ASTR6000: Cosmology (Spring 2026)

Nikhil Padmanabhan

2026-04-09

Table of contents

Preface

This book contains lecture materials for ASTR6000: Cosmology, taught in Spring 2026.

Acknowledgements

I have drawn heavily from two primary textbooks for this class:

- *A Course in Cosmology* by Dragan Huterer
- *Cosmology* by Daniel Baumann

There are many other excellent textbooks, not to mention review articles etc. As we go along, I will try to maintain a running bibliography.

This is a good place to acknowledge my debt to many sources for this class. A large portion of this class will follow the two textbooks and in many, many cases, my presentation and problem sets will directly borrow from them. Citing these sources as I proceed would be quickly tedious, but I will try to point to the relevant sections in the textbooks as we go along.

In cases where I borrow an argument from a different source, I will add this both to the bibliography and class notes.

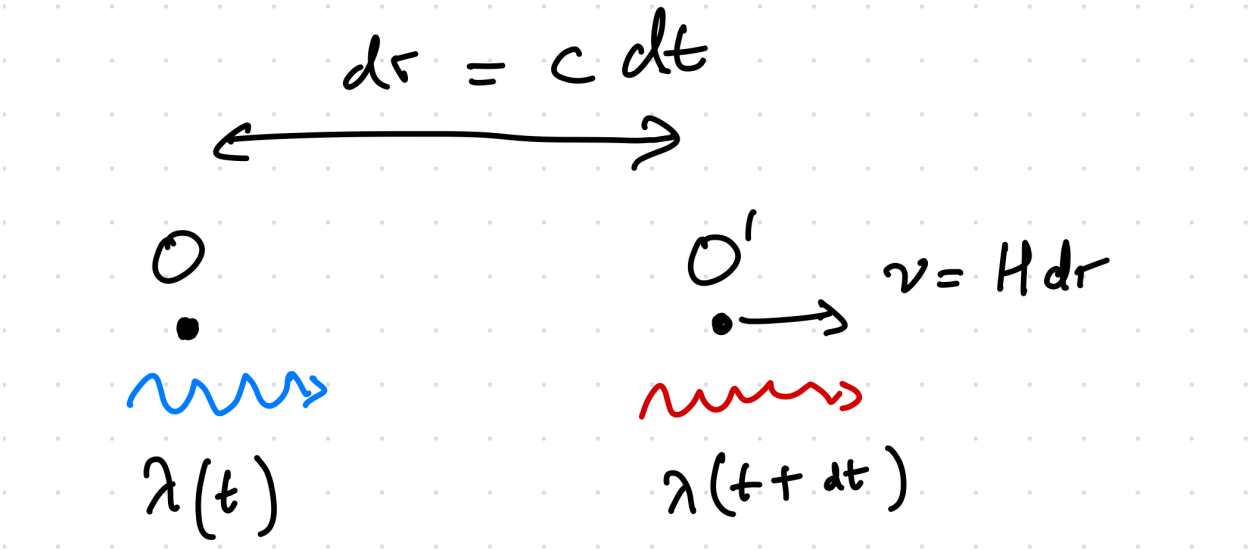
Part I.

The Homogeneous Universe

1. Trajectories in an Expanding Universe

We now consider the trajectories of free particles in an expanding Universe. The most direct and global treatment of this comes by deriving the geodesic equation, which all particles follow. However, one can get interesting intuition by considering local patches (where special relativity is valid), and stitching these together. We consider three such variants below.

1.1. Photons



(This treatment is essentially the same as our derivation of the connection between redshift and the scale factor, although the apparent presentation looks different) Consider a photon emitted by O at time t, and detected by a second observer O' at time $t + dt$. This second observer is therefore a distance cdt away, and must be receding with a velocity of $Hcdt$ by Hubble's law. The doppler shift of the photon is therefore

$$\frac{\lambda(t+dt)}{\lambda(t)} = \sqrt{\frac{1+v}{1-v}}$$

where, for simplicity, we now just use $c = 1$. Since we are assuming infinitesimal intervals, we can Taylor expand both sides to get

$$\frac{\dot{\lambda}\, dt}{\lambda} = v = H\, dt = \frac{\dot{a}}{a} dt$$

from which we get

$$\frac{d\lambda}{\lambda} = \frac{da}{a}$$

or

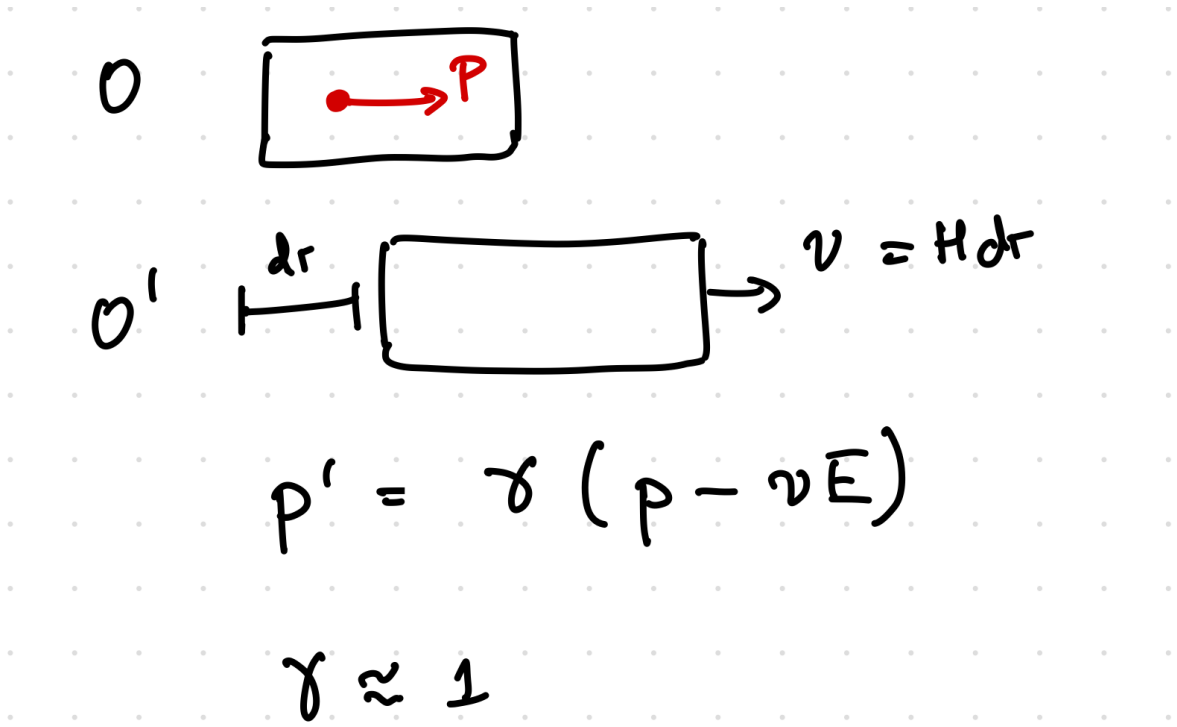
$$\lambda \propto a$$

This is just the definition of redshift, rewritten. Since the momentum and energy of a photon is inversely proportional to its wavelength, this implies

$$E \propto p \propto \frac{1}{a}$$

or that photons lose energy in an expanding Universe.

1.2. Momentum Decay



The figure above sets up the two frames that we want to consider. As before, we consider these to be almost coincident, so that the velocity is very small and $\gamma \approx 1$. Writing down the Lorentz transformation between the two frames, we have

$$p' = \gamma(p - vE)$$

In our limit, this yields

$$\begin{aligned} dp &\approx -vE \\ &= -(H\,dr)E \\ &= -\frac{\dot{a}}{a}(u\,dt)E \\ &= -\frac{\dot{a}}{a}(\frac{p}{E}\,dt)E \end{aligned}$$

where we used the fact that $dr = u\,dt$ with u being the velocity corresponding to the momentum p, and then, $u = p/E$ which is true in both the massless and massive regimes. Putting this all together, we get

$$\frac{dp}{p} = -\frac{da}{a}$$

or

$$p \propto \frac{1}{a}$$

Again, this is a statement that energy/momentum decays away in an expanding Universe. In the case of massive particles, this is purely a simple kinematic effect, with the particle needing to "catch up" with a series of receding observers.

1.3. A Lagrangian Approach

We consider a third approach, using a Lagrangian. Recall

$$S = -m \int ds \quad \leftarrow \text{Lorentz invariant}$$

which, if we limit to one spatial dimension gives us

$$= -m \int \left[1 - a(t)^2 \left(\frac{dx}{dt}\right)^2\right]^{1/2} dt \quad \leftarrow \text{1D motion}$$

from where we can read off the Lagrangian.

$$\mathcal{L} = -m \left[1 - a(t)^2 \left(\frac{dx}{dt}\right)^2\right]^{1/2}$$

We can now compute the conjugate momentum

$$p = \frac{\partial \mathcal{L}}{\partial \dot{x}} \quad \leftarrow \text{conjugate momentum}$$

$$\Rightarrow p = m \frac{1}{\left[1 - a(t)^2 \left(\frac{dx}{dt}\right)^2\right]^{1/2}} a^2 \dot{x}$$

We also observe that p is conserved. We can now solve for $\dot{x}$

$$p^2 \left(1 - a^2 \dot{x}^2\right) = m^2 a^4 \dot{x}^2$$

$$\dot{x}^2 = \frac{p^2}{a^2 \left(p^2 + m^2 a^2\right)}$$

Let us consider the cases of a massless and very massive (non-relativistic) particle.

$m = 0$

$$\dot{x} \propto \frac{1}{a}$$

$$v_{\text{pec}} \sim a\dot{x}$$

$$\sim \text{constant as expected for massless}$$

$m \gg p$

$$\dot{x} \propto \frac{1}{a^2}$$
$$v_{\rm pec} \propto \frac{1}{a} \quad \leftarrow \text{same as before.}$$

2. Expanding the Scale Factor

Let us build some intuition for the scale factor by expanding about the current time t_0 :

$$a(t_1) = a(t_0) + (t_1 - t_0)\dot{a}(t_0) + ...$$
$$\frac{a(t_1)}{a(t_0)} = 1 + (t_1 - t_0)H(t_0) + ...$$

The quantity $t_1 - t_0$ is known as the **lookback time**. We can also write this in terms of redshift

$$\frac{1}{1+z} = 1 + (t_1 - t_0)H(t_0) + ...$$

or

$$z \approx H_0(t_0 - t_1)$$

which shows us how we can relate the lookback time to the redshift (at low redshift/small lookback time). We can also relate this back to Hubble's law by multiplying both sides by c to get

$$v = cz = H_0[c(t_0 - t_1)] = H_0 d$$

We can go to higher orders as well

$$\frac{a(t)}{a(t_0)} = 1 + H_0(t - t_0) - \frac{1}{2}q_0 H_0^2 (t - t_0)^2 + ...$$

where

$$q_0 \equiv -\frac{\ddot{a}}{aH^2}\bigg|_{t=t_0}$$

is the "deceleration parameter". This particular choice reflects the assumption that the expansion rate of the Universe would be decelerating with time and pre-dates the discovery of the accelerated expansion of the Universe and dark energy. Our current best-guess $q_0 \approx -0.5$ (note the negative sign indicating acceleration).

One can invert this expression to get

$$z = H_0(t_0 - t) + \frac{1}{2}(2 + q_0)H_0^2(t_0 - t)^2 + ...$$

or

$$H_0(t_0 - t) = z - \frac{1}{2}(2 + q_0)^2 z^2 + ...$$

We note that these expressions are really only useful at low redshift - we will develop the tools necessary to extend these to higher redshifts later. However, they are very useful to get some intuition for how to interpret the scale factor and redshift.

3. Distances in Cosmology

Measuring the expansion history of the Universe involves measuring distances. In this section, we consider the various distance measures in cosmology. Our starting point is the FLRW metric, here written as

$$ds^2 = -dt^2 + a(t)^2 \left[d\chi^2 + S_k(\chi)^2 d\Omega^2\right]$$

3.1. Comoving Radial Distance

Our starting point is the comoving radial distance χ between points connected by a light ray emitted at t_1 and measured at t_0. Reading from the metric, we have

$$d\chi = c\frac{dt}{a(t)}$$

or

$$\chi = c\int_{t_1}^{t_0} \frac{dt}{a(t)}$$

Note that given $a(t)$, we can easily calculate χ. It is useful to change variables

$$a = \frac{1}{1+z}$$
$$da = -\frac{1}{(1+z)^2}dz$$
$$\implies -\frac{da}{a^2} = dz$$

Changing variables in our definition of χ, we have

$$\chi = c\int \frac{da}{\dot{a}}\frac{1}{a} = c\int_a^1 \frac{da}{a^2 H(a)} = c\int_0^z \frac{dz}{H(z)}$$

where we now just need knowledge of $H(a)$ to compute these distances.

3.2. Proper Distance

The proper distance (calculated at the same time) is

$$r_p = a\chi$$

which is however not directly measurable. When considering angles, it is also convenient to consider

$$d_M = S_k(\chi)$$

which is the factor that shows up in the angular part of the metric. Recall that, for $k = 0$, $S_k(\chi) = \chi$.

3.3. Luminosity Distance

Recall that in Euclidean space, the flux F is related to the luminosity L by

$$F = \frac{L}{4\pi r^2}$$

By analogy, in cosmology, we define

$$F = \frac{L}{4\pi d_L^2}$$

where d_L is the **luminosity distance**. The formula for the flux is modified from the simple Euclidean case by three modifications:

- $r \to d_M$
- Arrival rate is reduced by a factor of $a(t)$
- Photon energy is redshifted by a factor of $a(t)$

Putting this together, we get

$$F = \frac{L}{4\pi d_M^2 (1+z)^2}$$

which gives us

$$d_L = (1+z) d_M$$

If one has a set of astronomical objects with a known luminosity (standard candles) at different redshifts, then by measuring the flux to these objects, one can measure the luminosity distance as a function of redshift, which in turn can be used to determine the Hubble parameter (and as we will see later), dark energy. This technique is what allowed Type 1a SNe to be used to measure dark energy.

3.4. Angular Diameter Distance

Similarly, an object of a physical size D subtends an angle $\delta\theta$ given by

$$\delta\theta = \frac{D}{d_A}$$

where d_A is the angular diameter distance. Reading off the metric, we have

$$\delta\theta = \frac{D}{a(t) d_M}$$

or

$$d_A = \frac{d_M}{(1+z)}$$

This also implies

$$d_L = d_A (1+z)^2$$

Objects of a fixed physical size are standard rulers - measuring their angular extent allows one to measure the angular diameter distance. This forms the basis of the source of the dominant information in the CMB and the BAO technique in dark energy.

3.5. References

An excellent pedagogical reference is

- Hogg, D. W. 1999, , astro-ph/9905116. doi:10.48550/arXiv.astro-ph/9905116

4. The Friedmann Equations

Our discussion previously has laid the groundwork for describing the expansion of the Universe. We can now turn to the dynamics of that expansion - how that expansion is sourced by the various components of the Universe.

The complete derivation of these equations follows from assuming the FLRW metric and a homogeneous stress-energy tensor and substituting these into Einstein's equations. The calculation is straightforward, albeit tedious. However, since we will largely not need that machinery in this class, we will just state the resulting equations and then proceed.

It should be noted that in much of what follows, $c = 1$.

4.1. Introducing the Friedmann Equations

The first Friedmann eqn is

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3}\rho(t) - \frac{k}{R_0^2 a^2}$$

where $\rho(t)$ is the time-dependent energy density of the Universe (including all components - matter, radiation), k is the curvature constant that we defined and R_0 the radius of curvature today.

The second Friedmann eqn is

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3}(\rho(t) + 3P(t))$$

where P is the pressure of the matter component.

Finally, we have the continuity equation

$$\dot{\rho} + 3H(\rho + P) = 0$$

4.1.1. Non-independence

These three equations are not independent. To see this, start with the first Friedmann equation and take a time derivative

$$\begin{aligned}\frac{d}{dt}\left(\frac{\dot{a}}{a}\right)^2 &= 2\left(\frac{\dot{a}}{a}\right)\left[\frac{\ddot{a}}{a} - \frac{\dot{a}^2}{a^2}\right] \\ &= -2\left(\frac{\dot{a}}{a}\right)\left(\frac{12\pi G}{3}(\rho + P) - \frac{k}{R_0^2 a^2}\right) \\ &= \frac{8\pi G}{3}\dot{\rho} + 2\frac{k\dot{a}}{R_0^2 a^3}\end{aligned}$$

which just yields the continuity equation.

4.1.2. Newtonian Derivation

While the complete derivation of the first Friedmann and continuity equations requires GR, we can gain intuition from a partial Newtonian derivation of these. We leave the second Friedmann equation without explicit derivation.

Friedmann I

Consider a sphere of pressureless matter of radius $R_s(t)$ and energy/mass density $\rho(t)$. The mass inside is

$$M(t) = \rho(t)\frac{4}{3}\pi R_s^3(t)$$

The acceleration of this sphere is given by

$$\frac{d^2 R_s}{dt^2} = -G\frac{M(t)}{R_s^2}$$

Multiplying by $\dot{R}_s$ and integrating, we have

$$\frac{1}{2}\left(\frac{d}{dt}R_s\right)^2 = \frac{GM}{R_s} + E$$

where E is the integration constant. Now, writing $R_s = ar_s$ where r_s is the comoving radius (and therefore constant in time), we get

$$\frac{1}{2}r_s^2\dot{a}^2 = \frac{4}{3}\pi G\rho a^2 + E$$

or

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8}{3}\pi G\rho + \frac{2E}{r_s^2}\frac{1}{a^2}$$

which clearly has the form of the Friedmann equation. Note that in the relativistic version, ρ includes all sources of matter/energy. The integration constant here gets identified with the curvature:

- $E > 0$, $k = -1$, Universe expands forever
- $E < 0$, $k = 1$, maximum size
- $E = 0$, $k = 0$, stops in the far future. Note that these conclusions are in the absence of dark energy, which changes this discussion.

Why does this work at all? Birkhoff's theorem helps here.

Continuity Equation

Start from the first law of thermodynamics

$$dQ = dE + PdV$$

If we consider our previous sphere and adiabatic expansion ($dQ = 0$), this gives us

$$\dot{E} + P\dot{V} = 0$$

Now

$$\begin{aligned} E &= \rho V \\ \dot{E} &= \dot{\rho}V + \rho\dot{V} \\ \dot{\rho}V + \dot{V}(\rho + P) &= 0 \\ V = \frac{4}{3}\pi r_s^3 a^3 \implies \dot{V} &= 3\frac{\dot{a}}{a}V = 3HV \\ \dot{\rho} + 3H(\rho + P) &= 0 \end{aligned}$$

4.2. The Critical Density

Consider the first Friedmann eqn

$$\left(\frac{\dot{a}}{a}\right)^2 = H^2 = \frac{8\pi G}{3}\rho(t) - \frac{k}{R_0^2 a^2}$$

Dividing by H^2, we have

$$1 = \frac{8\pi G}{3H^2}\rho(t) - \frac{k}{R_0^2 a^2 H^2}$$

Notice that when

$$\rho = \frac{3H^2}{8\pi G}$$

the curvature k is zero. We therefore define a **critical density**

$$\rho_{\text{crit}} = \frac{3H^2}{8\pi G}$$

Note that the value of the critical density depends on H and therefore varies with time. The value today is

$$\begin{aligned} \rho_{\text{crit}} &= 1.877 \times 10^{-26} h^2 \text{kg m}^{-3} \\ &= 2.775 \times 10^{11} h^2 M_\odot \text{Mpc}^{-3} \\ &= 1.053 \times 10^{10} \text{eV}^4 \end{aligned}$$

We can measure the density relative to the critical density

$$\Omega = \frac{\rho}{\rho_{\text{crit}}}$$

Here, we're keeping Ω to represent all energy density, but very soon, we will write down these Ω's as a function of the type - radiation, matter, dark energy.

Plugging back into the Friedmann equation, we find

$$1 - \Omega(t) = -\frac{k}{R_0^2 a^2 H^2}$$

Note that the sign of $1-\Omega$ is fixed and doesn't change with time. In particular, if $k = 0$, $\Omega(t) = 1$.

5. Energy Components of the Universe with Time

Let us pick up again with the Friedmann equation

$$\left(\frac{\dot{a}}{a}\right)^2 = H^2 = \frac{8\pi G}{3}\rho(t) - \frac{k}{R_0^2 a^2}$$

To make progress, we need to know ρ as a function of a. We can get this from the continuity equation

$$\dot{\rho} + 3H(\rho + P) = 0$$

We need to specify what P is - we do so via an equation of state

$$P = w\rho$$

where w need not be constant (but is for many single components of the Universe). Substituting into the continuity equation, we have

$$\dot{\rho} = -3\frac{\dot{a}}{a}(1 + w(a))\rho$$

or

$$\rho = \rho_0 \exp\left(-3\int_0^{\ln a}(1 + w(a))\, d\ln a\right)$$

Now, standard energy components have constant w

- Pressure-less Matter/Dust

$$w = 0 \implies \rho = \rho_0 a^{-3}$$

- Radiation

$$w = \frac{1}{3} \implies \rho = \rho_0 a^{-4}$$

- Cosmological Constant

$$w = -1 \implies \rho = \rho_0$$

6. Single Component Universes

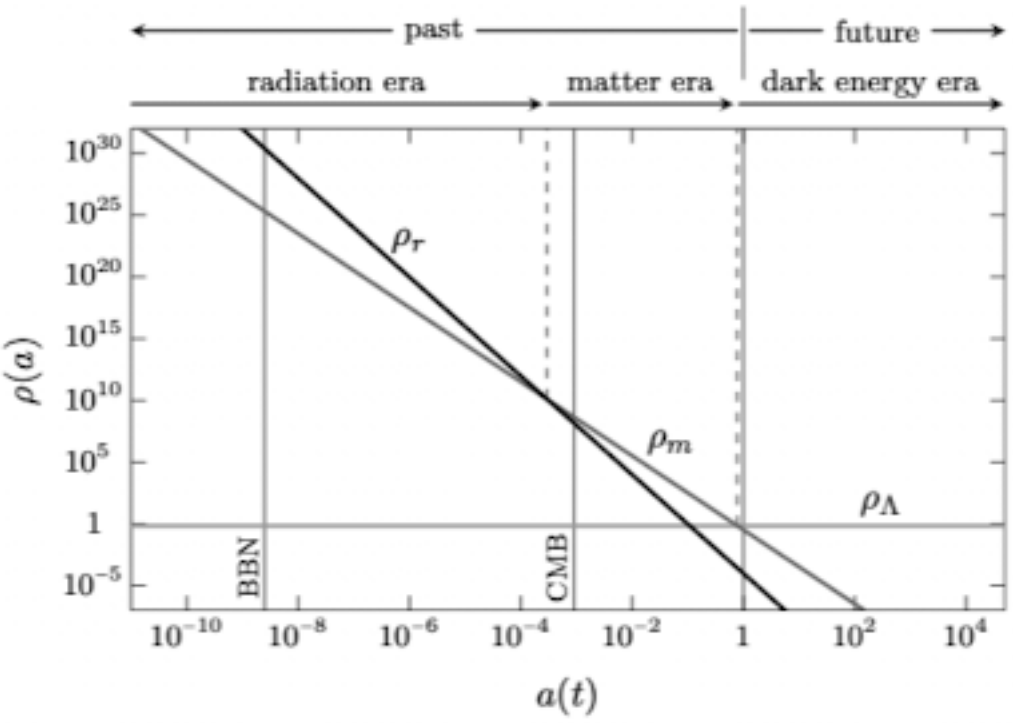


Figure 6.1.: Evolution of Energy Densities (Baumann Fig. 2.10)

We previously considered the evolution of the density of different components of the Universe. The figure above (from Baumann) shows that, for large parts of the history of the Universe, a single component was dominant. It is therefore quite useful to work through single component Universes.

Note that we treat curvature as one of these components, and so, we will set $k = 0$ in all of what we consider here (the case of curvature is an empty Universe; we do not consider this here, but may in a homework assignment).

There are three components that are largely relevant to modern cosmology today

- matter : $w = 0$
- radiation : $w = \frac{1}{3}$
- cosmological constant : $w = -1$

What we will do here is consider a single component with a constant equation of state $w \neq -1$. The case of $w = -1$ is special and does not follow from these; we leave this up to the reader. The values for matter and radiation can then be read off these results.

The Friedmann equation reads

$$\left(\frac{\dot{a}}{a}\right)^2 = H_0^2 a^{-3(1+w)}$$

where we use the fact that the density must be the critical density. This gives

$$\frac{\dot{a}}{a} = H_0 a^{-3(1+w)/2}$$

or

$$\frac{da}{a^{-1/2-3w/2}} = H_0 dt$$

Integrating, we get

$$t = \frac{1}{H_0}\frac{2}{3(1+w)} a^{3(1+w)/2}$$

or

$$a(t) = \left(\frac{t}{t_0}\right)^{2/(3(1+w))}$$

It follows from this that the age of the Universe is

$$t_0 = \frac{1}{H_0}\frac{2}{3(1+w)}$$

We can also calculate the comoving distance $\chi(z)$

$$\chi = c\int_0^z \frac{1}{H(z)}\,dz = \frac{c}{H_0}\int_0^z \frac{dz}{(1+z)^{3(1+w)/2}}$$

which gives

$$\chi(z) = \frac{c}{H_0}\frac{2}{1+3w}\left[1 - \frac{1}{(1+z)^{(1+3w)/2}}\right]$$

The particle horizon is (how far back/out we can see)

$$\chi_{\rm part} = \frac{c}{H_0}\frac{2}{1+3w}$$

7. Multi-Component Universes

7.1. The Friedmann Equation

Let's start from the first Friedmann equation

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3}\rho(t) - \frac{k}{R_0^2 a^2} + \frac{\Lambda}{3}$$

where I have now also included the cosmological constant term into the mix. It is conceptually useful to include the last two terms on the right as density terms. Note that this is a convenience, especially in the case of curvature, which isn't a energy density. However, let us define :

$$\rho_k(a) = -\frac{3k}{R_0^2 a^2 8\pi G}$$
$$\rho_\Lambda(a) = \frac{\Lambda}{8\pi G}$$

where we have kept the time/scale factor dependence explicit. The Friedmann equation can now be written as

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3}(\rho_m(a) + \rho_r(a) + \rho_k(a) + \rho_\Lambda(a))$$

where we have now explicitly broken out the matter and radiation components explicitly. We could also generalize the cosmological constant to be a general dark energy term with $w \neq 1$ and get

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3}(\rho_m(a) + \rho_r(a) + \rho_k(a) + \rho_{DE}(a))$$

Multiplying and dividing by H_0^2 on the right, and now explicitly putting in the scale factor dependence into the equation, we find

$$H(a)^2 = \frac{H_0^2}{\rho_{\mathrm{crit},0}}(\rho_{m,0}a^{-3} + \rho_{r,0}a^{-4} + \rho_{k,0}a^{-2} + \rho_{DE,0}a^{-3(1+w)})$$

A few comments are necessary here:

- The subscript 0 refers to the present day value of the quantity.
- For dark energy, this expression is only correct for a constant equation of state. For a time-varying equation of state, you would need to go back to the continuity equation to work out the variation with scale factor. The case of the cosmological constant is $w = -1$.

If we now define Ω for each individual component X

$$\Omega_X(a) \equiv \frac{\rho_X(a)}{\rho_{\mathrm{crit}}(a)}$$

we have

$$H(a)^2 = H_0^2 \left[\Omega_{m,0} a^{-3} + \Omega_{r,0} a^{-4} + \Omega_{k,0} a^{-2} + \Omega_{DE,0} a^{-3(1+w)}\right]$$

or

$$H(z)^2 = H_0^2 \left[\Omega_{m,0}(1+z)^3 + \Omega_{r,0}(1+z)^4 + \Omega_{k,0}(1+z)^2 + \Omega_{DE,0}(1+z)^{3(1+w)}\right]$$

These forms of the Friedmann equations are the forms that get used the most in cosmology - well worth committing to memory!

If, instead of dividing by H_0^2, we divided by $H(a)^2$, then we recover a very useful relationship

$$\sum_i \Omega_i(a) = 1$$

We note that this includes Ω_k as well, unlike our discussion earlier. In what follows, we'll continue to treat curvature as having its corresponding Ω, instead of treating it separately.

7.2. Distances and Ages

We can now write out the integrals to compute distances and ages. We have

$$\chi = c \int_0^z \frac{dz}{H(z)} = \frac{c}{H_0} \int_0^z \frac{1}{E(z)}\, dz$$

where

$$E(z) \equiv \left[\Omega_{m,0}(1+z)^3 + \Omega_{r,0}(1+z)^4 + \Omega_{k,0}(1+z)^2 + \Omega_{DE,0}(1+z)^{3(1+w)}\right]^{1/2}$$

Similarly, the time integral yields

$$t = \frac{1}{H_0} \int_0^a \frac{1}{aE(a)}\, da = \frac{1}{H_0} \int_{-\infty}^{\ln a} \frac{1}{E(a)}\, d\ln a$$

which gives us the age of the Universe

$$t_0 = \frac{1}{H_0} \int_0^1 \frac{1}{aE(a)}\, da = \frac{1}{H_0} \int_{-\infty}^{0} \frac{1}{E(a)}\, d\ln a$$

A note on units here. As discussed earlier, the uncertainty in the Hubble constant is often parametrized as $H_0 = 100h\,\mathrm{km/s/Mpc}$, which results in an h^{-1} factor in both distances and times.

In general, these integrals must be done numerically, but there are some useful analytic results in the case of two-component Universes. Baumann discusses these in his textbook, and I'd encourage you to explore these there.

7.3. Ω variation with time

If component X has a constant equation of state w, we can write out how it varies with time

$$\Omega_X(z) = \frac{\Omega_{X,0}(1+z)^{3(1+w)}}{E(z)^2}$$

7.4. Our Fiducial Model

The table presents our fiducial, standard model of cosmology. The results here are very similar to the legacy results from the Planck satellite, rounded up. These were derived from here, which is a large table exploring many different options. I am using the values from the `base_plikHM_TT_lowl_lowE_post_BAO` chain (rounded up), which sets the curvature contribution to zero. You may explore other variants (this file should be posted to Canvas). All of these are values today.

Parameter	Value
Ω_m	0.31
Ω_Λ	0.69
$\Omega_b h^2$	0.02225
H_0	70 km/s/Mpc**
$T_{\rm CMB}$	2.7255 K
$\Omega_\gamma h^2$	2.473×10^{-5}
$\Omega_\nu h^2$	1.68×10^{-5}
$\Omega_r h^2$	4.15×10^{-5}
Ω_k	0

One caveat : I've rounded the Hubble constant here, but there is tension between high and low redshift estimates of the Hubble constant as of this writing.

7.5. The Flatness Problem

The above table fixed the curvature to be flat (i.e. $k = 0$). If one attempts to constrain it, we find $\Omega_{k,0} = 0.001 \pm 0.002$. We can now ask how this value changes as we go to the very early Universe. We find

$$\Omega_k(z) \to \frac{\Omega_{k,0}}{\Omega_{r,0}} \frac{(1+z)^2}{(1+z)^4} \text{ as } z \to \infty$$

If we assume the best-fit non-zero value, then we find that Ω_k must have been vanishingly small but non-zero in the very early Universe. The challenge is to find the physics that would force the curvature to almost perfectly vanish in the early Universe - this is the **flatness problem** that gets solved by inflation.

8. Numerically Integrating the Friedmann Equation

As we mentioned earlier, one is often forced to numerically solve for quantities in a multi-component Universe. This notebook demonstrates this for a few illustrative cases.

A few notes about the notebooks here. These are meant to be self-contained and do not depend on additional files. As a result, certain function definitions might get repeated throughout.

Additionally, these are not meant to be robust software, but rather exploratory code, so in cases, we may use global variables, or hard-code in particular choices.

Finally, we will try to keep the requirements for these notebooks to the minimum. Hopefully, running these on eg. Google Colab would be straightforward.

8.1. The Fiducial Cosmology

We'll start by defining a fiducial cosmology that we will use in many of the examples below. As an implementation detail, we'll store the parameters in a dictionary, and will specify the parameters at the present epoch. We'll also specify some convenient constants.

We'll work assuming $H_0 = 100\text{h}$ km/s/Mpc, when necessary, but otherwise will just work in c/H_0 units for distances and $1/H_0$ units for time.

8.2. Useful Functions

Here we define a few helper functions that will be useful below.

In order to account for curvature, we need to compute the radius of curvature today, which is given by:

$$R_c = \frac{c/H_0}{\sqrt{|\Omega_k|}}$$

where $\Omega_k = 1 - \Omega_m - \Omega_r - \Omega_\Lambda$. Note that this follows from the Friedmann equation

$$1 - \Omega_0 = -\frac{c^2 k}{R_0^2 H_0^2}$$

and then working in units of c/H_0 for distances.

While this diverges for a flat Universe, we don't actually ever need to compute it in that case.

8.3. Distances

We can now calculate the cosmological distances by numerically integrating

$$\chi(z) = \int_0^z \frac{dz'}{E(z')}$$

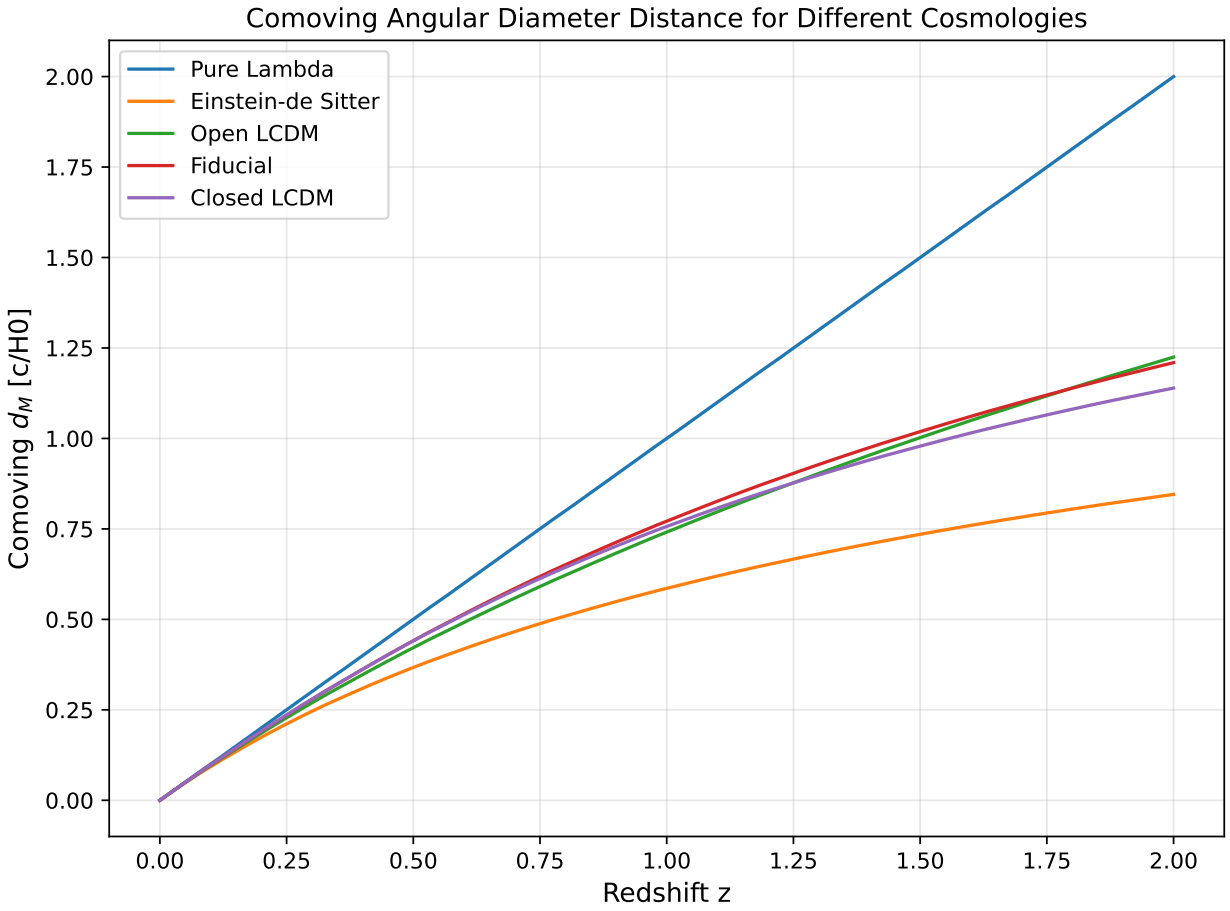
For simplicity, we just use the trapezoidal rule here and the corresponding `numpy` function.

We could use more sophisticated integration methods, and when plotting, we could just accumulate the integral as we go along to avoid repeated calculations. We leave these and other improvements as exercises to the reader.

Now let's plot $d_M = S_k(\chi)$ for our fiducial cosmology, but also for some additional choices :

- "Pure Lambda": $\Omega_m = 0.0, \Omega_\Lambda = 1.0$
- "Einstein-de Sitter": $\Omega_m = 1.0, \Omega_\Lambda = 0.0$
- "Open LCDM": $\Omega_m = 0.2, \Omega_\Lambda = 0.3$
- "Fiducial": our fiducial cosmology defined above.
- "Closed LCDM": $\Omega_m = 0.4, \Omega_\Lambda = 0.8$

This figure is comparable to Fig. 3.2 in Huterer.



8.4. Ages

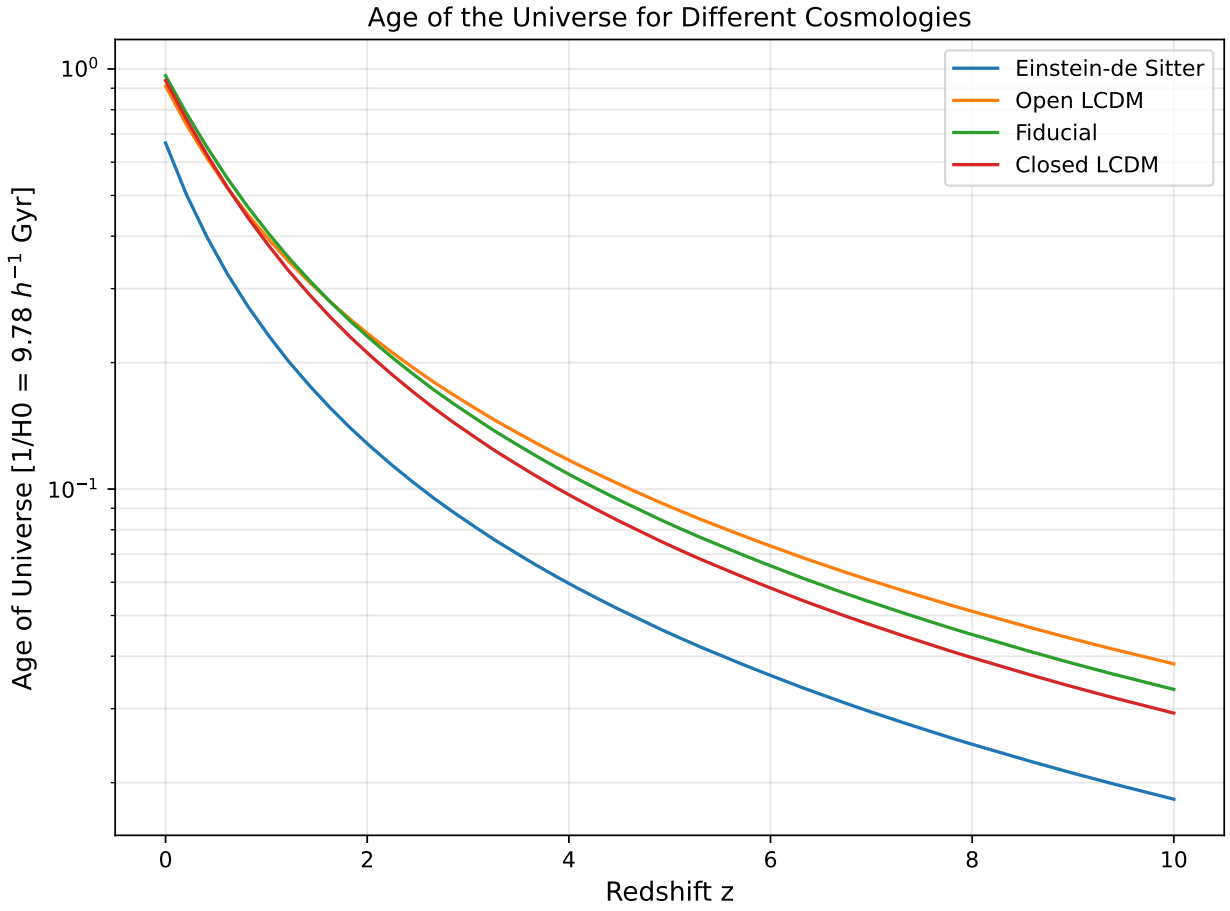
Next, we can compute the age of the Universe at a given redshift by integrating

$$t(a) = \int_{-\infty}^{\log a} \frac{d \log a'}{H(a')}$$

where $H(a) = H_0 E(z)$ as defined above, and $a = 1/(1+z)$.

We'll approximate the integral using the trapezoidal rule again, using a large negative number for the lower limit.

Note that we remove the "Lambda" cosmology here, since its age diverges.



9. Numerically Integrating the Friedmann Equation with JAX and Diffrax

This notebook demonstrates numerical solution of cosmological integrals using JAX and Diffrax, reformulating them as ordinary differential equations (ODEs). This approach enables automatic differentiation of cosmological quantities with respect to parameters, which is useful for parameter inference and sensitivity analysis.

We'll compute the same quantities as in the basic numerical integration notebook but using a modern differentiable programming framework. This allows us to:

- Use adaptive ODE solvers (Dormand-Prince 5(4)) with automatic step control
- Compute derivatives with respect to cosmological parameters via autodiff
- Validate against analytical solutions for single-component universes

A few notes about the notebooks here. These are meant to be self-contained and do not depend on additional files. As a result, certain function definitions might get repeated throughout.

Additionally, these are not meant to be robust software, but rather exploratory code, so in cases, we may use global variables, or hard-code in particular choices.

9.1. Running on Google Colab

To run this notebook on Google Colab, you'll need to install JAX and Diffrax:

```
!pip install jax jaxlib diffrax
```

For more details on JAX installation options, see: https://jax.readthedocs.io/en/latest/installation.html

9.2. The Fiducial Cosmology

We'll start by defining a fiducial cosmology that we will use in many of the examples below. As an implementation detail, we'll store the parameters in a dictionary, and will specify the parameters at the present epoch. We'll also specify some convenient constants.

We'll work assuming $H_0 = 100\text{h}$ km/s/Mpc, when necessary, but otherwise will just work in c/H_0 units for distances and $1/H_0$ units for time.

9.3. Useful Functions

Here we define a few helper functions that will be useful below. These are rewritten to use JAX numpy (`jnp`) instead of regular numpy to enable automatic differentiation.

In order to account for curvature, we need to compute the radius of curvature today, which is given by:

$$R_c = \frac{c/H_0}{\sqrt{|\Omega_k|}}$$

where $\Omega_k = 1 - \Omega_m - \Omega_r - \Omega_\Lambda$. Note that this follows from the Friedmann equation

$$1 - \Omega_0 = -\frac{c^2 k}{R_0^2 H_0^2}$$

and then working in units of c/H_0 for distances.

While this diverges for a flat Universe, we don't actually ever need to compute it in that case.

Important for JAX: The `Sk` function uses `jnp.where` instead of if/elif/else to ensure compatibility with JAX's tracing and automatic differentiation. JAX requires that control flow be expressed in a way that can be differentiated.

9.4. Distances with Diffrax

We can calculate the comoving radial distance $\chi(z)$ by reformulating the integral as an ODE initial value problem:

$$\frac{d\chi}{dz} = \frac{1}{E(z)}, \quad \chi(0) = 0$$

This is equivalent to computing:

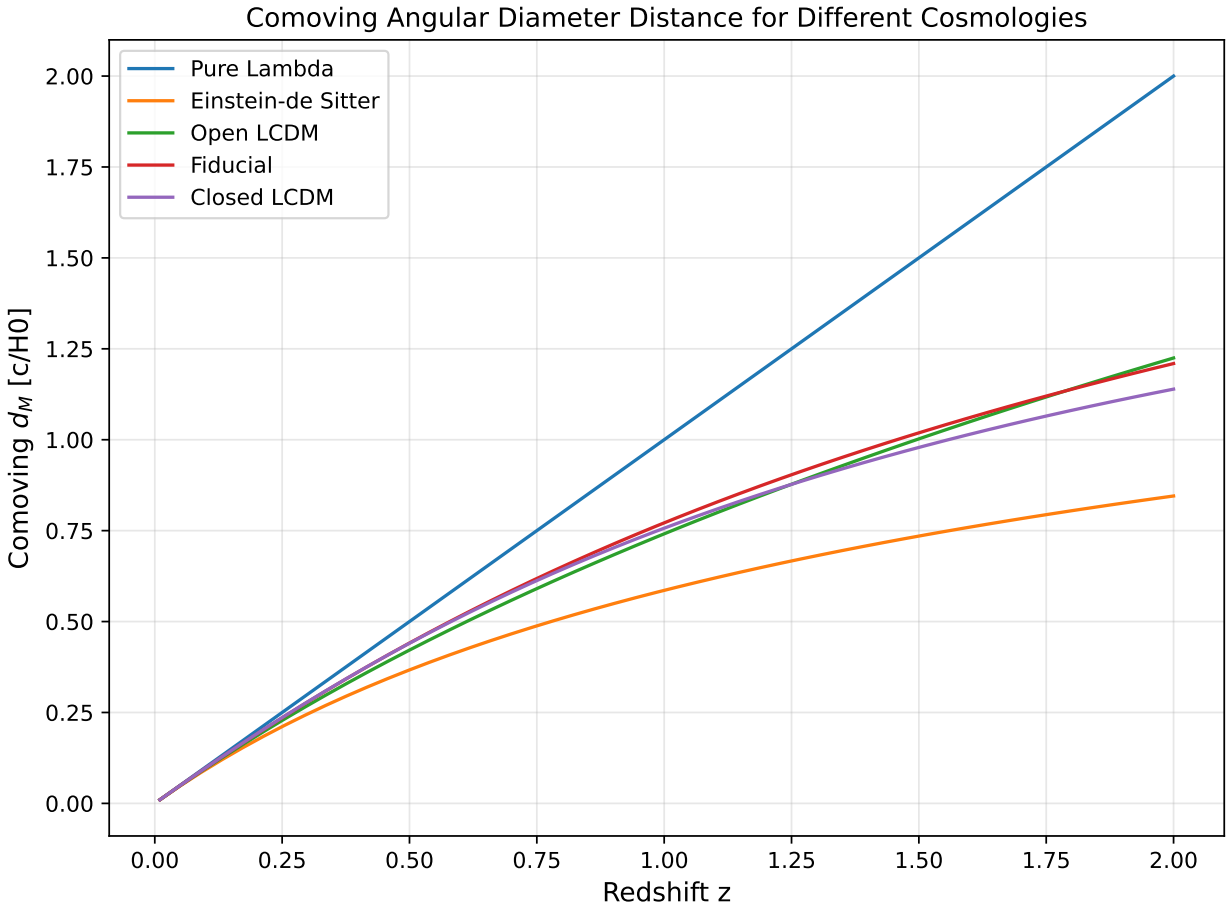
$$\chi(z) = \int_0^z \frac{dz'}{E(z')}$$

We use Diffrax's `diffeqsolve` with the Dopri5 solver (Dormand-Prince 5(4)), which provides adaptive step size control and good accuracy for smooth problems.

Now let's plot $d_M = S_k(\chi)$ for our fiducial cosmology, but also for some additional choices :

- "Pure Lambda": $\Omega_m = 0.0, \Omega_\Lambda = 1.0$
- "Einstein-de Sitter": $\Omega_m = 1.0, \Omega_\Lambda = 0.0$
- "Open LCDM": $\Omega_m = 0.2, \Omega_\Lambda = 0.3$
- "Fiducial": our fiducial cosmology defined above.
- "Closed LCDM": $\Omega_m = 0.4, \Omega_\Lambda = 0.8$

This figure is comparable to Fig. 3.2 in Huterer.



9.5. Ages with Diffrax

Next, we compute the age of the Universe at a given redshift by reformulating the integral as an ODE:

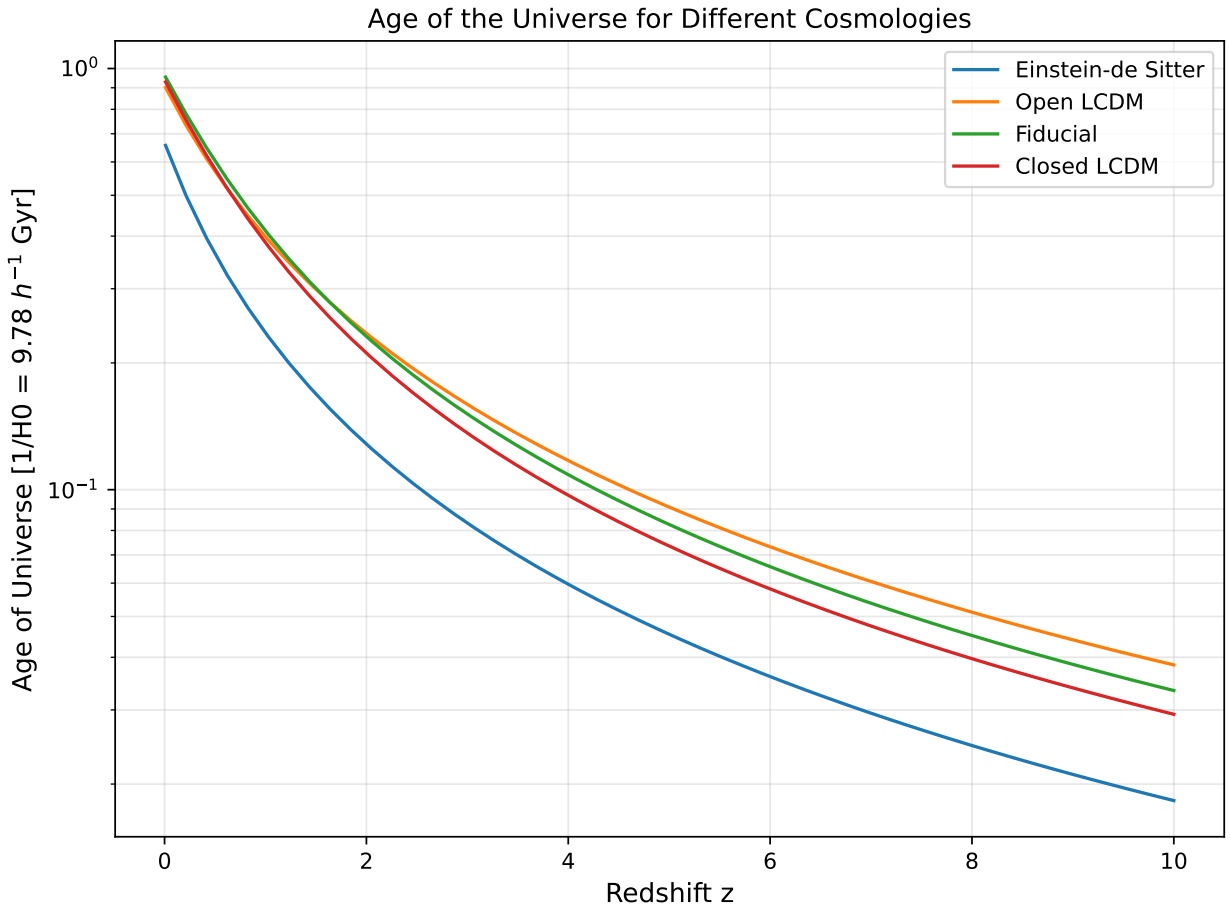
$$\frac{dt}{d\log a} = \frac{1}{E(z(a))}, \quad t(\log a \to -\infty) \approx 0$$

where $a = 1/(1+z)$ is the scale factor and $E(z) = H(z)/H_0$ as defined above. This is equivalent to:

$$t(a) = \int_{-\infty}^{\log a} \frac{d\log a'}{H(a')}$$

We approximate the integral by starting at a large negative value of $\log a$ (corresponding to very early times).

Note that we remove the "Pure Lambda" cosmology here, since its age diverges.



9.6. Automatic Differentiation with JAX

One of the key advantages of using JAX is the ability to compute derivatives of our cosmological quantities with respect to the input parameters using automatic differentiation. This is particularly useful for:

- Parameter inference and optimization
- Computing Fisher information matrices
- Sensitivity analysis

Here we demonstrate computing derivatives of the comoving distance with respect to Ω_m and Ω_λ at a fixed redshift.

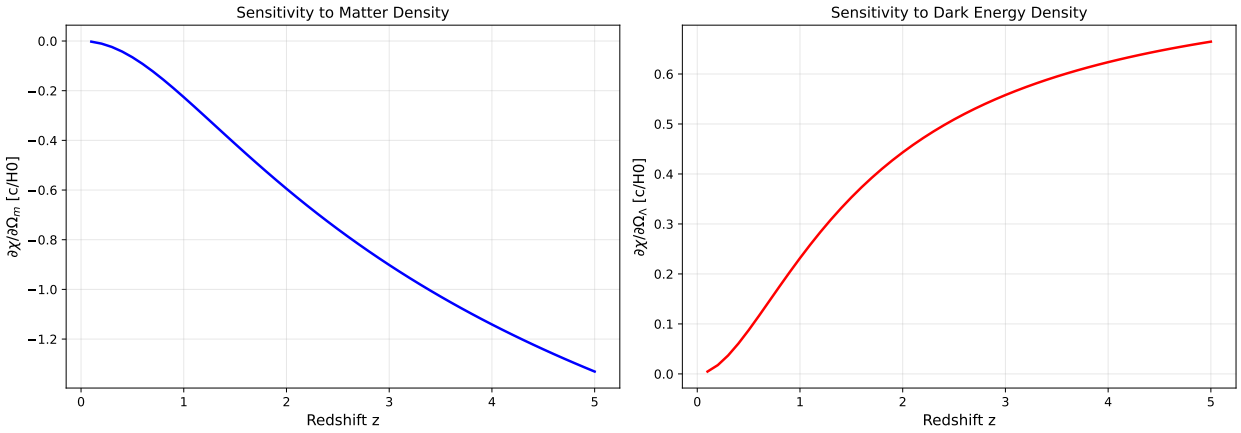
For a more complete and production-ready implementation of differentiable cosmology, see the JAX-COSMO library: https://github.com/DifferentiableUniverseInitiative/jax_cosmo

```
Derivatives of comoving distance chi(z) at fiducial cosmology:
======================================================================
       z      chi [c/H0]           / Ωm           / Ω
----------------------------------------------------------------------
```

```
0.5         0.440984        -0.065449       0.087409
1.0         0.771427        -0.226072       0.231644
2.0         1.209471        -0.594746       0.443205
5.0         1.815509        -1.330786       0.664718
```

9.6.1. Derivatives vs Redshift

Let's visualize how these derivatives change as a function of redshift.



9.6.2. Physical Interpretation

The derivatives tell us how sensitive the comoving distance is to changes in the cosmological parameters:

- $\partial\chi/\partial\Omega_m < 0$: Increasing matter density slows expansion, reducing distances at fixed redshift.
- $\partial\chi/\partial\Omega_\Lambda > 0$: Increasing dark energy density accelerates expansion, increasing distances at fixed redshift.

The magnitude of these derivatives increases with redshift, indicating that high-redshift observations are more sensitive to cosmological parameters. This is why surveys targeting high-redshift objects (like Type Ia supernovae at z $>$ 1 or the CMB at z $\sim$ 1100) are so powerful for constraining cosmology.

These derivatives are the building blocks for computing Fisher information matrices, which predict the parameter constraints achievable from a given dataset. In Bayesian inference, these gradients can be used for efficient gradient-based sampling methods like Hamiltonian Monte Carlo.

9.7. Analytical Validation

To verify our numerical integration, we compare against known analytical solutions for single-component flat universes. For a universe dominated by a single component with equation of state $w \neq -1$, the comoving distance and age have closed-form expressions.

9.7.1. Matter-Dominated Universe ($\Omega_m = 1, w = 0$)

For a flat matter-only universe:

$$\chi(z) = \frac{c}{H_0}\frac{2}{1+3w}\left[1 - \frac{1}{(1+z)^{(1+3w)/2}}\right] = \frac{2c}{H_0}\left[1 - (1+z)^{-1/2}\right]$$

$$t_0 = \frac{1}{H_0}\frac{2}{3(1+w)} = \frac{2}{3H_0}$$

9.7.2. Radiation-Dominated Universe ($\Omega_r = 1, w = 1/3$)

For a flat radiation-only universe:

$$\chi(z) = \frac{c}{H_0}\frac{2}{1+3w}\left[1 - \frac{1}{(1+z)^{(1+3w)/2}}\right] = \frac{c}{H_0}\left[1 - (1+z)^{-1}\right]$$

$$t_0 = \frac{1}{H_0}\frac{2}{3(1+w)} = \frac{1}{2H_0}$$

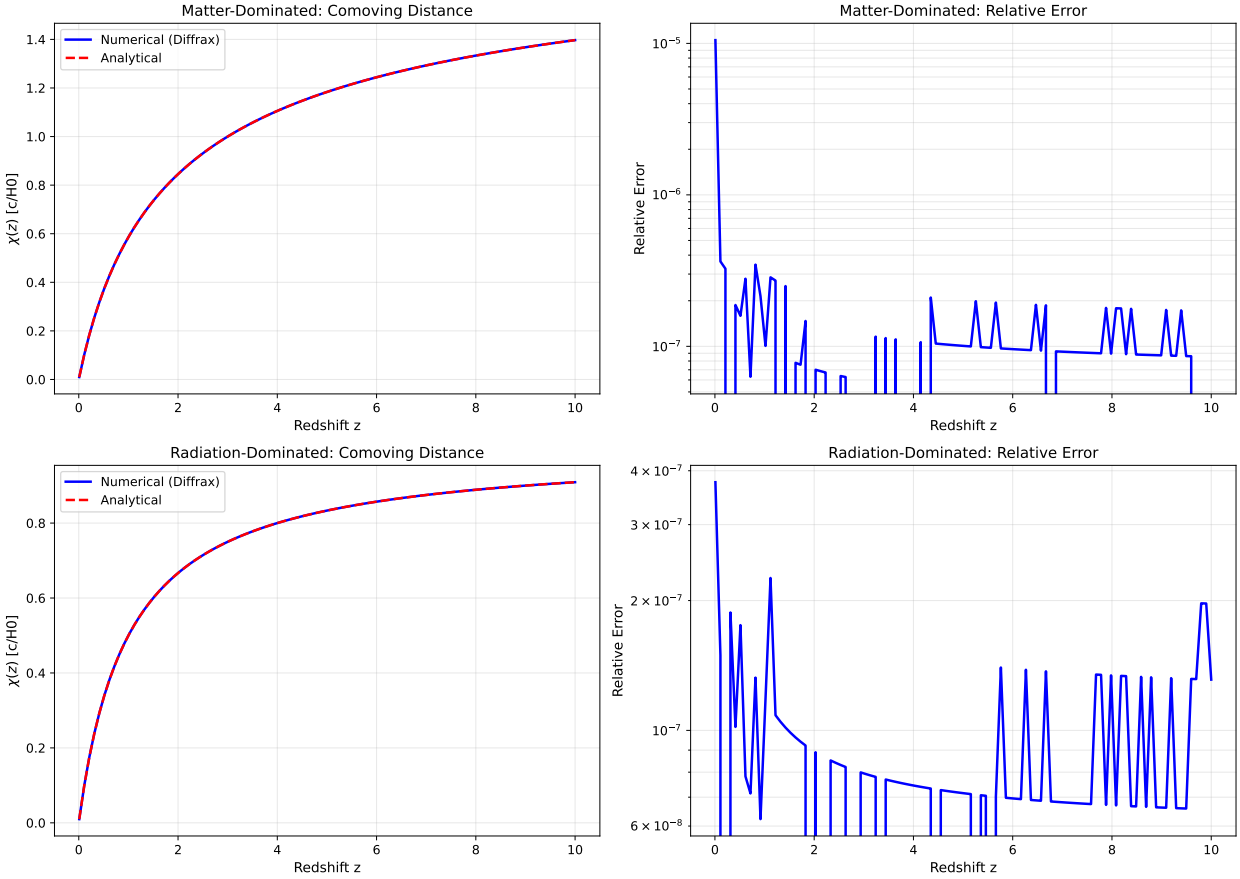
Let's implement these analytical solutions and compare.

```
Matter-Dominated Universe: Comoving Distance Validation
================================================================================
       z       Numerical      Analytical      Rel. Error
--------------------------------------------------------------------------------
     0.5    0.3670068681    0.3670068979        8.12e-08
     1.0    0.5857865214    0.5857864618        1.02e-07
     2.0    0.8452994823    0.8452994823        0.00e+00
     5.0    1.1835035086    1.1835033894        1.01e-07
    10.0    1.3969775438    1.3969773054        1.71e-07

================================================================================
Age at z=0: Numerical = 0.6666666269, Analytical = 0.6666666667
Relative Error: 8.94e-08
================================================================================


Radiation-Dominated Universe: Comoving Distance Validation
================================================================================
       z       Numerical      Analytical      Rel. Error
--------------------------------------------------------------------------------
     0.5    0.3333333731    0.3333333135        1.79e-07
     1.0    0.5000000596    0.5000000000        1.19e-07
     2.0    0.6666666865    0.6666666269        8.94e-08
     5.0    0.8333333731    0.8333333135        7.15e-08
    10.0    0.9090909958    0.9090908766        1.31e-07
```

```
================================================================================
Age at z=0: Numerical = 0.5000000000, Analytical = 0.5000000000
Relative Error: 0.00e+00
================================================================================
```

9.7.3. Visualization: Numerical vs Analytical

Let's create overlay plots to visually compare the numerical and analytical solutions.



9.7.4. Summary

The numerical integration using Diffrax agrees with the analytical solutions to very high precision (relative errors typically $< 10^{-6}$). This validates our implementation and demonstrates that:

1. The ODE formulation correctly represents the cosmological integrals
2. The Dopri5 solver with adaptive stepping provides excellent accuracy
3. JAX's automatic differentiation infrastructure works seamlessly with the ODE solver

The small residual errors are primarily due to:

- Finite tolerance settings (rtol=1e-8, atol=1e-10)

- Finite starting point for the age integral (log a = -20 instead of -∞)

These can be reduced further by tightening tolerances or starting the age integration at an even earlier time, at the cost of additional computation.

10. Standard Candles

Recall the relation between flux and luminosity

$$f = \frac{L}{4\pi d_L^2}$$

It follows directly from this that if one knew the luminosity of an object, then a measurement of the flux would give you the luminosity distance. Or if you had a class of objects with the same (possibly unknown) luminosity, then you get relative distances

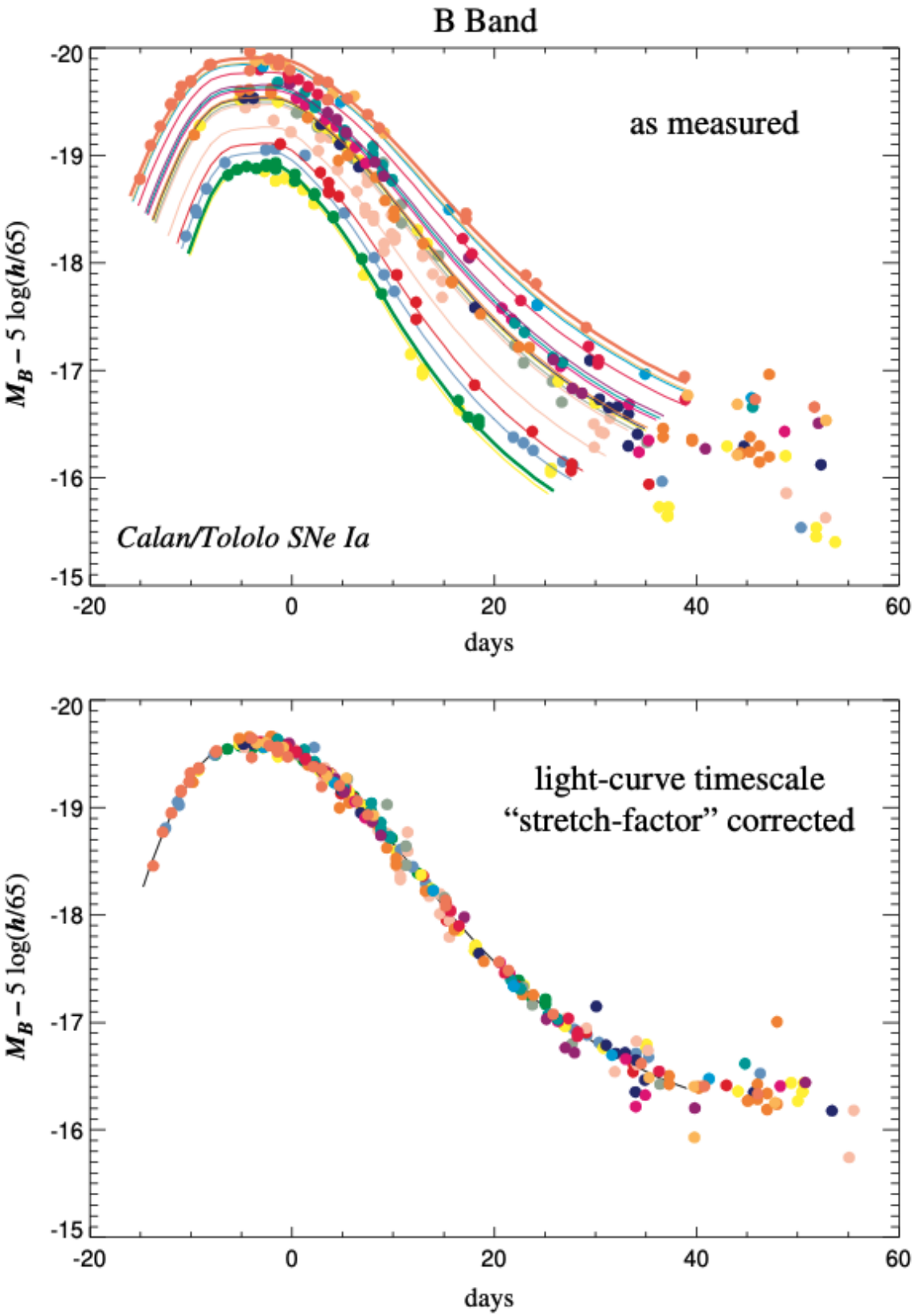
$$\frac{f_1}{f_2} = \frac{d_{L,1}^2}{d_{L,2}^2}$$

And given luminosity distance measurements, one can infer cosmological parameters.

10.1. Type 1a SNe

Type 1a SNe are probably the best known standard candles, especially in the context of dark energy where they led to the discovery of dark energy in 1998 and the 2011 Nobel Prize. These are the thermonuclear explosion of a carbon-oxygen white dwarf accreting mass from a companion star as it approaches its Chandrasekhar mass. The relative simplicity of this system means that these have very similar luminosities and are excellent candidates for standard candles. Futhermore, since their peak luminosity can equal or exceed the light of an entire galaxy, they can be seen to very high redshifts.

However, SNe-1a are not perfect standard candles and much work has gone into standardizing them. The figure below (taken from the Frieman review) shows different light curves and the diversity of SNe on the top panel and a standardized version on the bottom panel, using the light-curve timescale to standardize them.


B Band
as measured
Calan/Tololo SNe Ia
light-curve timescale
"stretch-factor" corrected
$M_B - 5\log(h/65)$
days
-20
-19
-18
-17
-16
-15
0
20
40
60

10.2. Distance Modulus

In astronomy, fluxes are usually measured in terms of apparent magnitudes

$$m = -2.5 \log_{10} \left(\frac{f}{f_{\rm ref}} \right)$$

while luminosities are measured in absolute magnitudes

$$M = -2.5 \log_{10} \left(\frac{L}{L_{\rm ref}} \right)$$

relative to a reference source. Taking the difference, we find

$$m - M = \mu(z) \equiv 5 \log_{10} \left(\frac{d_L}{10 \text{ pc}} \right)$$

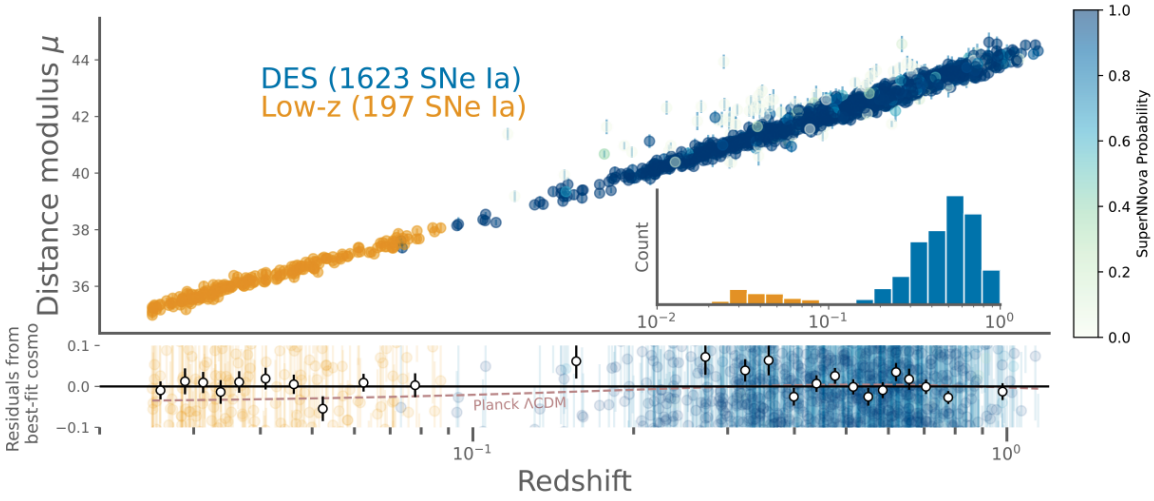
where we have defined the distance modulus and chosen our reference sources such that the distance modulus at 10pc is zero. We can rewrite this as

$$m = \left[M + 5 \log_{10} \left(\frac{c}{H_0} \frac{1}{10 \text{ pc}} \right) \right] + 5 \log_{10}(\tilde{d}_L)$$

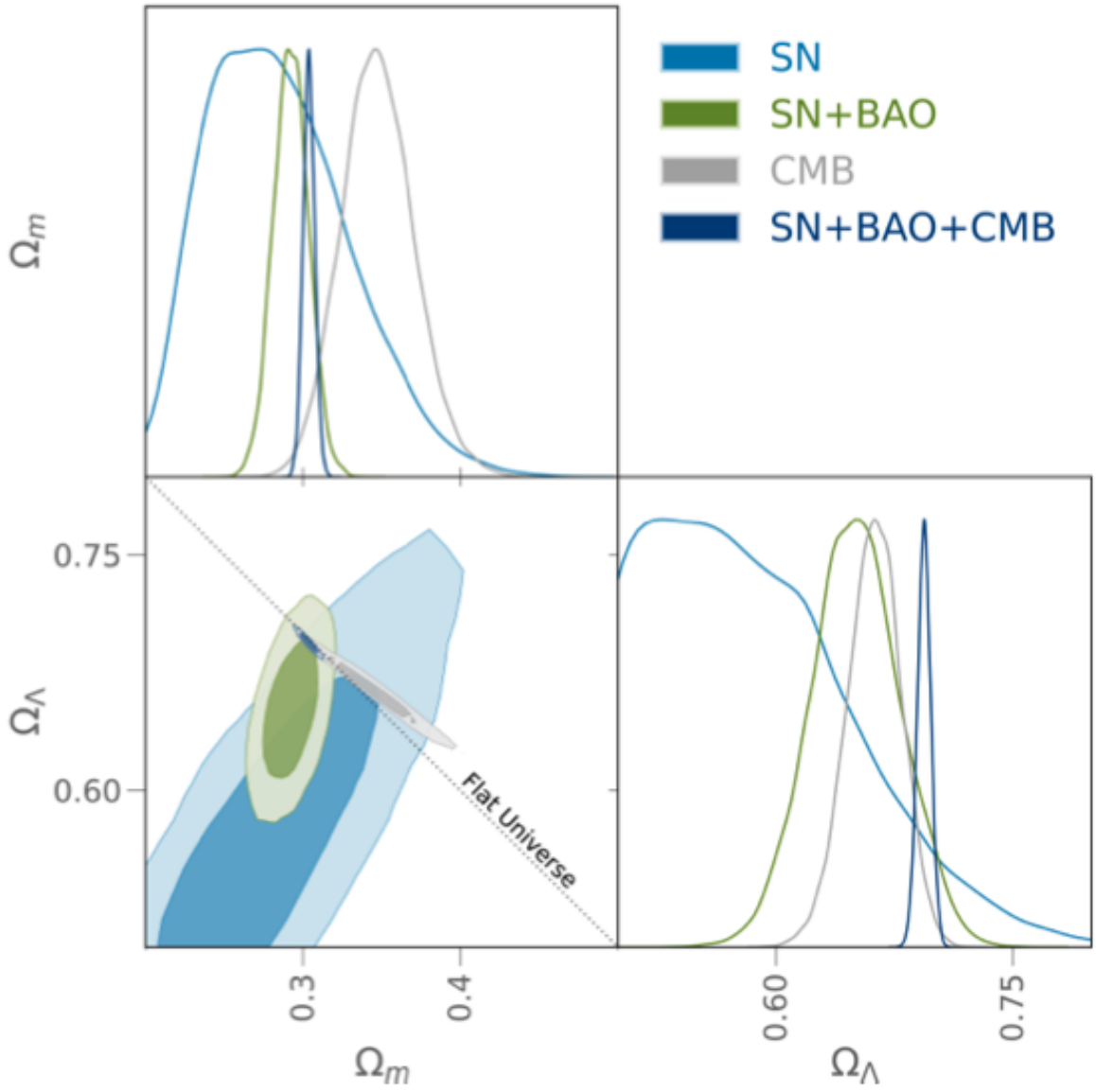
where $\tilde{d}_L$ is the luminosity distance measured in units of c/H_0. This form emphasizes the fact that SNe measure relative distances with one nuisance parameter that combines the Hubble constant and their absolute luminosity. It is only if one has a absolutely calibrated source that one can measure the Hubble constant.

10.3. Recent Results : DES 5yr

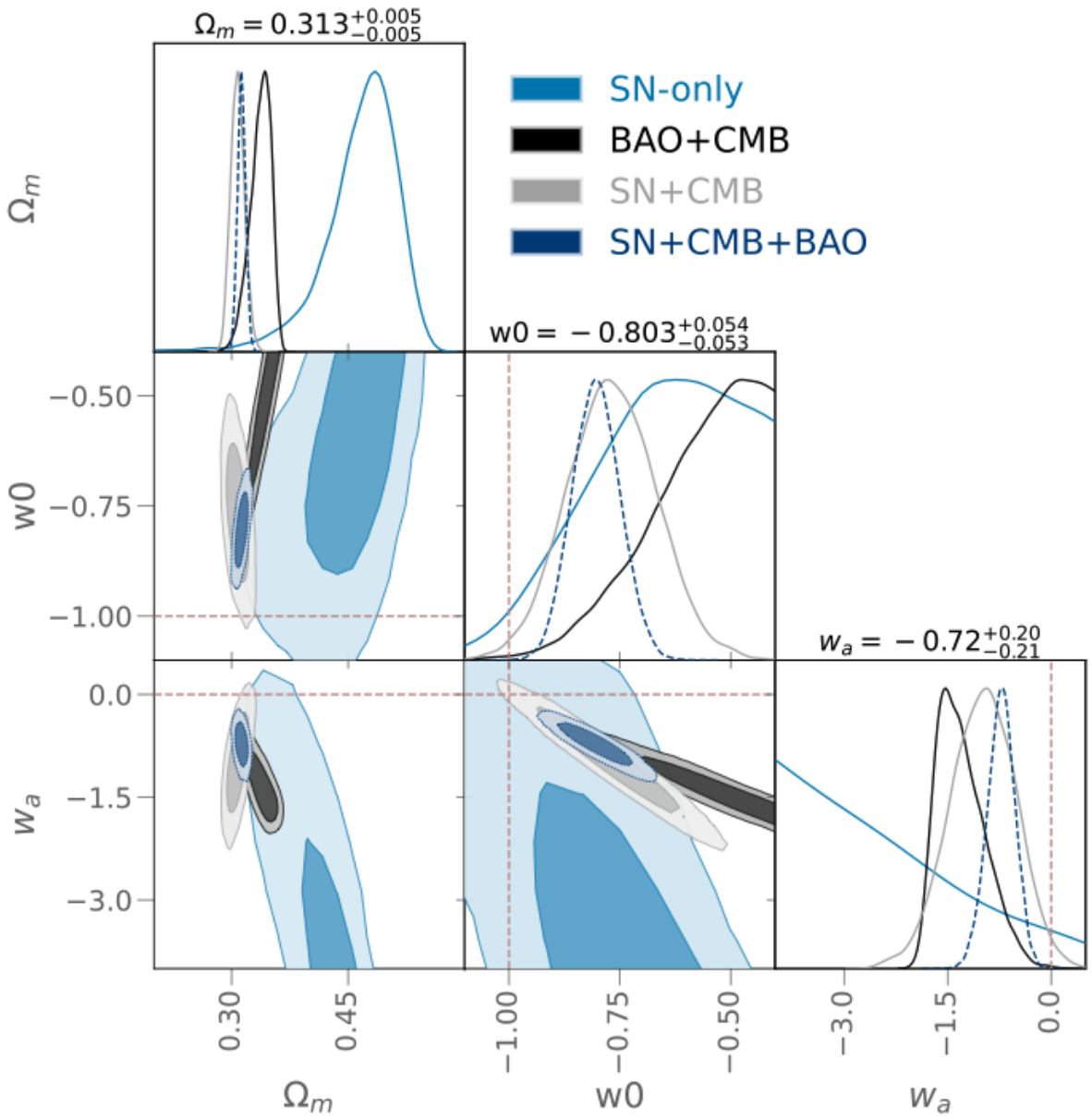
As an example, here are some recent results from the Dark Energy Survey (DES) SNe program (Popovic et al). Start with the Hubble diagram:



Constraints on a model allowing for a cosmological constant and curvature, and in combination with other data sets



And now looking for dark energy variation with redshift :



10.4. References

This follows Chap 12 in Huterer, and Frieman et al, 2008. The DES results are described here

11. A Worked Example : The Union3 Supernovae Dataset

As a worked example, let us take one of the latest supernovae datasets, the Union3 dataset and plot it and then add some simple curves showing different cosmological models. This will build off our previous examples, but we will keep this notebook fully self-contained for pedagogical purposes.

References :

- The data are from here : https://github.com/CobayaSampler/sn_data
- Complete references to all SN data sets can be found in the README of the repository.
- The Union3 data are from arXiv:2311.12098

11.1. Loading the Data

We use pandas to read the data directly from the URL. The data file is whitespace-separated with a comment header line. We only need the `zhel` (heliocentric redshift) and `mb` (distance modulus) columns.

```
Loaded 22 supernovae
Redshift range: 0.050 to 2.262
```

11.2. Loading the Covariance Matrix

The covariance matrix file contains the full covariance matrix for the distance modulus measurements. We'll extract the diagonal elements to get the variance (and hence the error) for each supernova.

```
Covariance matrix shape: (22, 22)
Typical distance modulus error: 0.095
```

11.3. Cosmological Distance Functions

We need functions to compute the luminosity distance for different cosmological models. We follow previous notebooks here.

One additional difference is the the SNe analyses plot the distance modulus

$$\mu = m - M = 5 \log_{10} \left(\frac{d_L(z)}{10\,\mathrm{pc}} \right)$$

If we choose to measure distances in units of c/H_0, then the distance modulus can be written as

$$\mu = m - M = 5 \log_{10} (d_L(z)) + \text{offset}$$

where the offset absorbs the unit conversion (approximately 43.16 for $h \sim 0.7$) as well as uncertainties in the absolute magnitude of the SNe.

11.4. Cosmological Models

We define four cosmological models to compare with the data:

- Fiducial ΛCDM: $\Omega_m = 0.3$, $\Omega_\Lambda = 0.7$ (flat)
- Open CDM: $\Omega_m = 0.3$, $\Omega_\Lambda = 0$ (open, no dark energy)
- Einstein-de Sitter: $\Omega_m = 1.0$, $\Omega_\Lambda = 0$ (flat, matter only)
- de Sitter: $\Omega_m = 0$, $\Omega_\Lambda = 1$ (pure dark energy)

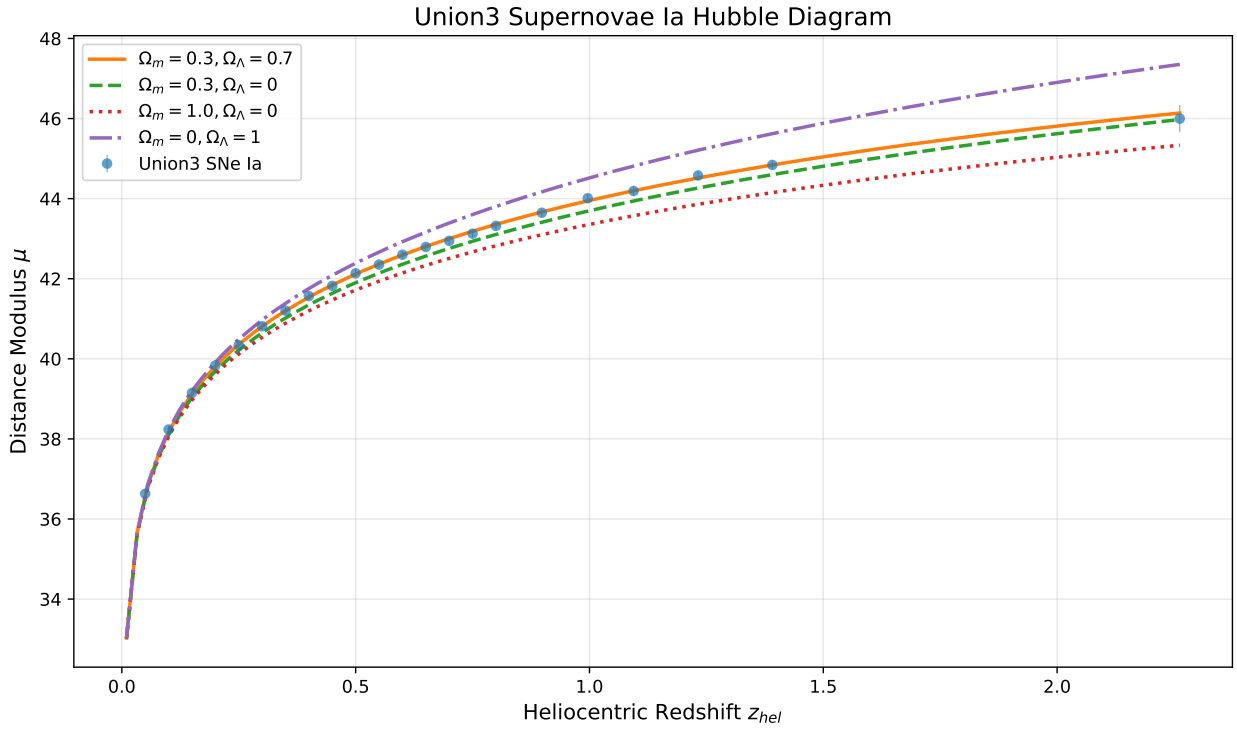
We fit the offset to the fiducial model, then use that same value for the other models to show how they deviate from the data.

This is a much simplified approach. A more complete analysis would fit for the cosmological parameters and the offset simultaneously.

```
Fitted offset = 43.012
```

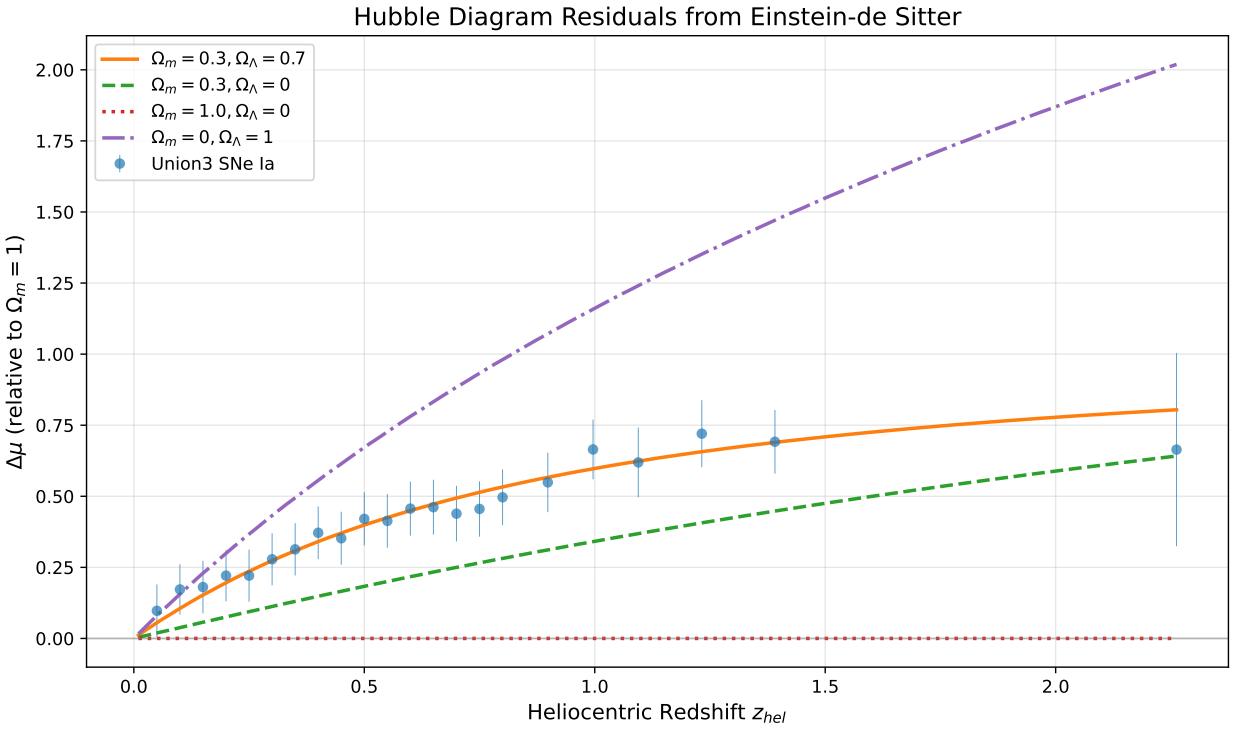
11.5. Plotting the Hubble Diagram

The classic "Hubble diagram" plots apparent magnitude versus redshift, while the SNe versions tend to plot the inferred distance modulus versus redshift.



11.6. Residuals from Einstein-de Sitter

To better see the differences between models, we plot the residuals relative to the Einstein-de Sitter ($\Omega_m = 1$) model. This is a common way to visualize the evidence for dark energy.


Hubble Diagram Residuals from Einstein-de Sitter
$\Omega_m = 0.3, \Omega_\Lambda = 0.7$
$\Omega_m = 0.3, \Omega_\Lambda = 0$
$\Omega_m = 1.0, \Omega_\Lambda = 0$
$\Omega_m = 0, \Omega_\Lambda = 1$
Union3 SNe Ia
$\Delta\mu$ (relative to $\Omega_m = 1$)
Heliocentric Redshift z_{hel}
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
0.0
0.5
1.0
1.5
2.0

12. Fisher Matrices in Cosmology

The discussion here parallels Chap. 10 in the Huterer textbook. I will take a Bayesian approach here, since that is most prevalent in cosmology.

12.1. Terminology

In cosmology, we are interested in inferring cosmological parameters given observational data. More formally, we are interested in the **posterior** probability distribution $P(\text{params}|\text{data})$. By Bayes theorem, we can write this as

$$P(\text{params}|\text{data}) \propto P(\text{data}|\text{params})P(\text{params})$$

where the first factor on the right is referred to as the **likelihood** while the second is the **prior**. The prior is meant to quantify our prior beliefs about the value of the data.

Let us write the data as a vector $\mathbf{d}$ and the model similarly as $(\{\theta\})$, where the parameters are $\{\theta\}$. A very useful form of the likelihood $\mathcal{L}$ is the Gaussian form

$$\mathcal{L}(\{\theta\}) = \frac{1}{(2\pi)^{N/2}|\mathbf{C}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{d}-\mu(\{\theta\}))^\top \mathbf{C}^{-1}(\mathbf{d}-\mu(\{\theta\}))\right)$$

where $\mathbf{C}$ is the covariance matrix.

12.2. Fisher Matrices

Suppose you were designing an experiment and wanted to understand how the errors on your measurements translated into cosmological parameters. One approach would be to compute the Fisher matrix

$$F_{ij} = \left\langle -\frac{\partial^2 \ln \mathcal{L}}{\partial\theta_i \partial\theta_j} \right\rangle$$

The inverse of the Fisher matrix then gives you the best possible errors you can find (the Cramer-Rao bound)

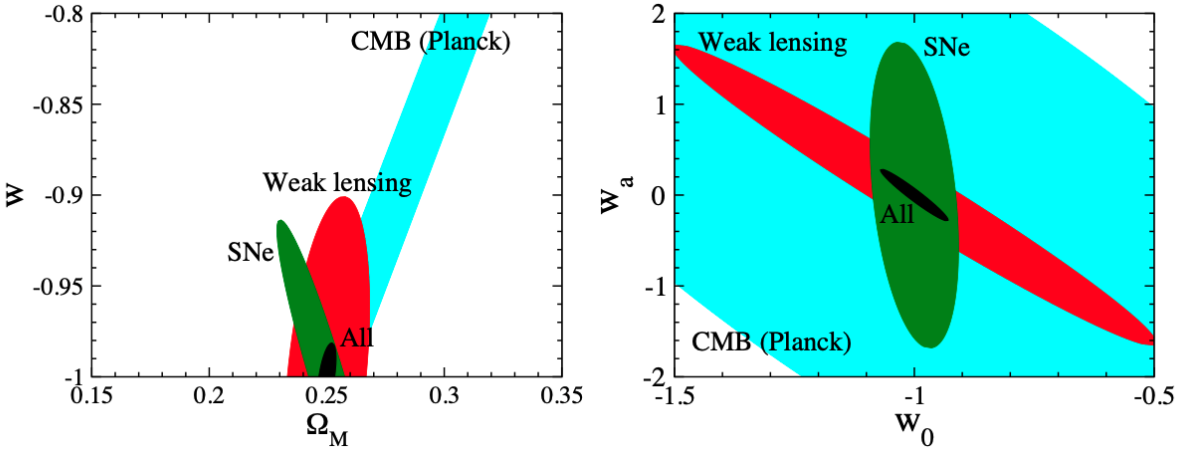
$$\mathbf{C}_\theta = \mathbf{F}^{-1}$$

For our Gaussian likelihood, you can show that

$$F_{ij} = \mu_{,i}^\top \mathbf{C}^{-1}\mu_{,j} + \frac{1}{2}\text{Tr}[\mathbf{C}^{-1}\mathbf{C}_{,i}\mathbf{C}^{-1}\mathbf{C}_{,j}]$$

where we allow for the fact that our covariance depends on the parameters. If not, then we only have the first term.

Fisher forecasts are ubiquitous in cosmology. Below are two examples from the Frieman et al, 2008 review of dark energy.



13. Fisher Error Estimation : A Simple Worked Example

As a simple example of computing a Fisher matrix and estimating parameter errors in a cosmological context, we consider the following toy problem. Imagine that we measure the comoving distance χ (in units of c/H_0) at a single redshift with a known measurement error σ_χ. We'll assume a simple cosmological model with only Ω_m, and we'd like to estimate the Fisher matrix and the expected error on Ω_m from this single measurement.

In this case, the derivatives are analytically tractable, so you could compute the Fisher matrix with standard numerical integration. However, as a demonstration, we will continue to explore differentiable programming and compute the Fisher matrix using automatic differentiation. See previous notebooks for examples of using JAX for automatic differentiation in cosmology.

Let's just write down the Fisher matrix for this simple case.

The general form of the Fisher matrix for a fixed covariance matrix C and a model μ is

$$F_{ij} = \frac{\partial \mu^T}{\partial \theta_i} C^{-1} \frac{\partial \mu}{\partial \theta_j}$$

where θ_i are the model parameters (in our case, just Ω_m), μ is the model prediction for the observable (here, the comoving distance χ), and C is the covariance matrix of the measurement (here, just σ_χ^2).

In our case, this simplifies to a single parameter and a single measurement, so the Fisher matrix is just a scalar:

$$F = \frac{1}{\sigma_\chi^2} \left(\frac{\partial \chi}{\partial \Omega_m} \right)^2$$

and the expected error on Ω_m is simply

$$\sigma_{\Omega_m} = \frac{1}{\sqrt{F}}.$$

13.1. Cosmological Distance Functions

We use JAX and Diffrax to compute the comoving distance, following the approach in the Friedmann-Numerical-JAX notebook. The key advantage is that JAX gives us automatic differentiation for free.

13.2. Computing the Fisher Error

Now we use JAX's automatic differentiation to compute $\partial\chi/\partial\Omega_m$. The Fisher error on Ω_m is then:

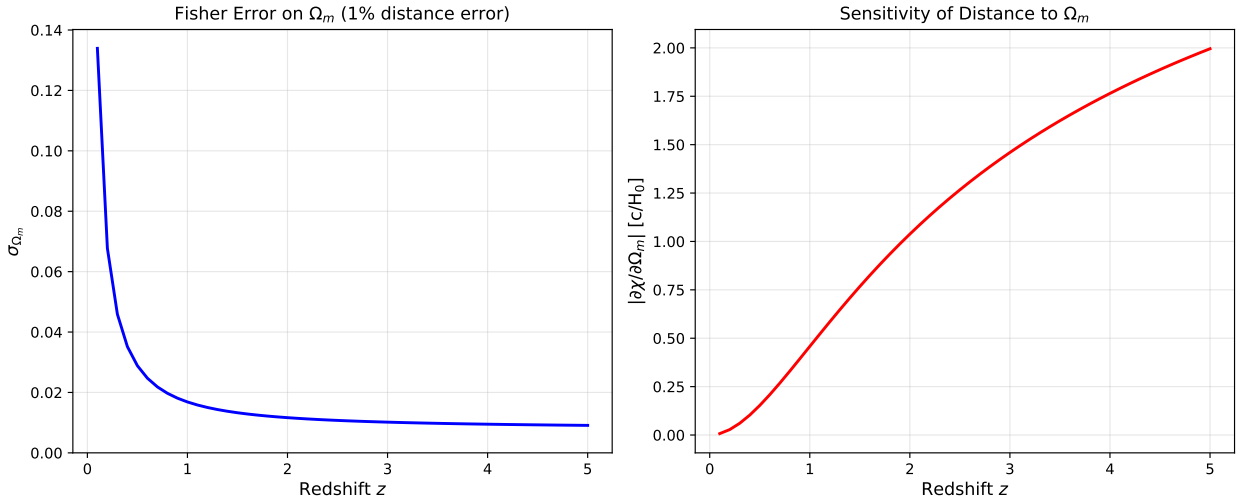
$$\sigma_{\Omega_m} = \frac{\sigma_\chi}{|\partial\chi/\partial\Omega_m|}$$

We'll assume a fixed fractional error on the distance measurement, say 1%.

```
Fisher error on Ωm from 1% distance measurement:
================================================
       z       [c/H0]        / Ωm         (Ωm)
------------------------------------------------
     0.5       0.4410      -0.1529       0.0288
     1.0       0.7714      -0.4577       0.0169
     2.0       1.2095      -1.0380       0.0117
     3.0       1.4840      -1.4596       0.0102
```

13.3. Fisher Error vs Redshift

Let's plot how the constraining power on Ω_m changes with the redshift of the measurement. Higher redshift measurements probe larger distances where the sensitivity to Ω_m is greater.



13.4. Interpretation

The plots show that:

1. **Higher redshift = better constraints**: The Fisher error decreases with redshift, meaning high-z distance measurements constrain Ω_m better than low-z ones (for fixed fractional error).

2. **Why?** The sensitivity $|\partial\chi/\partial\Omega_m|$ increases with redshift. At higher redshifts, the comoving distance has accumulated more "history" of the expansion, making it more sensitive to the matter density. At low redshift, $\chi \sim z + ...$ and so, there is much less sensitivity to Ω_m at low redshift.
3. **Practical implications**: This is why cosmological surveys target high-redshift objects (Type Ia supernovae at $z \sim 1$, BAO at $z \sim 0.5 - 2$, CMB at $z \sim 1100$).

Of course, this is a simplified example. Real surveys must account for:

- Multiple parameters (not just Ω_m)
- Correlations between measurements
- Systematic errors

14. The Horizon Problem

We've already encountered two interesting puzzles with our standard Big Bang cosmology. The first, although not emphasized, was that the age of the Universe - given only matter and radiation and a Hubble constant close to its value today - is inconsistent with observations.

The second was the flatness problem - Ω_K being so close to 0 today implies that it must have been vanishingly small in the past. The puzzle is then to dynamically explain where such a small value could come from.

The third is the horizon problem. This is best discussed with a picture of the CMB

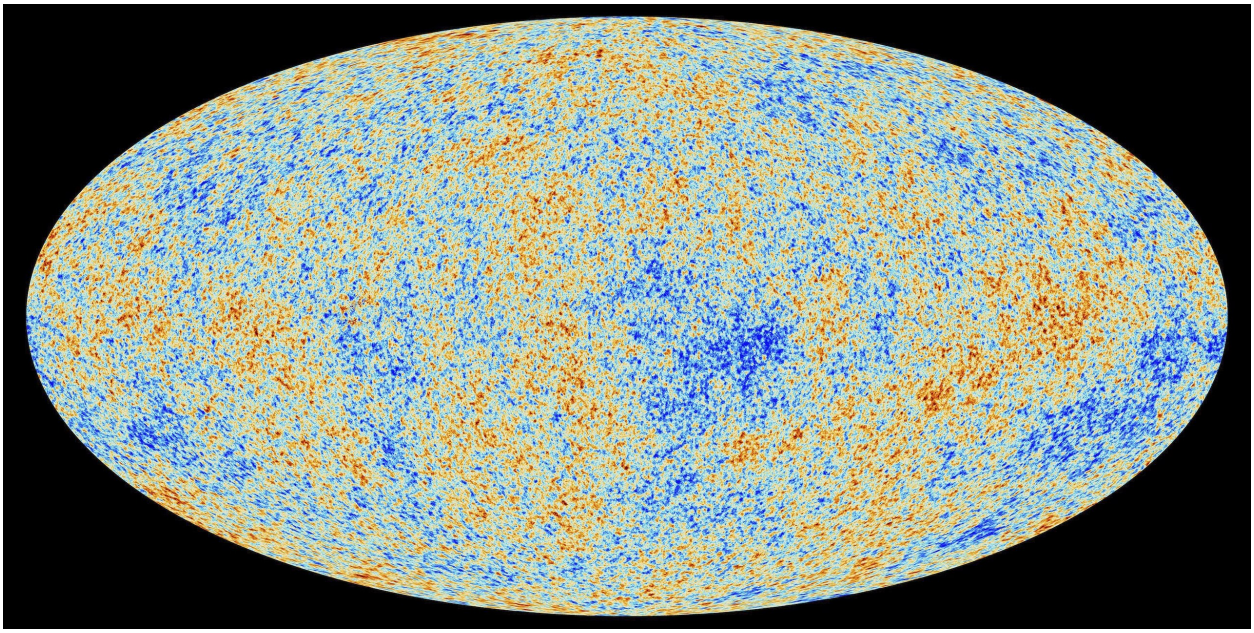


Figure 14.1.: By ESA and the Planck Collaboration

The fluctuations in the above map are a part in 10^{-5}; the temperature itself is a uniform 2.73K across the entire sky. However, as we'll see, different parts of this map have not been in causal contact at the time the CMB was created. So the puzzle is how does one dynamically get such a uniform map across the entire sky.

There is a second (and almost even worse) problem. It's not just that the map is uniform, but it's also that the fluctuations in the map are actually correlated with one another, best seen in the CMB angular power spectrum shown below.

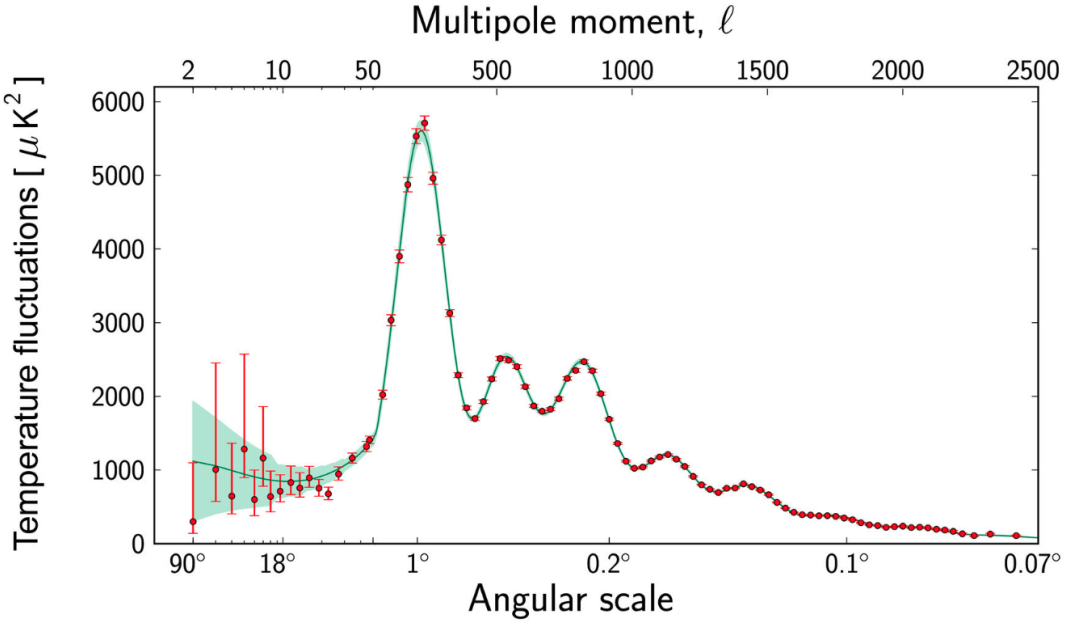


Figure 14.2.: CMB Power Spectrum, ESA & Planck

As we'll see the solution to all of these is a period of accelerated expansion - or equivalently a period of a decreasing Hubble radius - in time. For the age problem, this is provided by dark energy, while for the horizon and flatness problem, this is solved by inflation.

14.1. Horizons

Before we proceed, it is useful to define different concepts of "horizons" that show up in cosmology. In what follows, it is more elegant and convenient to work in conformal time coordinates

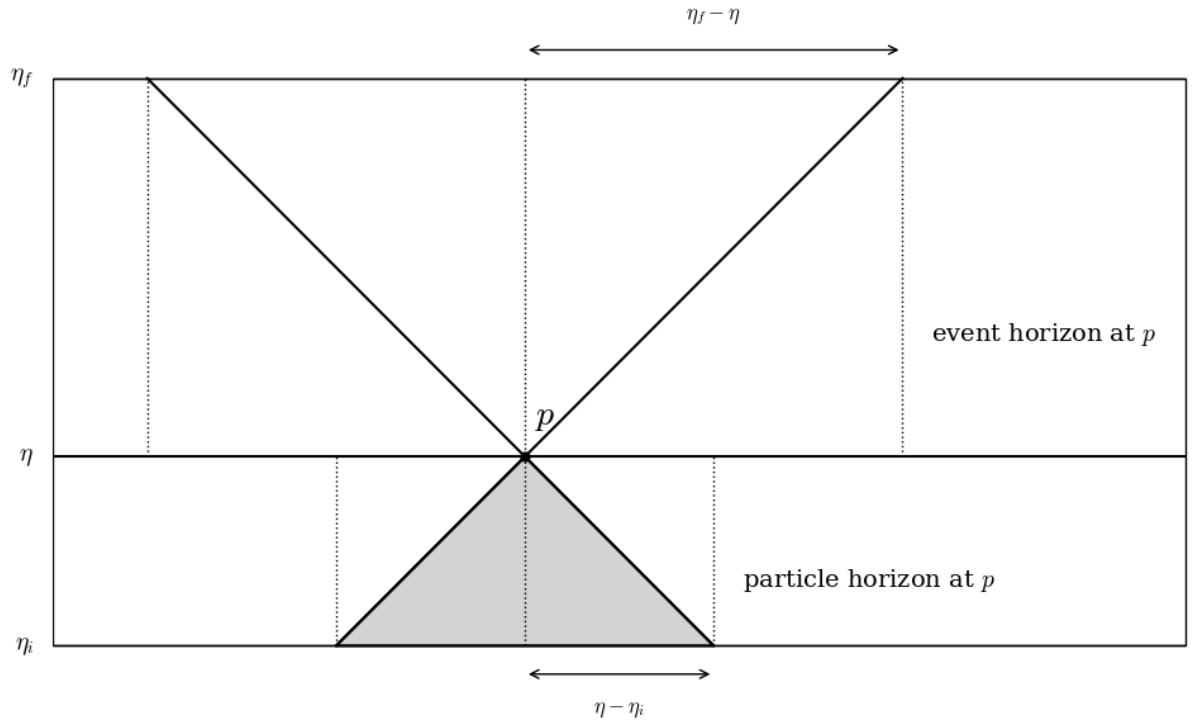
$$ds^2 = a(t)^2 \left[-d\eta^2 + d\chi^2\right]$$

The particle horizon is defined as the region could have causally affected a point. From the metric, we see that the particle horizon is defined by

$$\chi_p = \eta - \eta_i = \int_{t_i}^{t} \frac{1}{a(t)}\, dt$$

where η_i is our initial time. The particle horizon is what is most relevant for our discussion below.

The second horizon is the event horizon, which is the maximal radius out to which we can be in causal contact in the future. The figure below sketches out the ideas of a particle and event horizon.



Finally, we have the comoving Hubble radius $(aH)^{-1}$, which is how far a particle could travel "now" (i.e. before the Universe appreciably expands). As we'll see below, for matter and radiation, $\chi_p \sim (aH)^{-1}$ and the two are often just conflated.

14.2. The Horizon Problem, restated

The horizon problem can now be easily restated with the following figure (Fig. 4.2 in Baumann)

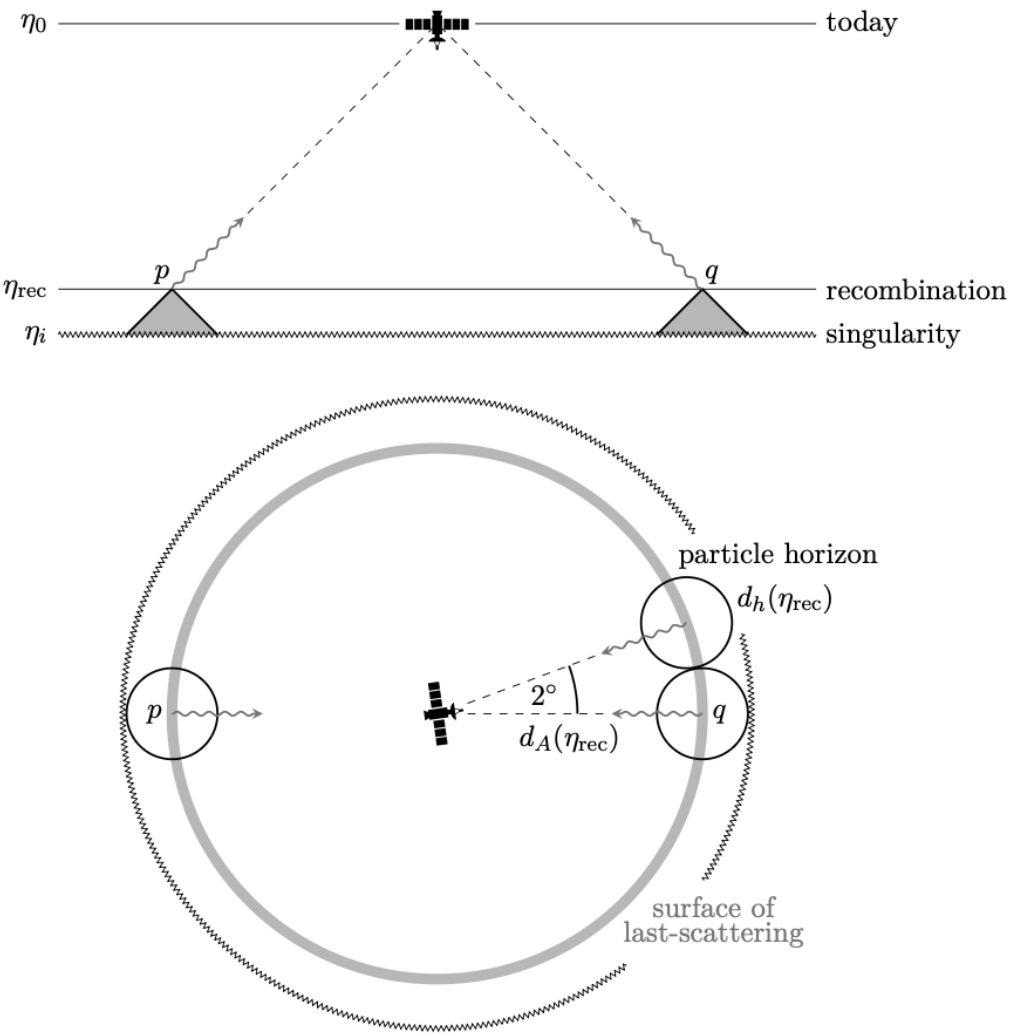


Figure 14.3.: Baumann, Fig. 4.2

We can see the issue in the following. The particle horizon can be written as

$$\chi_p = \int_{t_i}^{t} \frac{1}{a(t)}\, dt = \int_{a_i}^{a} \frac{1}{a\dot{a}}\, da = \int_{\ln a_i}^{\ln a} \frac{1}{aH(a)}\, d\ln a$$

relating the particle horizon to evolution of the comoving Hubble radius. For normal matter, the comoving Hubble radius is monotonically increasing and so, this integral is dominated by late times, giving rise to the picture above.

We can be more explicit by considering a constant equation of state w. In this case, we have

$$H(a) = H_0 a^{-3(1+w)/2}$$

or

$$(aH)^{-1} = H_0^{-1} a^{(1+3w)/2}$$

which is clearly monotonically increasing if $w > -\frac{1}{3}$ (which was our condition for a decelerating Universe). The particle horizon is then

$$\chi_p(a) = \eta - \eta_i = \frac{2H_0^{-1}}{1+3w}\left[a^{(1+3w)/2} - a_i^{(1+3w)/2}\right]$$

Note that (for $w > -\frac{1}{3}$)

$$a^{(1+3w)/2} \to 0 \text{ as } a \to 0$$

again emphasizing that the particle horizon is dominated by late time contributions.

We can estimate these for $w = 0$. The particle horizon at the CMB $a \sim 10^{-3}$ is

$$\chi_{CMB} = 2 \times 3000 \times \sqrt{0.001} \approx 200 \text{ Mpc/h} \approx 286 \text{ Mpc}$$

while the distance to the CMB is

$$r_{CMB} = (6000 - 200) \approx 6000 \text{ Mpc/h} \approx 8600 \text{ Mpc}$$

The angle subtended by the particle horizon on the CMB is therefore $\sim 2^\circ$.

A similar story can be told for the superhorizon correlations seen in the figure below (Fig. 4.3 in Baumann) where a fluctuation of a scale λ is compared with the comoving Hubble radius (which as we see above is similar to the particle horizon for standard matter). However, we see scales in the CMB that are correlated that would have been greater than the particle horizon/Hubble radius at the time of the CMB.

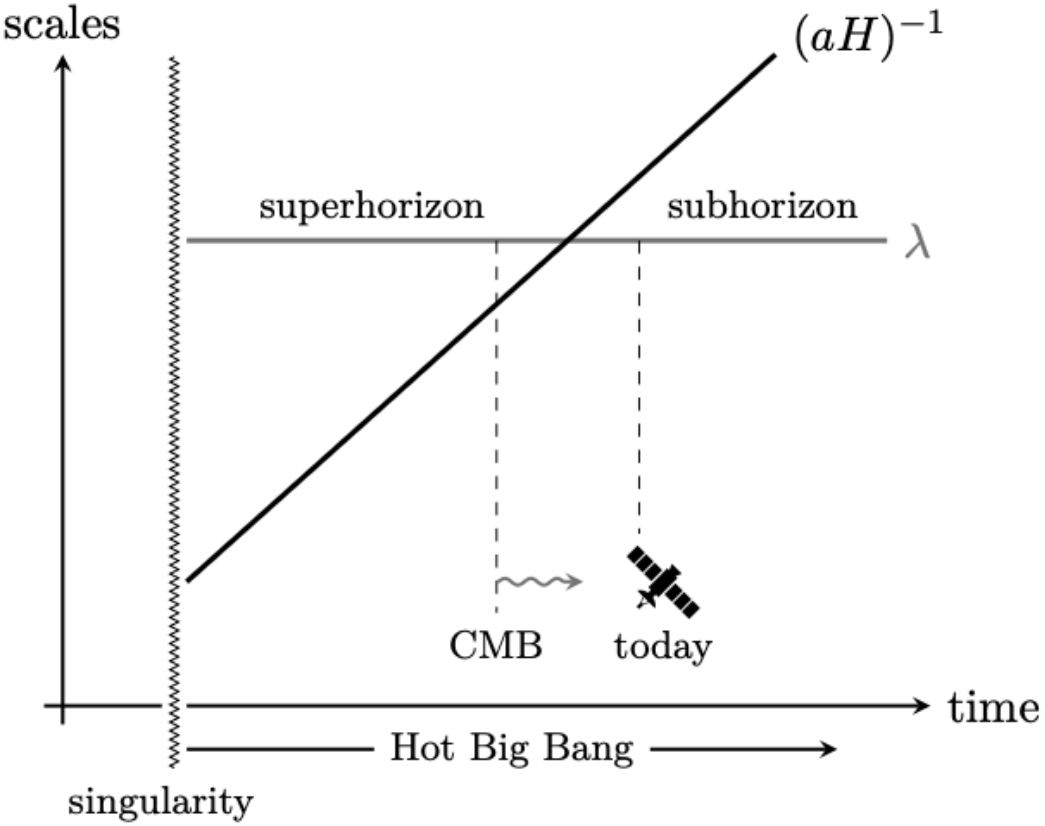


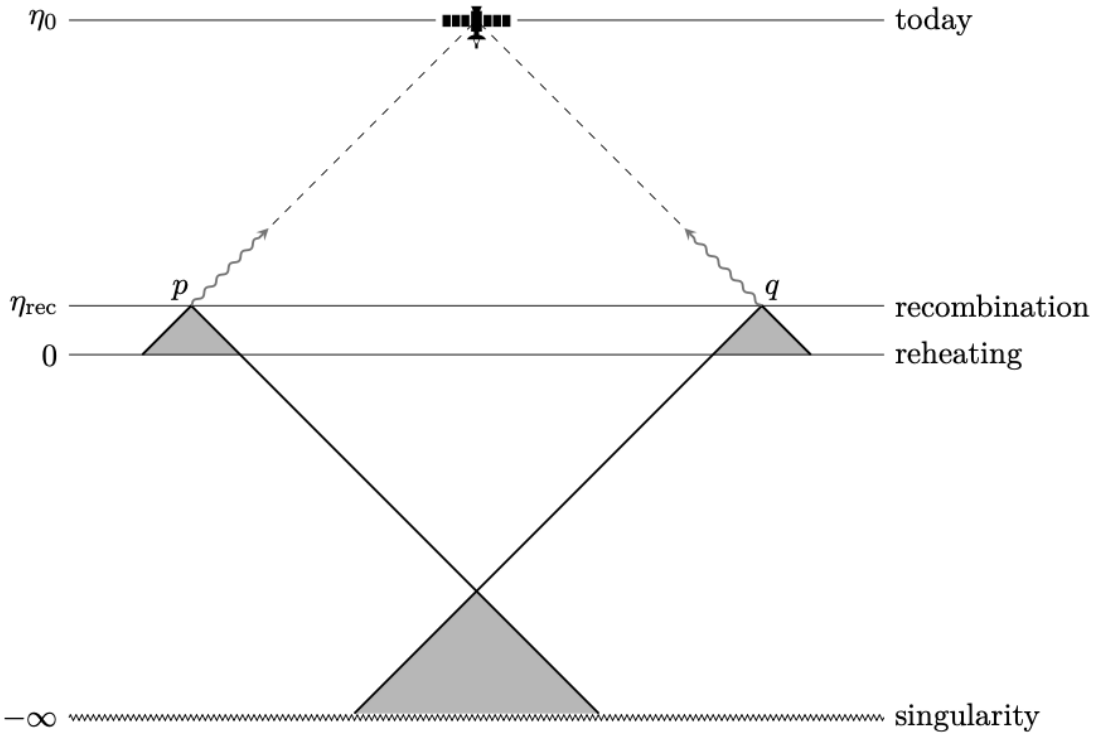
Figure 14.4.: Baumann, Fig. 4.3

14.3. The Inflationary Solution

One solution to this problem is to introduce a period where the Hubble radius is shrinking with time. We can do this in our simple model with a period (referred to as inflation) dominated by a component with $1 + 3w < 0$ or $w < -1/3$. In this case, we have

$$\eta_i \to -\infty \text{ as } a_i \to 0$$

giving us plenty of time to bring regions into causal contact. This is shown in the figure below (Fig 4.4 in Baumann). Note that in this picture, the hot Big Bang is not a singularity, but rather just a point in time in the Universe when this inflation ends.



