Homework SURP: Learning to work with a Boltzmann code

I. INSTALLING CAMB

There are two widely used Boltzmann codes: `CAMB` and `CLASS`. The difference between the two is that the first is written in Fortran 90, and the second one in c++. The latter code is a little more transparent, while the first one is more powerful and has been around for over 10 years. There is also a big community support forum: `http://cosmocoffee.info`. Because some of the code you will be using for your project this summer will mostly be in Fortran (some Python plotting) we will start by using `CAMB`. Joel is an expert in `CLASS` and we always switch if needed.

In order to get the code, you go to `http://camb.info`. There you will find some documentation (how to's) and a link to the source code. It requires signing up, so the earlier you do so, the better (in my experience verification is instant). If for some reason you do not get the code, let me know since I can provide a direct link.

After downloading the code to your home directory (easiest is to use the command `wget` followed by the `url`). To install, you need to load the correct packages in order to get the latest `Intel` compiler. Once installed, you can read the manual. The most important command is `./camb` followed by a parameter input file `params.ini`. `camb` will read in the files settings and run the Boltzmann code.

II. FAMILIARIZING YOURSELF WITH SETTINGS

Next, you should try an change parameters. The first assignment is to generate a cosmology with the best-fit parameter values as constrained by Planck (1502.02114, page 9, Table 3). Be carefull since the current default might not set to a 6 parameter ΛCDM Universe. The root of the output files should be called "`Planckbestfit2016`". After running the code with the correct parameters, there should be a few files, one with sclCls, one with tensCls and one with lensCls. Each file will contain several columns. Online you can find the content of each column. Note that if you set `lensing = F` or `tensors = F` in the params.ini file, the latter files will not be generated.

The `CAMB` folder contains various plotting routines. You can use those or your own favorite plotting routine to plot the spectra. I would like you to do several things:

- Plot the temperature power spectrum (TT) and the E-mode power spectrum (EE) and the cross power spectrum (ET) for the scalars only.
- Do the same, now for the lensed spectra.
- Compare the two spectra; what is the difference?

III. THE EFFECT OF COSMOLOGICAL PARAMETERS

Now, I want to you start changing the values of the cosmological parameters, one by one (i.e. H_0, $\Omega_b h^2$, $\Omega_c h^2$, τ, n_s and A_s). Make sure you rename the root file accordingly (e.g. "`Planckbestfit2016_H0`")

- Describe the effect of each parameter. In order to best see the difference it is easiest to substract the two power spectra generated. It is sufficient to do this only for the scalar power spectra (so no lensing needed)
- Next, consider deviations from the ΛCDM Universe by changing the following parameters: neutrino mass and density, curvature and $N_{\rm eff}$. Again, describe the changes in the spectrum.

IV. THE PRIMORDIAL UNIVERSE

Your project will focus on the primordial Universe. I want you to familiarize yourself with the parameters that control the early Universe (and how the effect the observed power spectrum, and bispectrum). The default parameters are n_s and A_s. The first controls the scale dependence of the primordial power and A_s the total amplitude, i.e.

$$P_\zeta \propto \frac{A_s}{k^3}\left(\frac{k}{k_*}\right)^{n_s-1} \tag{1}$$

The observed temperature power spectrum is related to the spectra you just generated through e.g.

$$C_\ell^{TT} \propto \int dk k^2 P(k) \Delta_\ell(k)^2 \tag{2}$$

with $\Delta_\ell(k)^2$ the photon transfer function. The transfer functions depends on H_0, $\Omega_b h^2$, $\Omega_c h^2$ and τ. The `params.ini` contains several other parameters. They function of each parameter is described, to some extend, in the code file: `power_tilt.f90`. In that file you will also find the primordial spectrum that generates the tensor spectrum.

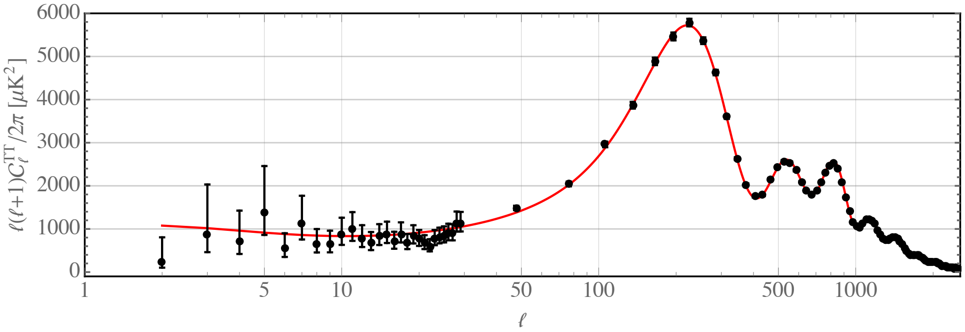

FIG. 1: An example of the temperature power spectrum including the marginalized errors from Planck. The red line represent the ΛCDM best fit spectrum. You should generate a spectrum that has a curve that has lower power on large scales (below ℓ of a few).

- Turn `tensors = T` and run `camb` (make sure you rename root e.g. "`Planckbestfit2016_r`") Set the tensor to scalar ratio r to 0.1.
- There should be a file containing the tensor spectra. Plot the following spectra: the contribution of tensors to TT, the contribution of tensors to EE and the contribution of tensors to BB.
- Play with other parameters, such as `scalar_nrun` and `tensor_spectral_index`. Describe what they do.
- Change the pivot scales. How does the pivot scale change the spectra?

V. MODIFY THE CODE

Next I want you to change part of the code. Change the initial power spectrum in such a way that you can fit better fit for the low quadrupole. The measured CMB temperature power spectrum C_ℓ^{TT} has relatively low power on $\ell = 2$ (i.e the quadrupole). Given the above equation that related the late time power to the primordial power, propose a shape of the power spectrum that produces a lack of power on large scales.

- Modify `power_tilt.f90` and change the function `function ScalarPower(k,in)` to generate this new spectrum. First, write a routine that prints the standard power spectrum (by inserting a `write` statement in the `function ScalarPower(k,in)`, preferably to a file).
- When modifying the code, make sure you first understand how parameters are called inside `function ScalarPower(k,in)` . It is much easier to modify the code when you understand the default. There is a parameter read routine at the end of the file. Adding parameters there will allow you to read in a parameter directly from the `params.ini` file.
- After you modify the power spectrum, plot the primordial power spectrum using the same routine.
- Plot the temperature power spectrum using your own primordial power spectrum including the error bars you can find on `http://irsa.ipac.caltech.edu/data/Planck/release_2/ancillary-data/previews/ps_index.html`. There is a FITS file that can be read and elements in the fits file represent different spectra including error bars (as described on the website). The plot should be looking something like Fig. 1.

If you have any questions, feel free to email me! Good luck.

Extra: If you have time left, write a routine (in Fortran or any other language) that reads in the error bars and your new spectrum and computes the χ^2 per degree of freedom of the default best fit and your new spectrum. Is its better? Is it better significantly?