CAMB and EFTCAMB

A survivor's guide

Matteo Martinelli - Instituut Lorentz

Madrid, March 14th 2017

What are we going to do?

CAMB

1. Structure
2. Compiling and running
3. Easy exercise: running the code with different cosmologies
   Less easy: add new parameters

EFTCAMB

1. What’s new?
2. Compiling and running
3. Easy exercise: run the code in pure EFT and designer modes
   Less easy: add a new parameterization

CAMB in a few words

by Antony Lewis and Anthony Challinor

http://camb.info/readme.html for detailed documentation

Good:

- constantly updated
- well tested by a huge community
- powerful and reasonably flexible
- easy to modify
  (once you know your way)

Less good:

- constantly updated (stay updated!)
- fortran
- non trivial formalism

Useful links

http://camb.info/readme.html CAMB readme

http://cosmologist.info/notes/CAMB.pdf implementation notes

What’s in the files?

Utilities

- bessels.f90
- inifile.f90
- Matrix_utils.f90
- subroutines.f90
- utils.F90
- writefits.f90

Cosmology

- camb.f90
- cmbmain.f90
- cosmorec.f90
- equations.f90
- halofit.f90
- hyrec.f90
- lensing.f90
- modules.f90
- power_tilt.f90
- recfast.f90
- reionization.90
- SeparableBispectrum.f90

Driver and params

- inidriver.F90
- params.ini

A closer look at cosmology

Evolution:

- equations.f90
- equations_ppf.f90

Initial power spectrum:

- power_tilt.f90
  initial power parameters definition and reading

Modules:

- Modules.f90
  variables declaration, background functions, P(k), output

Recombination:

- recfast.f90
- cosmorec.f90
- hyrec.f90

Reionization:

- reionization.f90

CMB lensing:

- lensing.f90

Non linear P(k):

- halofit.f90
- halofit_ppf.90

Bispectrum:

- SeparableBispectrum.f90

Standard parameters: definition and reading

Densities and background

Defined in modules.f90

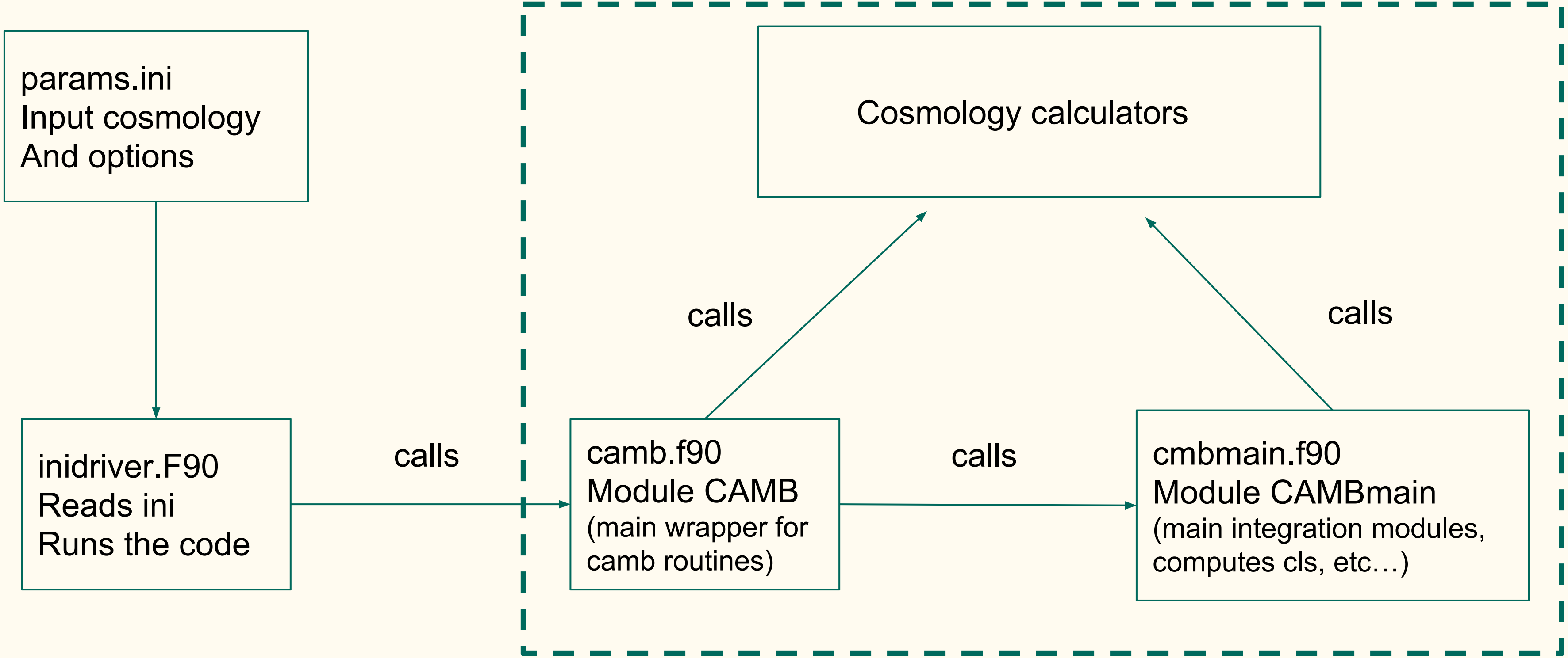
Read in inidriver.f90

Initial power spectrum

Defined in power_tilt.f90

Read in power_tilt.f90

Schematic structure



To use CAMB in an external code: use CAMB, call CAMB_GetResults (see CosmoMC)

Let’s compile!

Makefile and Makefile_main contain the compilation instructions

Set everything to work on your system

> make clean
> make

Hopefully everything will go fine

Setting the cosmology and running the code

Input cosmology and running options are in params.ini

- Output options
- Cosmological parameters
- Running options

Once the cosmology is set:

```
> ./camb params.ini
```

The output

Outputs

Unlensed scalar angular power spectra are output to **output_root**_scalCls.dat. The columns are

$l\ C_{TT}\ C_{EE}\ C_{TE}\ [C_{\Phi}\ C_{\Phi T}]$

Here all C_X are l(l+1)C_l/2pi except for C_{Φ} and $C_{\Phi T}$ which are $C_{\Phi}= l^4\ C_l^{\Phi}$, where C_l^{Φ} is the (CMB) lensing potential power spectrum, and $C_{\Phi T} = l^3\ C_l^{\Phi T}$. The lensing terms in square brackets are only produced if **do_lensing = T**. If **CMB_outputscale = 7.4311e12** ($[T_{CMB} 10^6]^2$, the default), the units are μK^2. Note that lensing spectra are also multiplied by CMB_outputscale, so you may want to divide this out of the answer to get a sensible dimensionless spectrum or use the **lens_potential_output_file** file mentioned below. If requested the lensed power spectrum is output to **output_root**_lensedCls.dat

Tensor angular power spectra are output to **output_root**_tensCls.dat if requested. The columns are

$l\ C_{TT}\ C_{EE}\ C_{BB}\ C_{TE}$

If scalars and tensors are generated, the total spectrum is in **output_root**_totCls.dat, in the same format as the tensor output file.

If **do_lensing=T** and **lens_potential_output_file** is specified a file is output containing unlensed scalar (+tensor if calculated) spectra along with the lensing potentials in this format:

$l\ C_{TT}\ C_{EE}\ C_{BB}\ C_{TE}\ C_{dd}\ C_{dT}\ C_{dE}$

where as before C_X are $l(l+1)C_l/2\pi$, and d is the deflection angle, so $C_{dd}=[l(l+1)]^2 C_l^{\Phi}/2\pi$, $C_{dT}=[l(l+1)]^{3/2} C_l^{\Phi T}/2\pi$, $C_{dE}=[l(l+1)]^{3/2} C_l^{\Phi E}/2\pi$. These are the spectra required for simulating lensed skies using LensPix.

If transfer functions are requested the columns in the **output_root**_transfer.dat output file are:

1	k/h	wavenumber in h Mpc^{-1}
2	Delta_CDM/k^2	CDM
3	Delta_b/k^2	baryons
4	Delta_g/k^2	photons
5	Delta_r/k^2	massless neutrinos
6	Delta_nu/k^2	massive neutrinos
7	Delta_tot/k^2	CDM+baryons+massive neutrinos
8	Delta_nonu/k^2	CDM+baryons
9	Delta_totde/k^2	CDM+baryons+massive neutrinos+ dark energy(numerator only)
10	Φ	The Weyl potential $(\varphi+\psi)/2$
11	vel_Newt_cdm/k^2	vel_Newt_cdm is $-v_{cdm}$ k/H (Newtonian-gauge CDM velocity v_{cdm})
12	vel_Newt_b/k^2	vel_Newt_b is $-v_b$ k/H (Newtonian-gauge baryon velocity v_b)
13	vel_baryon_cdm/k^2	relative baryon-CDM velocity (v_b-v_{cdm})

where Delta_X is defined as (delta rho_X)/rho_X in the synchronous gauge and evaluated at the requested redshift, given a unit primordial curvature perturbation on superhorizon scales (for adiabatic modes, chi_0=-1). The column numbers correspond to the Transfer_xx integer constants defined in the Transfer module (modules.f90).

output_root_matterpower.dat contains the conventionally normalized matter power spectrum (for baryons+cdm+massive neutrinos), in h/Mpc units.

