generated successfully!



Code executed successfully. Last execution message:



