N-body exercises

In this exercise sheet we will implement and study the wCDM model, which is a simple generalization of the ΛCDM model where dark energy is described by a barotropic fluid with equation of state $w \neq -1$. For simplicity, we will only consider the case $w = constant$ (*CLASS* allows w to vary in time) and shall not consider any perturbations of the fluid (*CLASS* generally allows for such perturbations, but if $w \sim -1$ and $c_s^2 \simeq 1$ these perturbations remain very small and can be neglected here).

The energy density for such a homogeneous fluid scales as

$$\rho_{\mathrm{fld}} \propto a^{-3(1+w)} \tag{1}$$

Adopting the parametrization of *CLASS*, the fluid parameters are given by

$$\begin{aligned} \texttt{Omega\_fld} &= \frac{8\pi G \rho_{\mathrm{fld}}^0}{3H_0^2} \\ \texttt{w0\_fld} &= w \end{aligned}$$

The ΛCDM model is a special case of wCDM where $w = -1$. Other values of w will lead to a different expansion history and hence to a modified growth of structure. Conversely, the observed growth rate of structure can be used to constrain the fluid parameters of wCDM.

Exercise 1: Implementing the wCDM Model in *gevolution*

This exercise requires coding practice. If you want to skip this exercise, you can download the solution from `https://banshee.obspm.fr/index.php/s/UxgJrTFYIiMbRG0` and proceed directly with Exercise 2.

Since only a modification of the background equations is required, the implementation of the wCDM model can be done in a few simple steps:

1. add model parameters to the metadata structures (`metadata.hpp`)
2. allow parsing of model parameters from settings (`parser.hpp`)
3. modify background (`background.hpp`)
4. *optional (not required for this exercise): adjust hibernation routines* (`hibernation.hpp`)
5. *optional (not required for this exercise): adjust CLASS interface* (`class_tools.hpp`)

To get started, create a new branch of *gevolution* derived from version 1.1:

```
git checkout -b wcdm
```

You can switch between branches using the `checkout` functionality of *git*. The command above will create the new branch “wcdm” and switch to this branch.

Metadata Take a look at `metadata.hpp`. It contains a few `structs` that hold various metadata information, roughly grouped according to purpose. The logical point where one would insert the new model parameters is the `struct` called `cosmology`. Add two `double` fields to this `struct`, one for `Omega_fld` and one for `w0_fld`.

Parser The new model parameters need to be parsed from the settings file. The place where this should be done is the function `parseMetadata` in `parser.hpp`. Cosmological parameters are parsed towards the end of this code section, starting from line `1203`. In order to read new parameters from the settings file one can simply add new calls to `parseParameter`. The syntax for this auxiliary function is as follows:

```
parseParameter(params, numparam, "<name string>", <variable>)
```

The *<name string>* is simply what appears on the left of the equal-sign in the settings file, e.g. for an entry like

```
w0_fld = -1.1
```

one would have `"`*<name string>*`" = "w0_fld"`.

The *<variable>* should be a reference to the metadata field where the parameter is to be stored.

The function `parseParameter` returns *true* if the parameter could be parsed (i.e. it exists in the settings file and its value, appearing on the right of the equal-sign, could be interpreted according to the *type* of *<variable>*) and *false* otherwise. If the function returns *false* one may want to set *<variable>* to a default value.

The periodic boundary conditions used in almost all N-body codes enforce vanishing global curvature of the spatial sections. In order to fulfill this global constraint, *gevolution* never parses the value of the cosmological constant; instead, it assumes that the curvature is zero and hence infers

$$\Omega_\Lambda = 1 - \sum_{i \neq \Lambda} \Omega_i \tag{2}$$

It is necessary that we include the new Ω_{fld} in this equation, i.e. it has to be added to the corresponding line of `parser.hpp` which (prior to our modification) reads

```
cosmo.Omega_Lambda = 1. - cosmo.Omega_m - cosmo.Omega_rad;
```

In this way we can replace, or just partially replace, the cosmological constant by a more general dark energy fluid.