CAMB readme

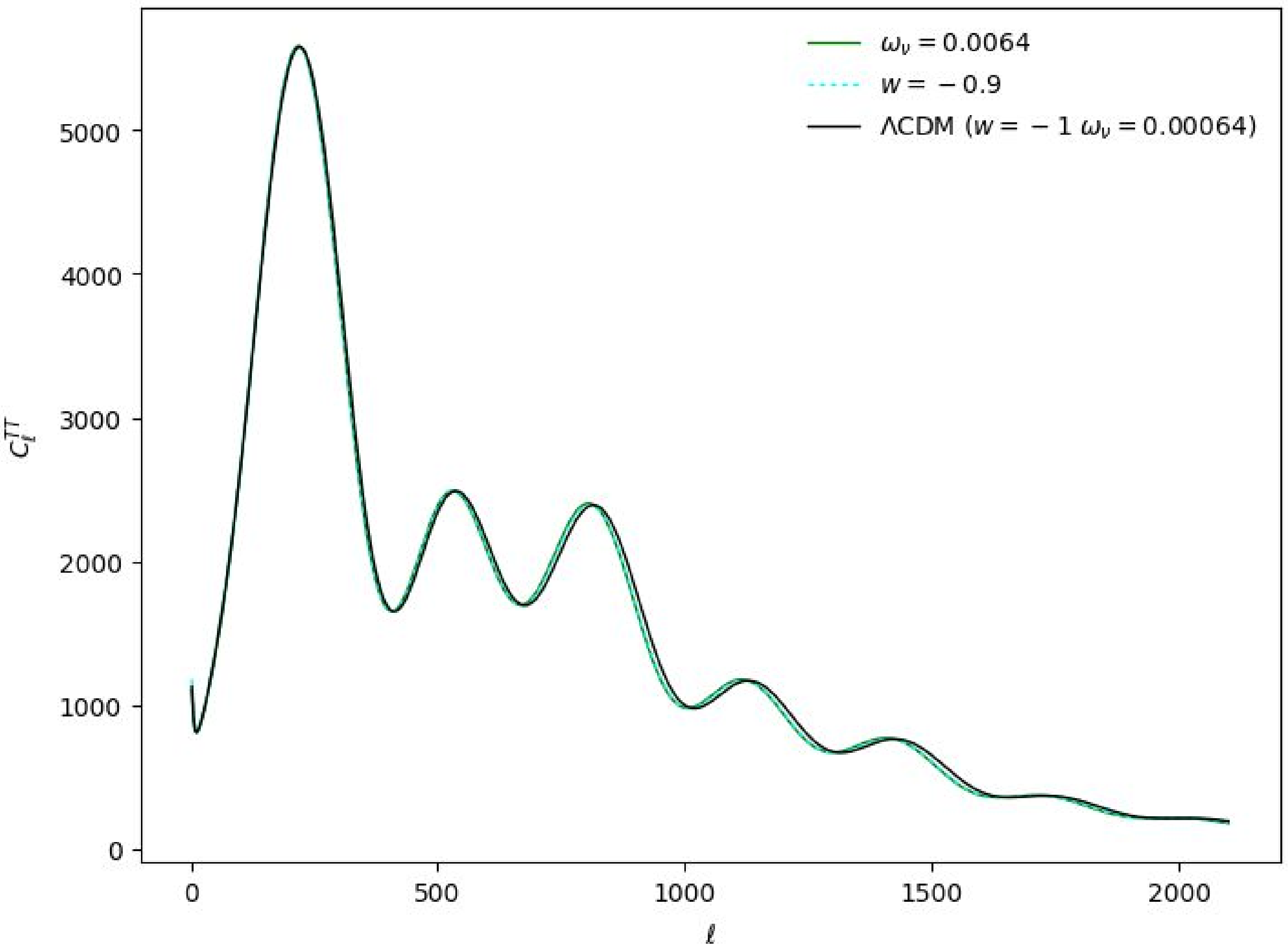
Exercise

Run CAMB for

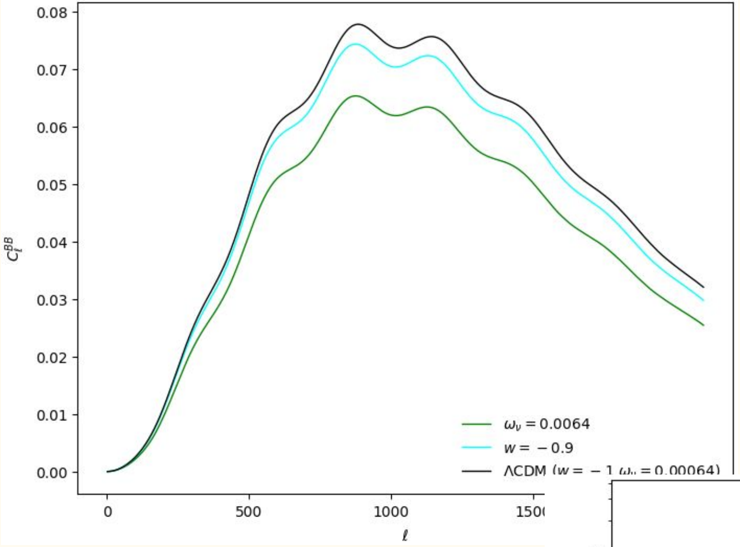
- our fiducial LCDM cosmology (w=-1, omeganuh2 = 0.00064)
- fiducial + omeganuh2 = 0.0064
- fiducial + w = -0.9

Use the options to obtain lensed C_l and P(k)
What's the best observable to distinguish the two extended models?

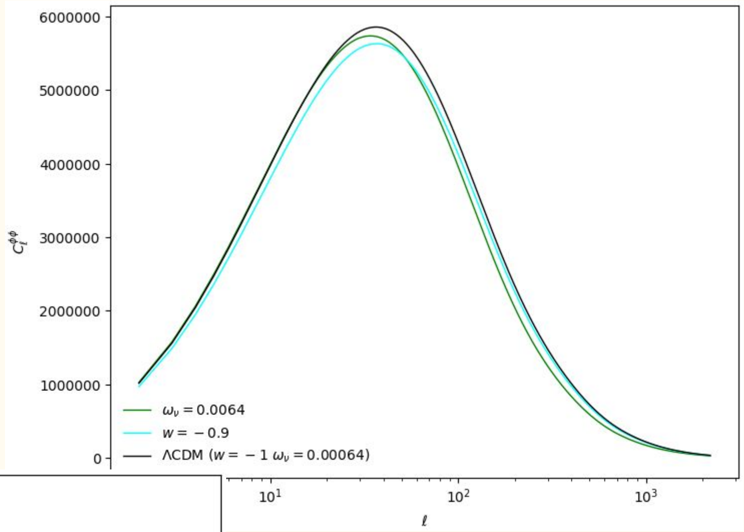
