Exercise: Evolution of Small Two-Fluid Perturbations

All structure in the Universe is thought to arise by gravitational collapse from the tiny fluctuations we see in the CMB. We will in this Exercise study how such density fluctuations grow in an expanding Universe.

The fluid equations in an expanding universe

Eulerian perturbation theory considers itself with solving equations for perturbations around the mean density of the Universe. This is done in a coordinate frame that is fixed in the Universe (either expanding or co-moving with the expansion) so that the fluid of dark matter and baryons has a non-zero velocity with which it moves. We first write down the standard equations for a fluid with density ρ and velocity $\mathbf{u}$ in coordinates co-moving with the expansion of the universe. These are

$$\begin{aligned} \frac{\partial \rho}{\partial t} + a^{-1}\nabla(\rho\mathbf{u}) &= 0 && (1)\\ \frac{\partial \mathbf{u}}{\partial t} + a^{-1}\mathbf{u}\cdot\nabla\mathbf{u} + H\mathbf{u} &= -a^{-1}\nabla P - a^{-1}\nabla\phi && (2)\\ a^{-2}\nabla^2\phi &= 4\pi G\left(\rho - \bar{\rho}\right). && (3) \end{aligned}$$

While they may look rather complicated, they are actually quite simple. The first equation is just the continuity equation, rewritten for an expanding universe. The second is the Euler equation including the Hubble drag that slows down peculiar velocities ($H\mathbf{u}$), with a pressure term on the right hand side as well as the gravitational force. The third is simply Poisson's equation that couples the density fluctuations to potential fluctuations. Since the mean density is accounted for already in the evolution of the homogeneous background, which is described by the Friedmann equation(s), we have to subtract it out for the (Newtonian) potential perturbations around the mean expanding universe.

Looking at small perturbations

The next step to make this analytically tractable is to consider fluctuations only, i.e. let's write

$$\rho \equiv \bar{\rho}\left(1+\delta\right), \quad \mathbf{u} = \bar{\mathbf{u}} + \mathbf{v}, \tag{4}$$

where we consider both δ and $\mathbf{v}$ to be small. Note that in the co-moving frame, the gradients and time derivatives of the mean quantities will vanish! Also, we are free to choose $\bar{\mathbf{u}} = 0$ without loss of generality. If we insert them into the full set of equations, we find

$$\begin{aligned} \frac{\partial \delta}{\partial t} + a^{-1}\nabla\left((1+\delta)\mathbf{v}\right) &= 0 && (5)\\ \frac{\partial \mathbf{v}}{\partial t} + a^{-1}\mathbf{v}\cdot\nabla\mathbf{v} + H\mathbf{v} &= -a^{-1}\nabla P - a^{-1}\nabla\phi && (6)\\ a^{-2}\nabla^2\phi &= 4\pi G\bar{\rho}\delta && (7) \end{aligned}$$

Let's now use that $\bar{\rho} = \Omega_m a^{-3}\rho_c = a^{-3}\Omega_m 3H_0^2/8\pi G$ and let's only keep terms that are linear in the perturbation variables (if δ and $\mathbf{v}$ are small, any of their products will be much smaller). We shall also make use of the fact that $\nabla P = \frac{\partial P}{\partial \rho}\nabla \rho = c_s^2$, where c_s is the speed of sound. Then we find

$$\frac{\partial \delta}{\partial t} + a^{-1}\nabla \cdot \mathbf{v} = 0 \tag{8}$$

$$\frac{\partial \mathbf{v}}{\partial t} + H\mathbf{v} = -\frac{c_s^2}{a}\nabla\delta - \frac{1}{a}\nabla\phi \tag{9}$$

$$\nabla^2\phi = \frac{3}{2}H_0^2\Omega_m a^{-1}\delta \tag{10}$$

Note that the advection term for the velocities has completely dropped out. We are left with a set of coupled *linear* partial differential equations. This does not look so much better than the set of coupled *nonlinear* partial differential equations that we started out with, but it is!

Linear partial differential equations become ordinary ones in Fourier space

The key is that the equations are now linear. *For a linear partial differential equations, the method of choice for their solution is Fourier transformation!* This is because Fourier transformation turns differential operators into multiplications.

To see this, let's consider the Fourier representation of a function $f(x)$

$$f(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} \hat{f}(k)\exp\left[ikx\right]\mathrm{d}k \tag{11}$$

$$\frac{\partial}{\partial x}f(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} \hat{f}(k)\frac{\partial}{\partial x}\exp\left[ikx\right]\mathrm{d}k = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} ik\hat{f}(k)\exp\left[ikx\right]\mathrm{d}k, \tag{12}$$

i.e. the pair $f(x) \leftrightarrow \hat{f}(k)$ becomes the pair $\partial_x f(x) \leftrightarrow ik\hat{f}(k)$. This means that if we Fourier transform our set of *linear* partial differential equations, it will become a set of linear ordinary differential equations! Let's do that:

$$\frac{\partial \hat{\delta}}{\partial t} + a^{-1}i\mathbf{k}\cdot\hat{\mathbf{v}} = 0 \tag{13}$$

$$\frac{\partial \hat{\mathbf{v}}}{\partial t} + H\hat{\mathbf{v}} = -\frac{c_s^2}{a}i\mathbf{k}\hat{\delta} - \frac{1}{a}i\mathbf{k}\hat{\phi} \tag{14}$$

$$-k^2\hat{\phi} = \frac{3}{2}H_0^2\Omega_m a^{-1}\hat{\delta}. \tag{15}$$

We see that we can express everything in terms of the velocity divergence $\theta = \nabla\cdot\mathbf{v}, \hat{\theta} = i\mathbf{k}\cdot\hat{\mathbf{v}}$ by multiplying the second equation by $i\mathbf{k}$ to find

$$\dot{\delta} = -a^{-1}\theta \tag{16}$$

$$\dot{\theta} + H\theta = \dot{\theta} - Ha\dot{\delta} = \frac{c_s^2}{a}k^2\delta - \frac{3}{2}H_0^2\Omega_m a^{-2}\delta, \tag{17}$$

where we have omitted the hat for Fourier transforms now. Taking the second time derivative of the first equation yields

$$\ddot{\delta} = -a^{-1}\dot{\theta} + Ha^{-1}\theta = -a^{-1}\dot{\theta} - H\dot{\delta}. \tag{18}$$

Putting it all together

We see that we can combine our results into one single equation

$$\ddot{\delta} + 2H\dot{\delta} + \left(k^2\frac{c_s^2}{a^2} - \frac{3H_0^2\Omega_m}{2a^3}\right)\delta = 0 \tag{19}$$

Since this equation is an ordinary differential equation in time only, it will allow us to separate time and spatial parts as $\delta(\mathbf{x},t) = D(t)\delta(\mathbf{x},t_0)$, which is a feature of such linear partial differential equations. We call the function $D(t)$ the (linear) growth factor. Note that this equation embodies the Jeans stability criterion! Depending on the sign of the term in brackets in the last equation, the solution will be either growing or oscillating in time. We see that there is a critical wave number k_J which separates these two regimes given by

$$k_J = a^{-1}\frac{3H_0^2\Omega_m}{2c_s^2}. \tag{20}$$

In the case of dark matter, $c_s = 0$, so that there are no oscillating solutions. One can show that in that case (and assuming one can neglect the radiation component Ω_r), the growth factor takes the form:

$$D(a) \propto \frac{H(a)}{H_0}\int_0^a \left[\Omega_m a'^{-1} + \Omega_\Lambda a'^2 + (1-\Omega_m-\Omega_\Lambda)\right]^{-3/2} \mathrm{d}a', \tag{21}$$

and one usually chooses $D(t_0) = 1$ at the present time. In the presence of radiation, the full differential equation for D has to be solved.

Exercise 1 (Simple) *Integrate numerically the equation for the growth factor, eq. (21), and plot the solutions for various choices of Ω_m and Ω_Λ. Show that $D(a) = a$ for $\Omega_m = 1, \Omega_\Lambda = 0$.*

Exercise 2 *Re-derive similarly to what we did above, from the full nonlinear fluid equations, the linearised equations for the evolution of a fluid of dark matter and baryons, coupled through gravity. The linearised equations are*

$$\begin{aligned}
\frac{\partial \delta_c}{\partial t} &= -a^{-1}\nabla\cdot\mathbf{v}_c && (22)\\
\frac{\partial \delta_b}{\partial t} &= -a^{-1}\nabla\cdot\mathbf{v}_b && (23)\\
\frac{\partial \mathbf{v}_c}{\partial t} &= -a^{-1}\nabla\phi - H\mathbf{v}_c && (24)\\
\frac{\partial \mathbf{v}_b}{\partial t} &= -a^{-1}\nabla\phi - H\mathbf{v}_b - \frac{c_s^2}{a}\nabla\delta_b && (25)\\
a^{-2}\nabla^2\phi &= 4\pi G\bar{\rho}(f_b\delta_b + f_c\delta_c), && (26)
\end{aligned}$$

where $f_b = \Omega_b/\Omega_m$ and $f_c = 1 - f_b$ are the baryon fraction and cold dark matter fraction. Write a program in Python *that shows the transfer function $T_c(t) = \delta_c(t)/\delta_c(0), T_b(t) = \delta_b(t)/\delta_b(0)$ for various times. Choose appropriate initial conditions yourself. If you want to get fully realistic, the sound speed in the gas is*

$$c_s = \sqrt{\frac{\gamma k T_b}{\mu m_H}}, \tag{27}$$

where $\gamma = 5/3$ for an ideal monoatomic gas, $\mu = 1.22$ is the mean molecular weight of a neutral gas assuming a Helium fraction of 24 per cent, m_H is the mass of the hydrogen atom, and T_b is the kinetic temperature of the baryons. The latter can be approximated from more involved calculations as roughly

$$T_b(a) = \frac{2.726\,\mathrm{K}}{a}\left[1 + \frac{119a}{1+(115a)^{-3/2}}\right]^{-1}. \tag{28}$$

Based on numerical experiments, investigate (1) what is the effect of a higher baryon fraction on $\delta_c(k)$ and $\delta_b(k)$, (2) what is the effect of the sound speed after recombination, (3) what changes if you change the redshift of recombination, (4) describe how the baryon bias $\delta_b(k)/\delta_c(k)$ evolves over time before and after recombination, and (5) can you identify features seen in the CMB spectrum from Figure **??** *in the baryon spectrum $P_b = \delta_b^2(k)$?*

Note that while this exercise shows the emergence of the baryon acoustic oscillation (BAO) in the gravitationally coupled fluid of dark matter and baryons, many important effects are not included, most notably the relativistic effect of different growth rates of perturbations that are larger/smaller than the horizon.

Hints for this Exercise:
The skeleton of a Python *program that numerically integrates the coupled ODEs is given below. You just have to fill in the actual equations and get it to run.*

Hints: Structure of the program

This section provides a skeleton for the time integration of the gravitationally coupled dark matter and baryonic fluid through recombination from Exercise 2. We will use the following Python packages, most of them as usual

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad
from scipy.integrate import odeint
from scipy.integrate import ode
from scipy.special import erf,erfc
```

Next we define the usual constants, in addition the baryon fraction $f_b = \Omega_b/\Omega_m$ and the cold dark matter fraction $f_c = 1 - f_b$.

```python
H0      = 70.0
Omegab  = 0.045
Omegam  = 0.3
Omegal  = 0.7
Omegar  = 8e-5
fb      = Omegab/Omegam # baryon fraction
fc      = 1.0-fb        # CDM fraction
```

The more involved bit is that we need to model the recombination of the plasma and the evolution of the baryon sound speed in some crude way. We assume that prior to recombination, the baryons are tightly coupled to the photons and that the sound speed is $c_s = c/3$. After recombination, we will use the fitting formula given in the exercise to estimate the baryon temperature, and calculate the sound speed from it (we will make this value artificially $20\times$ larger in order to have the effect on larger scales). Then, we can either assume instantaneous recombination at $a = 10^{-3}$, or spread it out artificially thus mocking that it is not instantaneous. The two functions for slow and fast recombination are just that. Altogether we thus have something like

```python
def T_baryon(a):
    # parameterisation of baryon temperature from Tseliakhovich & Hirata 2010
    return 2.726/a/(1+119.*a/(1.0+(115.*a)**-1.5))

def slow_recombination( a, c1, c2 ):
    f = 0.5*(1.0-erf((np.log10(a)+3)*16))
    return f*c1 + (1.0-f)*c2

def fast_recombination( a, c1, c2 ):
    if a<1e-3:
        return c1
    else:
        return c2

def sound_speed(a):
    c1 = clight/3.0
    c0_squared = 0.0112764987
    # for illustration, make post-recomb c_s 20 times larger!!
    c2 = np.sqrt(c0_squared*T_baryon(a)) * 20.0
    cs = slow_recombination(a,c1,c2)
    return cs
```

We will use the `scipy` function `odeint` to integrate the ordinary differential equations. It will take an array of variables `u[...]` describing the current state vector. We will store in it all fields that we want to evolve in time, i.e. we set $\mathbf{u} = (\delta_c, \delta_b, \theta_c, \theta_b)$. The set of coupled ordinary differential equations then takes the form $\mathrm{d}u/\mathrm{d}a = F(\mathbf{u}, a)$ and we will also have to provide the function F taking the right hand side of the differential equations, which is why we will call this function RHS. We will need only one additional parameter (all constants we have as global variables), the wave number k that we want to evolve. The function RHS then has the form

```python
###################################################################
# the RHS of the coupled equations of baryons and dark matter
# u[0] = delta_c
# u[1] = delta_b
# u[2] = theta_c
# u[3] = theta_b
def RHS(u,a,param):
    k = param[0]
    du = np.zeros_like(u)   # make du the same shape as u
    H = Hubble_a(a)          # we will need H(a), so calculate it
    cs = sound_speed(a)   # we will also need the sound speed at a, so get it

    du[0] = ...  # RHS for  d delta_c / da
    du[1] = ...  # RHS for  d delta_b / da
    du[2] = ...  # RHS for  d theta_c / da
    du[3] = ...  # RHS for  d theta_b / da
    return du
```

and returns the vector $\mathbf{du}$ containing the right-hand-sides of the ODEs. The last thing we will need is to set up initial conditions, and write the actual loop over the range of wave numbers k for which we want to evolve $\mathbf{u}$. For the initial conditions, we will simply set $\delta_c(t_0) = 1, \delta_b(t_0) = \theta_c(t_0) = \theta_b(t_0) = 0$, we choose to start at $a = 10^{-10}$ and evolve to some $a = \texttt{astop}$ that we can adjust as we like. For the wave numbers, we will choose 200 values logarithmically spaced $10^{-3} \leq k \cdot \mathrm{Mpc} \leq 10$. We get something like this

```
ainit=1e-10  # the starting scale factor
astop=1e-4   # we want to finish at astop

## setup the initial conditions for CDM density and theta, also for baryon
deltac0 = 1.0
thetac0 = 0.0
deltab0 = 0.0
thetab0 = 0.0

## package all initial conditions into one vector
uinit = np.array([deltac0, deltab0, thetac0, thetab0])

## these will be the wave numbers k for which we compute the result:
kvals = np.logspace(-3,1,200)

## setup vectors for the results deltab(k), deltac(k), ...
deltab = np.zeros_like(kvals)
deltac = np.zeros_like(kvals)
thetab = np.zeros_like(kvals)
thetac = np.zeros_like(kvals)
```

Finally, we need the main loop with the integration. We choose a fixed number of 1000 integration steps in logarithmic time. If we increase to larger values of k this will not be enough. In fact, the correct time step should be chosen based on the sound speed. For simplicity we will simply choose a fixed number and make sure that the result is converged. The main loop then looks like this

```
## set the number of time steps in log time in a rough way
nsteps = 1000 # normally this should depend on k, so check convergence!

## these will be the times at which we evaluate
asteps  = np.logspace(np.log10(ainit),np.log10(astop),nsteps)

## loop over all wave numbers k and evolve them
for i,k in enumerate(kvals):
    ## integrate the system for all ’a’ in ’asteps’,
    u = odeint(RHS, uinit, asteps, args=([k],))

    ## store last result (in time) in the respective results
    deltac[i] = u[-1,0]
    deltab[i] = u[-1,1]
    thetac[i] = u[-1,2]
    thetab[i] = u[-1,3]

print ’Finished integration to a=’,astop
```

This completes the program. We can plot the results and find a k-dependent amplitude for δ_c and δ_b as

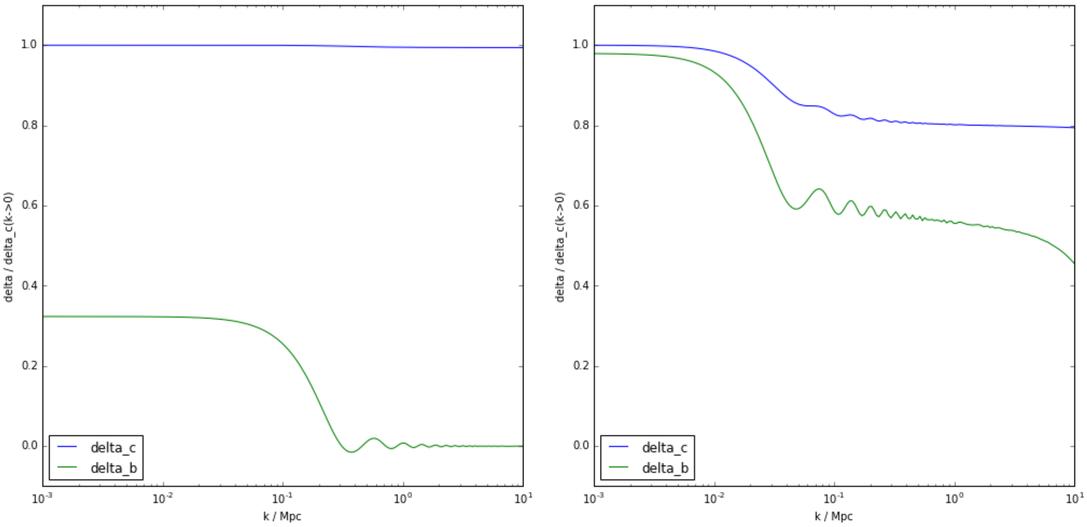

Figure 1: The amplitude of dark matter and baryon density fluctuations prior to recombination at $a = 10^{-4}$ (left) and some time after recombination at $a = 10^{-2}$ calculated with a simple linear perturbation approximation for the coupled fluid. The amplitudes are normalised so that $\delta_c = 1$ for the lowest k that we calculated.

shown in Figure 1 at $a = 10^{-4}$.