Background Finally, we have to implement the physical consequences of the model into the part of the code that deals with the background evolution: `background.hpp`. In particular, we have to change the function `Hconf` that computes the conformal Hubble rate $\mathcal{H}$ such that it includes a new contribution to the homogeneous energy density.

$$\mathcal{H}^2 = \frac{8\pi G}{3} a^2 \sum_i \rho_i = \frac{8\pi G}{3} a^2 \rho^0_{\mathrm{crit}} \sum_i \Omega_i a^{-3(1+w_i)} \tag{3}$$

Note that the function `Hconf` returns $\mathcal{H}$ in units where $\rho^0_{\mathrm{crit}} = 1$. One can therefore directly use the dimensionless density parameters Ω_i.

Three auxiliary functions can be used in *gevolution* in order to compute the time-dependent density parameters of matter, radiation, and cosmological constant. For full internal consistency, these auxiliary functions (called `Omega_m`, `Omega_rad`, and `Omega_Lambda`, respectively) should also be modified to account for the dark fluid. However, these functions are never used in the main part of the code. Adjusting this code section is therefore optional for this exercise.

Hibernation *(Optional)* In order to make the hibernation functionality of *gevolution* aware of the new model parameters, they should appear in the output of `writeRestartSettings`, a function defined in `hibernation.hpp`. Around line `110` the cosmological parameters are printed. Here one should simply add the new model parameters, so that they can be re-parsed when a simulation is resumed from a hibernation point.

CLASS Interface *(Optional)* If *CLASS* is linked as a library, it is straightforward to make the interface compatible with the new model. The relevant routine is called `initializeCLASSstructures` in `class_tools.hpp`. It essentially constructs a `file_content` structure as used in *CLASS* and fills it with the information that has been parsed into *gevolution*'s metadata `structs`. In order to include the two new model parameters, we have to increase `num_entries` by two and then add a couple of lines analogous to the ones where the other cosmological parameters (e.g. `Omega_cdm`) are entered.

Exercise 2: N-body Power Spectra for ΛCDM and wCDM

In this exercise, we will compare the growth of structure in two different cosmologies: ΛCDM standard cosmology and a wCDM model with $w = -1.1$. Let us first fix some common parameters that are well measured from CMB observations, namely

`A_s = 2.215e-9`	*inflationary amplitude of scalar perturbations*
`n_s = 0.9619`	*scalar spectral index*
`k_pivot = 0.05`	*pivot scale in units of inverse Mpc*
`omega_b = 0.022032`	*physical density of baryonic matter*
`omega_cdm = 0.12038`	*physical density of cold dark matter*
`T_cmb = 2.7255`	*CMB temperature in units of K*
`N_ur = 3.046`	*effective number of additional ultra-relativistic species (three massless neutrino species)*

For the ΛCDM model we set

```
h = 0.67556
```

whereas for the wCDM model we set

```
h = 0.71316
Omega_fld = 0.7199
w0_fld = -1.1
```

These parameters are chosen such that the linear matter power spectrum, when given in physical units (i.e. Mpc instead of Mpc/*h*), matches exactly between the two models at redshift $z = 0$ (on sub-horizon scales).

For the N-body simulations, we will choose identical box sizes in *physical* units. Following parameters can be used for a simulation with 128^3 particles:

`Ngrid = 128`	
`template file = sc1_crystal.dat`	*standard 4×4×4 particle template*
`tiling factor = 32`	*factor 1/4 w.r.t.* `Ngrid` *because of template size*
`initial redshift = 100.0`	
`Courant factor = 75`	
`snapshot redshifts = 1, 0`	
`snapshot outputs = Gadget2`	
`Pk redshifts = 30, 10, 3, 1, 0`	
`Pk bins = 1024`	
`Pk outputs = phi, B, chi, hij, delta`	

Furthermore, for the ΛCDM model we set

`Tk file = lcdm_tk.dat`	*linear transfer functions at initial redshift*
`boxsize = 101.334`	*in units of Mpc/h, therefore 150 Mpc*
`time step limit = 0.0927`	*see explanation below*

whereas for the wCDM model we set

`Tk file = wcdm_tk.dat`	*linear transfer functions at initial redshift*
`boxsize = 106.974`	*in units of Mpc/h, therefore 150 Mpc*
`time step limit = 0.1000`	*see explanation below*

The parameters above were adjusted such that both simulations have the same comoving boxsize of 150 Mpc. One should keep in mind that the comoving wavenumbers (in the power spectra outputs) will be given in units of *h*/Mpc, i.e. one should multiply these numbers with the respective values of *h* to arrive at the same units.

Time Stepping and Output It is very important to be aware that *gevolution* uses a time stepping that does not care about when output is requested. This means that output is not written *exactly* at the redshift requested, but rather at the first time step below that redshift. If one compares models with different expansion histories, as in the present case, one may end up comparing spectra at slightly different *actual* redshifts. The values for "`time step limit`" quoted above were carefully chosen such that the two simulations should produce their respective $z = 1$ outputs at *exactly* the same redshift. Without this precaution the matter power spectra would acquire an additional offset due to the different output times.

Initial Transfer Functions The initial transfer functions (`lcdm_tk.dat` and `wcdm_tk.dat`) should be computed using a linear Boltzmann code, e.g. *CLASS*. Alternatively, you can download them from `https://banshee.obspm.fr/index.php/s/UxgJrTFYIiMbRG0` or link *CLASS* as a library to have *gevolution* generate the transfer functions at runtime (the line "`Tk file = ...`" should then be omitted from the settings file). The latter requires the *CLASS* interface to be adjusted to the wCDM model, see Exercise 1 for details.

Comparison of Matter Power Spectra at Redshift z = 1 Run *gevolution*! Take the numerical power spectra of δ computed by the N-body code for the two models at redshift $z = 1$ and, after rescaling to common units, determine the fractional change $\delta P/P = (P_{\mathrm{wCDM}} - P_{\Lambda\mathrm{CDM}})/P_{\Lambda\mathrm{CDM}}$. You should obtain something like the figure below.

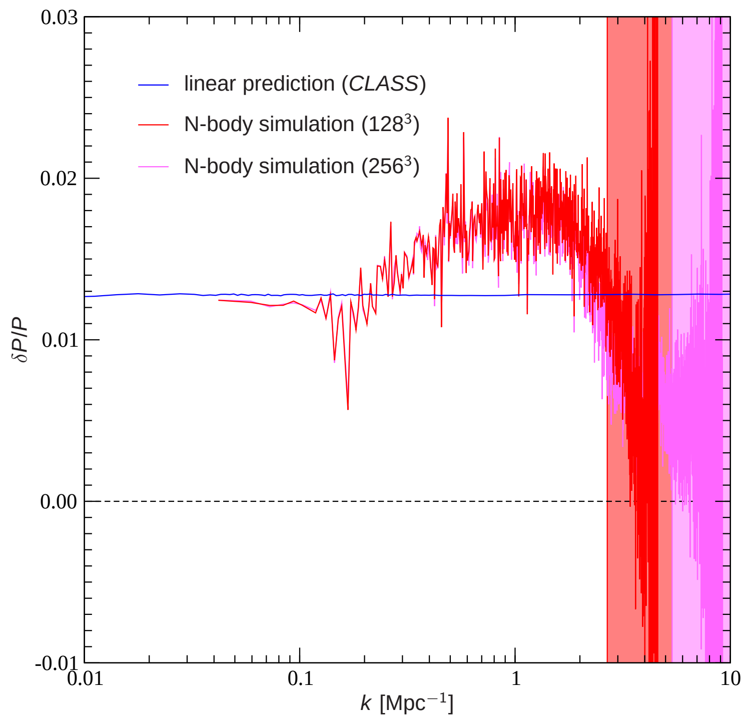

Figure 1: Fractional change in the matter power spectrum between wCDM and ΛCDM models at redshift $z = 1$. The shaded areas indicate the region beyond the Nyqvist wavenumber.

If you have time, run two more simulations at twice the resolution (for convergence study). The parameters for these should be as follows:

```
Ngrid = 256
tiling factor = 64
Courant factor = 150
Pk bins = 2048
```