Keep practicing:
reproduce some of Wayne Hu's animation http://background.uchicago.edu/~whu/metaanim.html



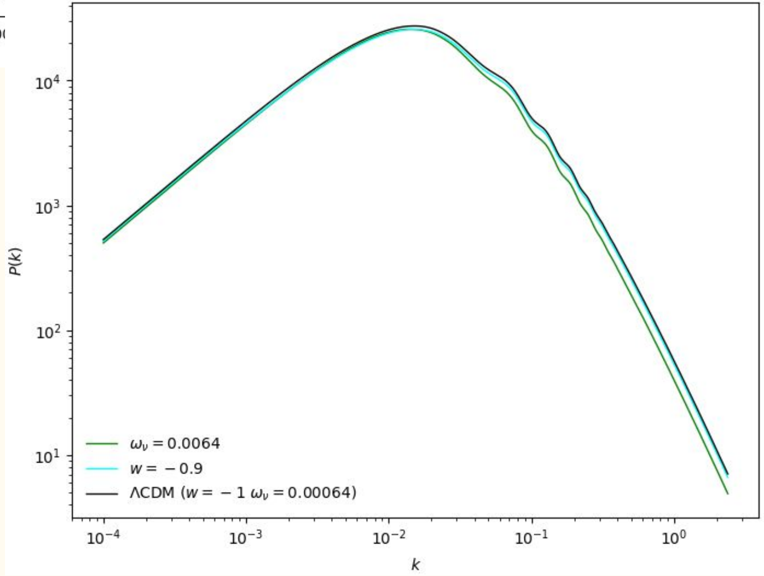
$\omega_\nu = 0.0064$
$w = -0.9$
ΛCDM ($w = -1$ $\omega_\nu = 0.00064$)
5000
4000
3000
2000
1000
0
C_ℓ^{TT}
0
500
1000
1500
2000
ℓ



C_ℓ^{BB}
ℓ
$\omega_\nu = 0.0064$
$w = -0.9$
ΛCDM ($w = -1$ $\omega_\nu = 0.00064$)

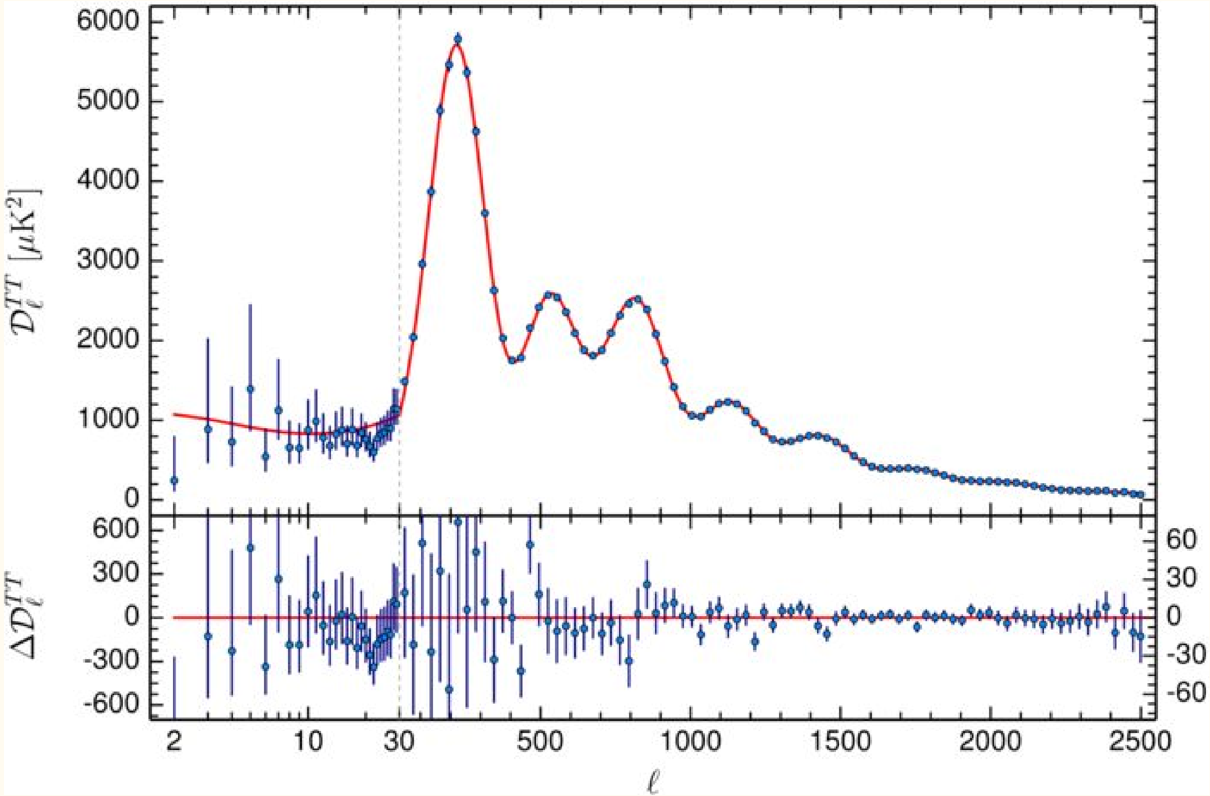


$C_\ell^{\phi\phi}$
ℓ
$\omega_\nu = 0.0064$
$w = -0.9$
ΛCDM ($w = -1$ $\omega_\nu = 0.00064$)



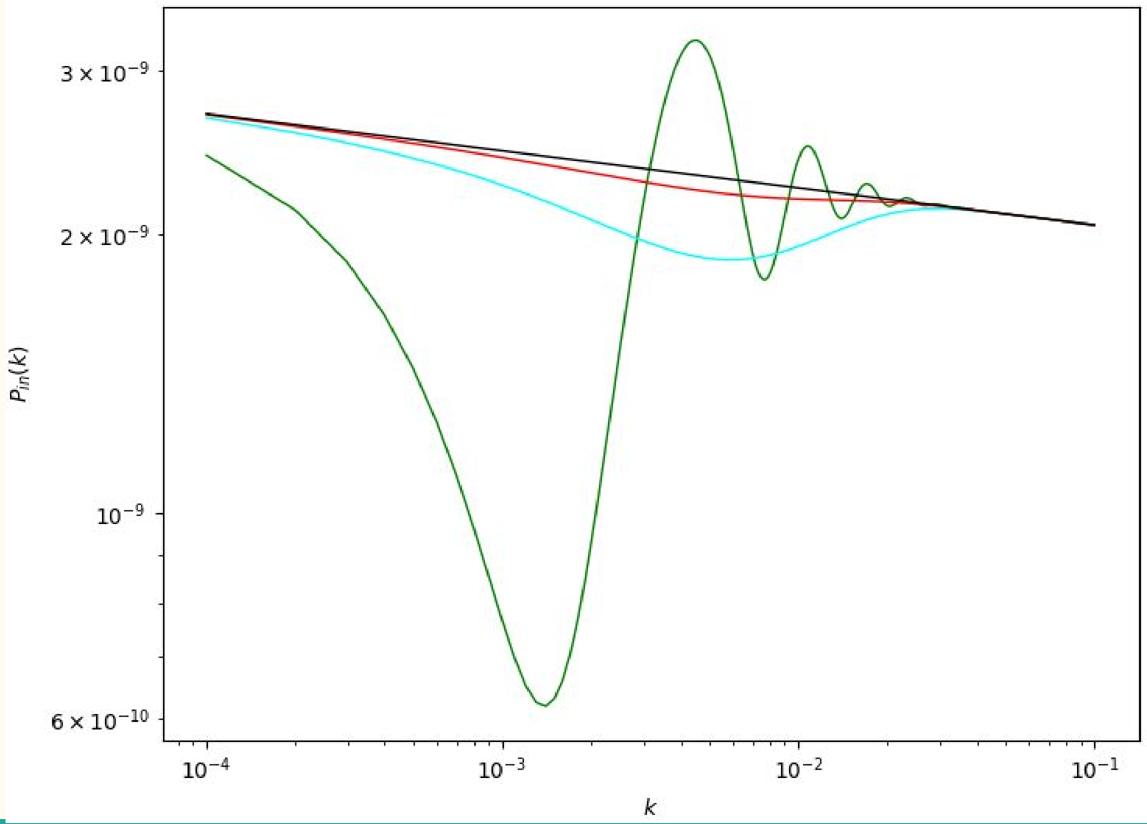
$P(k)$
k
$\omega_\nu = 0.0064$
$w = -0.9$
ΛCDM ($w = -1$ $\omega_\nu = 0.00064$)

Homework: features in the primordial P(k)



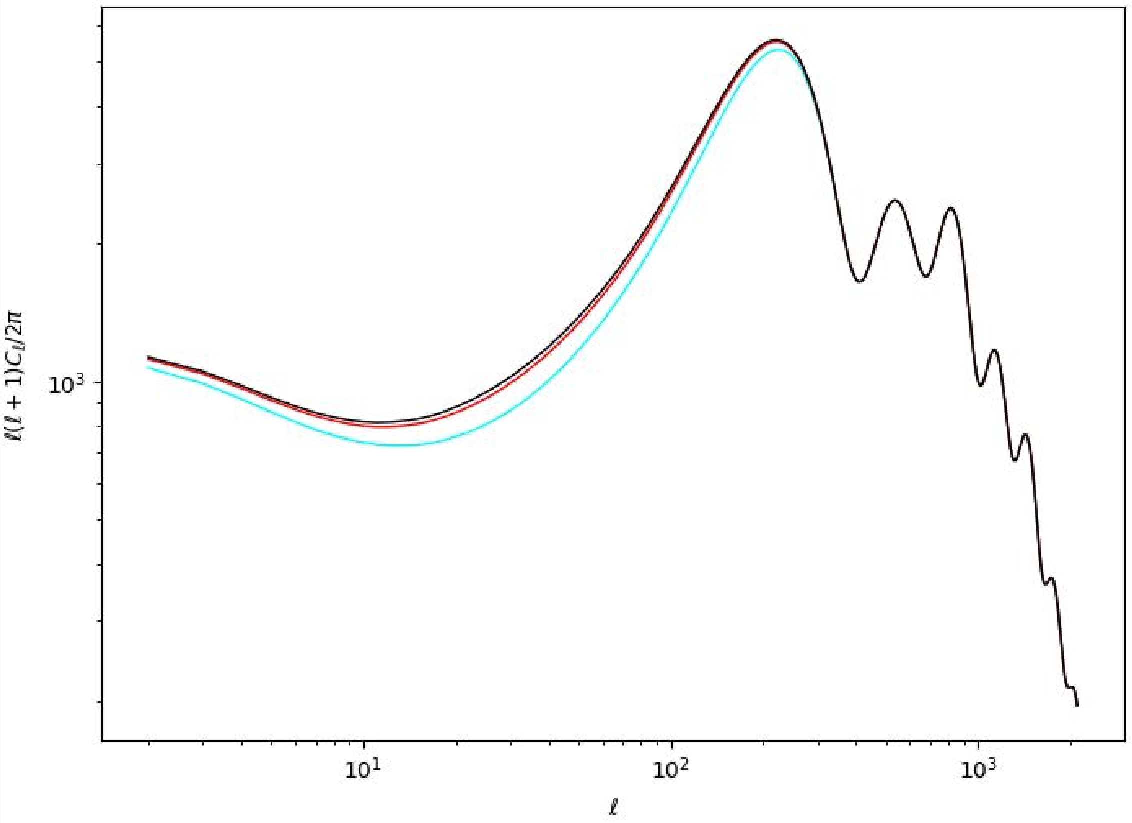
Can we solve the low-l discrepancy changing the primordial power spectrum?

Homework: features in the primordial P(k)



Can we solve the low-l discrepancy changing the primordial power spectrum?

Homework: features in the primordial P(k)



Can we solve the low-l discrepancy changing the primordial power spectrum?

Homework: features in the primordial P(k)

Include new initial power parameters in CAMB:

- declare and read them in power_tilt.f90
- include the modification in the ScalarPower function
- add parameters to the params.ini
- run the code

Tip: to add new parameters, follow an existing one and look how this is implemented.

EFTCAMB

By B. Hu, M. Raveri, N. Frusciante and A. Silvestri

http://eftcamb.org/ for more details.

Latest stable version based on CAMB Feb15 - more updated version on github

EFTCAMB: what’s new and what’s different

- EFT_def.f90
  Definitions and options (sets when the code goes back to GR)
- EFT_designer.f90
  Module for designer models
- EFT_horndeski.f90
  Implementation of Horndeski models
- EFT_functions.f90
  Contains definitions of pure EFT parameterizations and calls designer EFT functions
- EFT_main.f90
  Checks stability of selected model and computes return to GR
- equations_EFT.f90
  Computes evolution for perturbations. Written in term of EFT functions (no need to be changed if you change model)
- cmbmain.f90
  Modified to account for EFT evolution

To find modifications, look for the ! EFTCAMB MOD tag
Or have a look at the numerical notes arXiv:1405.3590

Input flags and parameters

Input given by params.ini (standard) + params_EFT.ini

EFTflag:
determines if the code is used as pure EFT, designer or GR
Depending on this choice, further parameters will or will not be used

Compiling and running

Modified Makefile includes the new modules

Set everything to work on your system

```
> make clean
> make
```

Hopefully everything will go fine

```
> ./camb params.ini
```

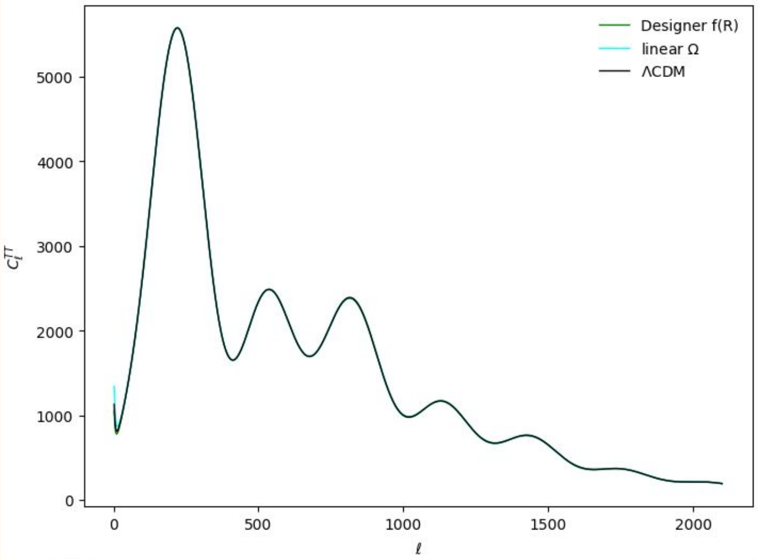
Exercise

Compare pure EFT (linear Omega) and designer f(R):

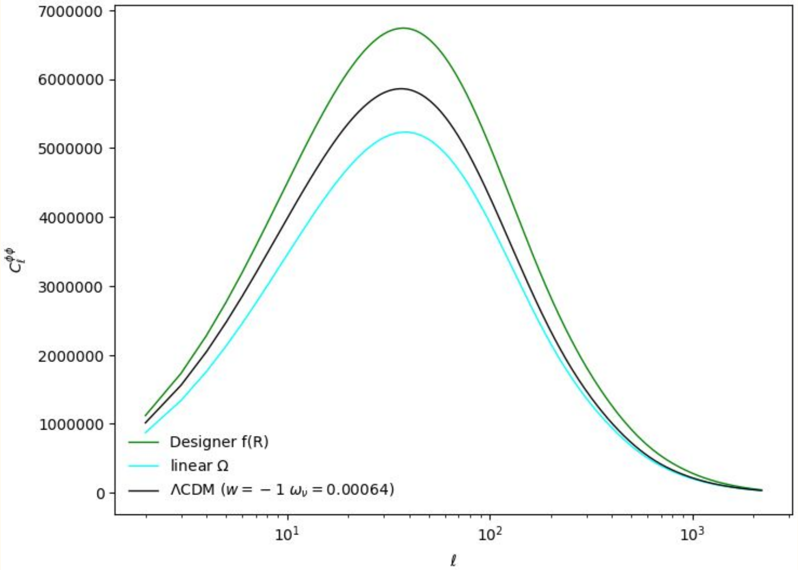
- EFTflag = 1
- PureEFTmodelOmega = 2

- EFTflag = 2
- DesignerEFTmodel = 1
- Look for the designer mapping parameter

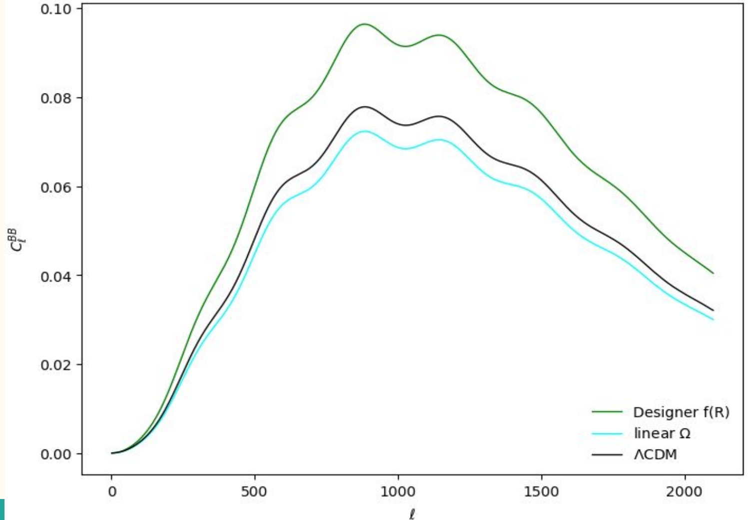
Try other parameterizations for Omega or the full mapping



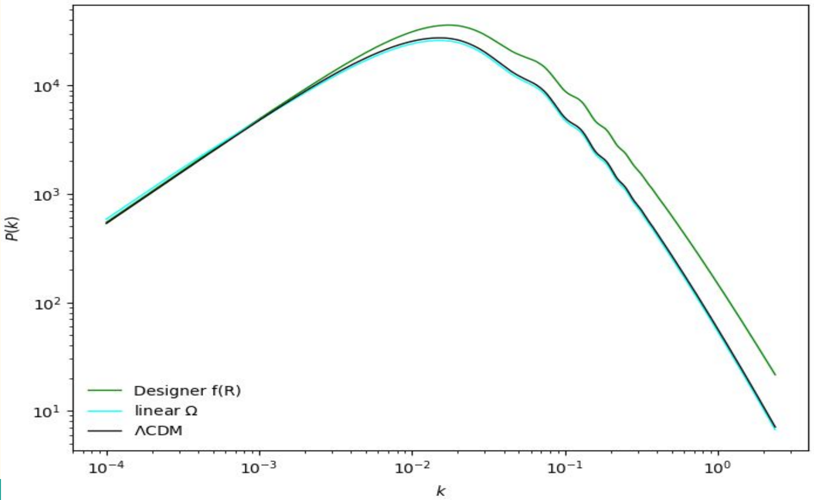
Designer f(R)
linear Ω
ΛCDM
C_ℓ^{TT}
ℓ
0
1000
2000
3000
4000
5000
500
1500
2000



Designer f(R)
linear Ω
ΛCDM ($w = -1$ $\omega_\nu = 0.00064$)
$C_\ell^{\phi\phi}$
ℓ
0
1000000
2000000
3000000
4000000
5000000
6000000
7000000
10^1
10^2
10^3



Designer f(R)
linear Ω
ΛCDM
C_ℓ^{BB}
ℓ
0.00
0.02
0.04
0.06
0.08
0.10
0
500
1000
1500
2000



Designer f(R)
linear Ω
ΛCDM
$P(k)$
k
10^1
10^2
10^3
10^4
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^0

Homework: add a new parametrization

Add a new parametrization for Omega

- Choose your parameterization (JBP)
- Implement it as the “user defined” in EFT_functions.f90
- If necessary, declare the new parameters in modules.f90
- Compute and implement derivatives

Tip: look at an existing parameterization and follow the same procedure.

Disclaimer: procedure will change in the new EFTCAMB version (it will get easier)