Note that a shrinking Hubble sphere is equivalent to an accelerated expansion of the Universe

$$aH = \dot{a}$$
$$\frac{d}{dt}(aH)^{-1} < 0 => \frac{d}{dt}\frac{1}{\dot{a}} < 0$$
$$\ddot{a} > 0$$

It is also sometimes referred to as a period of exponential expansion, which is what one gets if $w \sim -1$ (just as with dark energy).

The picture for superhorizon correlations also gets modified to the figure below (Fig. 4.7 in Baumann), where we see a mode start inside the horizon, exit and then re-enter the horizon.

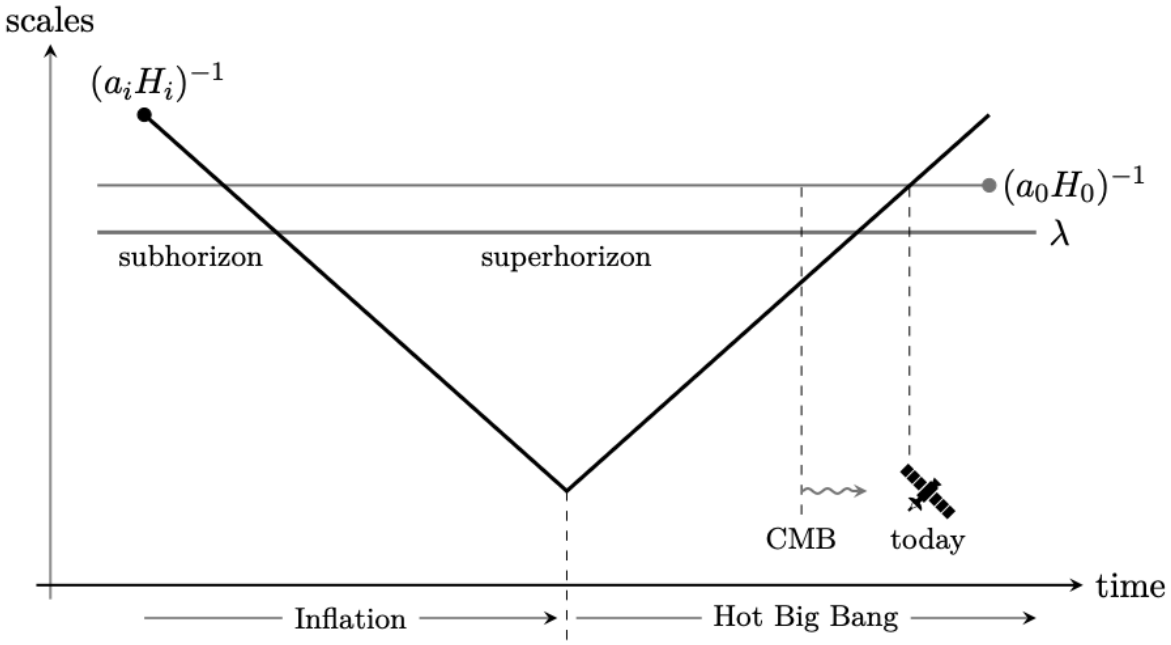


Figure 14.5.: Baumann, Fig. 4.7

14.3.1. How much inflation?

Let us now estimate how long this period of inflation would need to be. A simple condition for inflation to work is that

$$(a_0H_0)^{-1} < (a_iH_i)$$

i.e. that the comoving Hubble radius today is less than the comoving Hubble radius at the start of inflation. As before, today will be denoted by 0, while the start of inflation will be i. We will also denote the end of inflation (which is the "hot Big Bang", or the reheating epoch) as e.

For simplicity, we will just consider $w = -1$ to describe the inflationary epoch. This implies $H_e = H_i$. Furthermore, let us just imagine the Universe post reheating/inflation was just radiation dominated and so $H \propto a^{-2}$. This gives us

$$\frac{a_0H_0}{a_eH_e} = \frac{a_0}{a_e}\left(\frac{a_e}{a_0}\right)^2 = \frac{a_e}{a_0} \sim \frac{T_0}{T_e} \sim 10^{-28}\left(\frac{10^{15}\text{ GeV}}{T_e}\right)$$

Now define the number of e-folds of inflation

$$N = \ln\left(\frac{a_e}{a_i}\right)$$

our condition becomes

$$\frac{a_iH_i}{a_0H_0} > 1$$
$$e^{-N}\frac{a_eH_e}{a_0H_0} > 1$$
$$N > 64 + \ln\left(\frac{T_e}{10^{15}\text{ GeV}}\right)$$

14.3.2. Flatness Revisited

We can now see whether this same solution addresses the flatness problem. Suppose at the start of inflation, the Universe had some non-zero $\Omega_{K,i}$. At the end of inflation, we have

$$\Omega_{K,e} \sim \Omega_{K,i} \left(\frac{a_i}{a_e} \right)^{-2}$$

since H is approximately constant. From our previous estimate, this gives

$$\Omega_{K,e} \sim \Omega_{K,i} e^{-2N}$$

Inflation drives down the curvature to a negligible value.

Part II.

Linear Perturbations

15. Introduction to Linear Perturbations

We now turn to inhomogeneities in the Universe and how these evolve with time. Our treatment here will be limited to the Newtonian regime - so considering velocities $\ll c$ and length scales much smaller than the horizon. It turns out that these scales are important when considering much of the scales of current galaxy surveys and so, making this approximation is both justified and relevant.

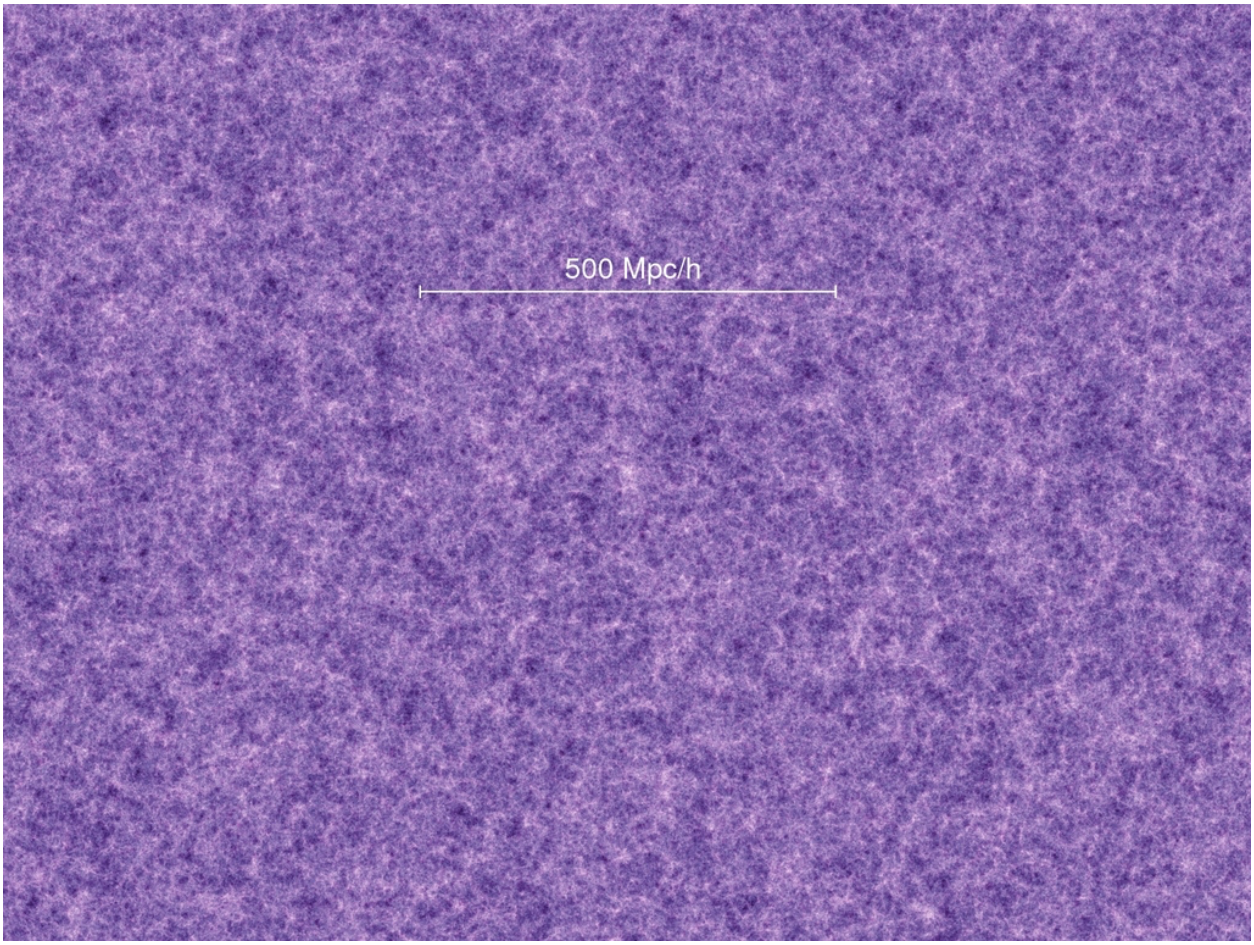
We will consider both baryons and dark matter here, and there will be places where we will distinguish between the two.

The treatment here follows Chap. 5 in Baumann closely.

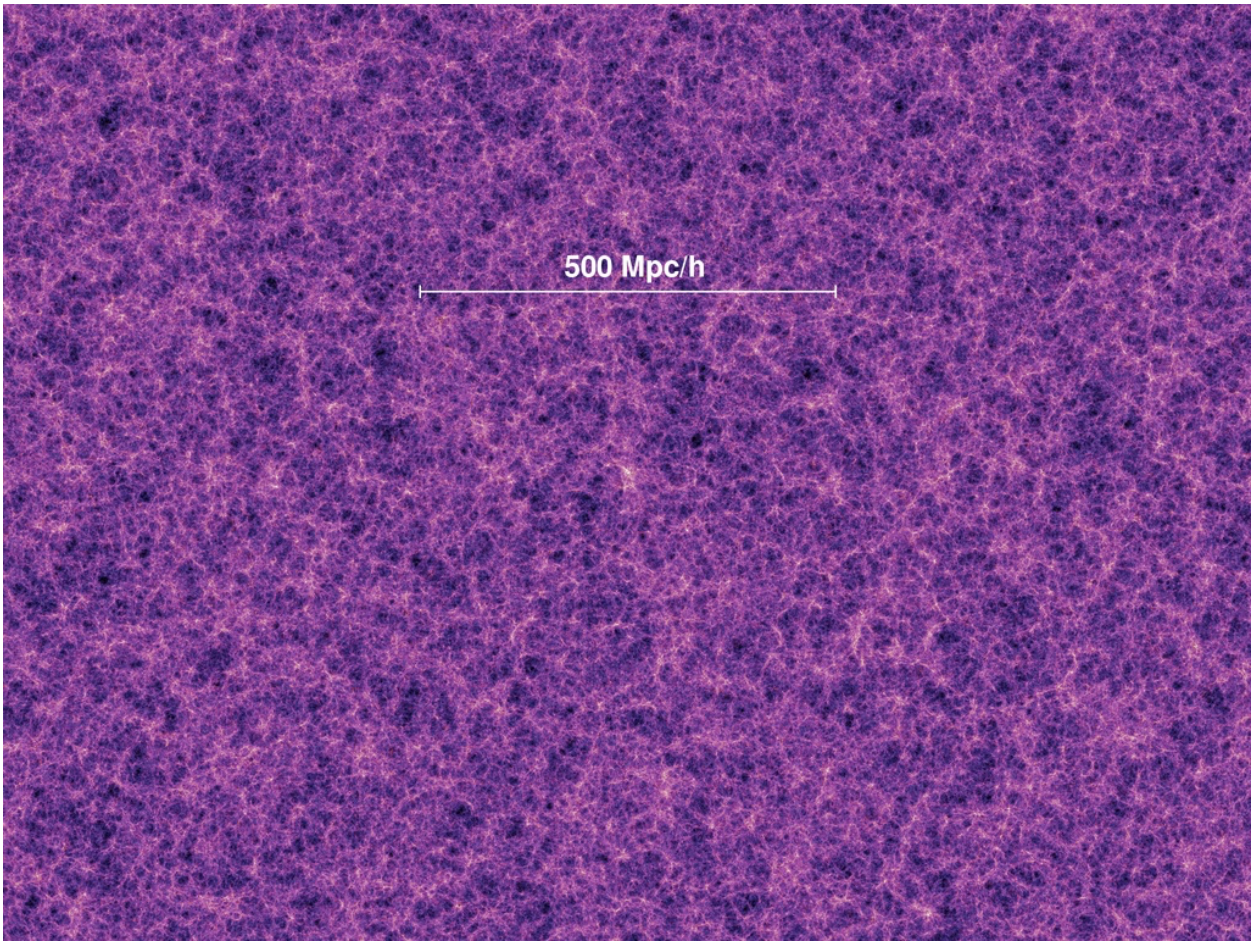
15.1. Motivation

To motivate our discussion, consider the following sequence of snapshots of the dark matter distribution from the Millennium simulation, showing the evolution of the cosmic web as a function of time. As we see, structures grow with time from small perturbations in the early Universe through gravitation and form the cosmic web that we see today.

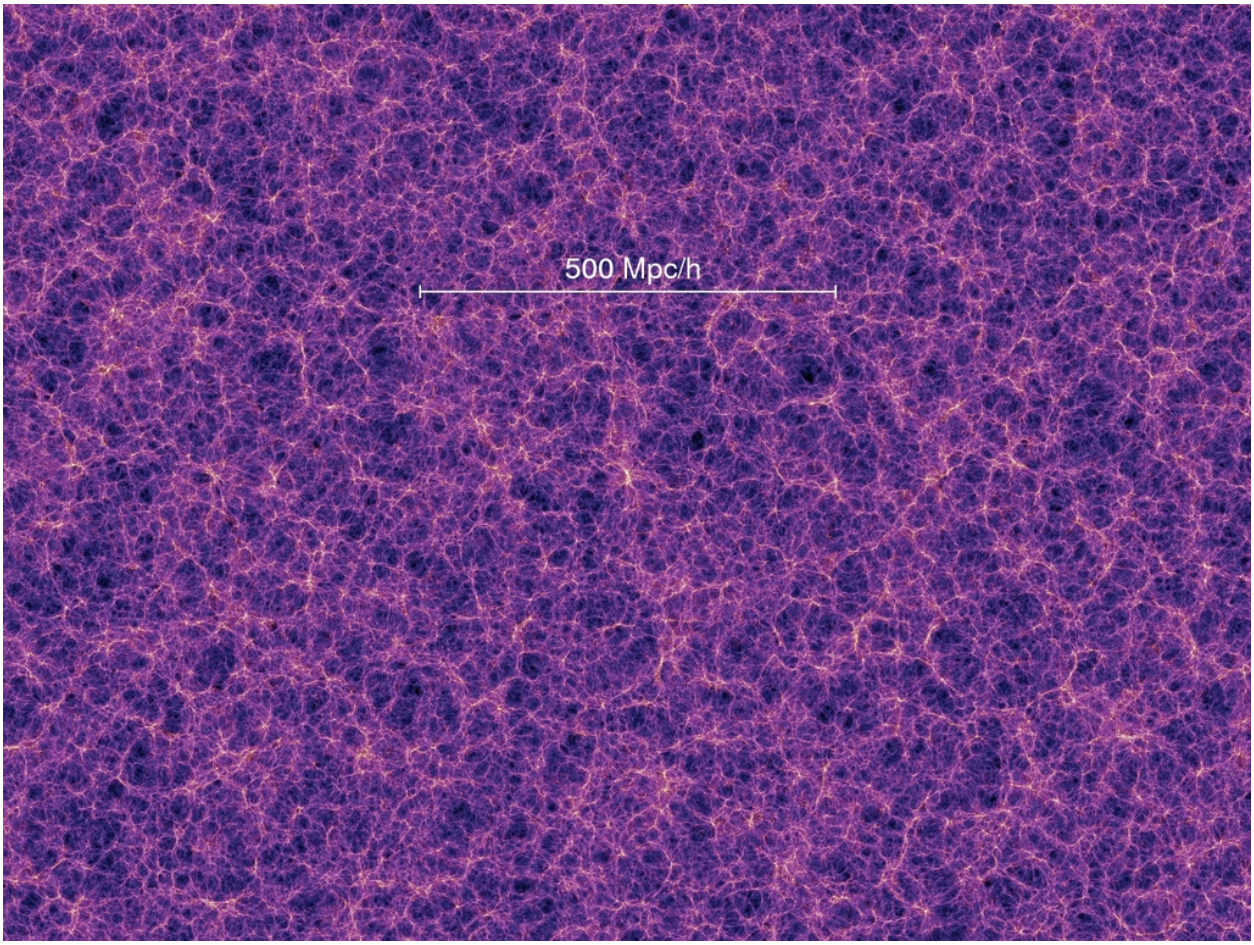
z=18.3



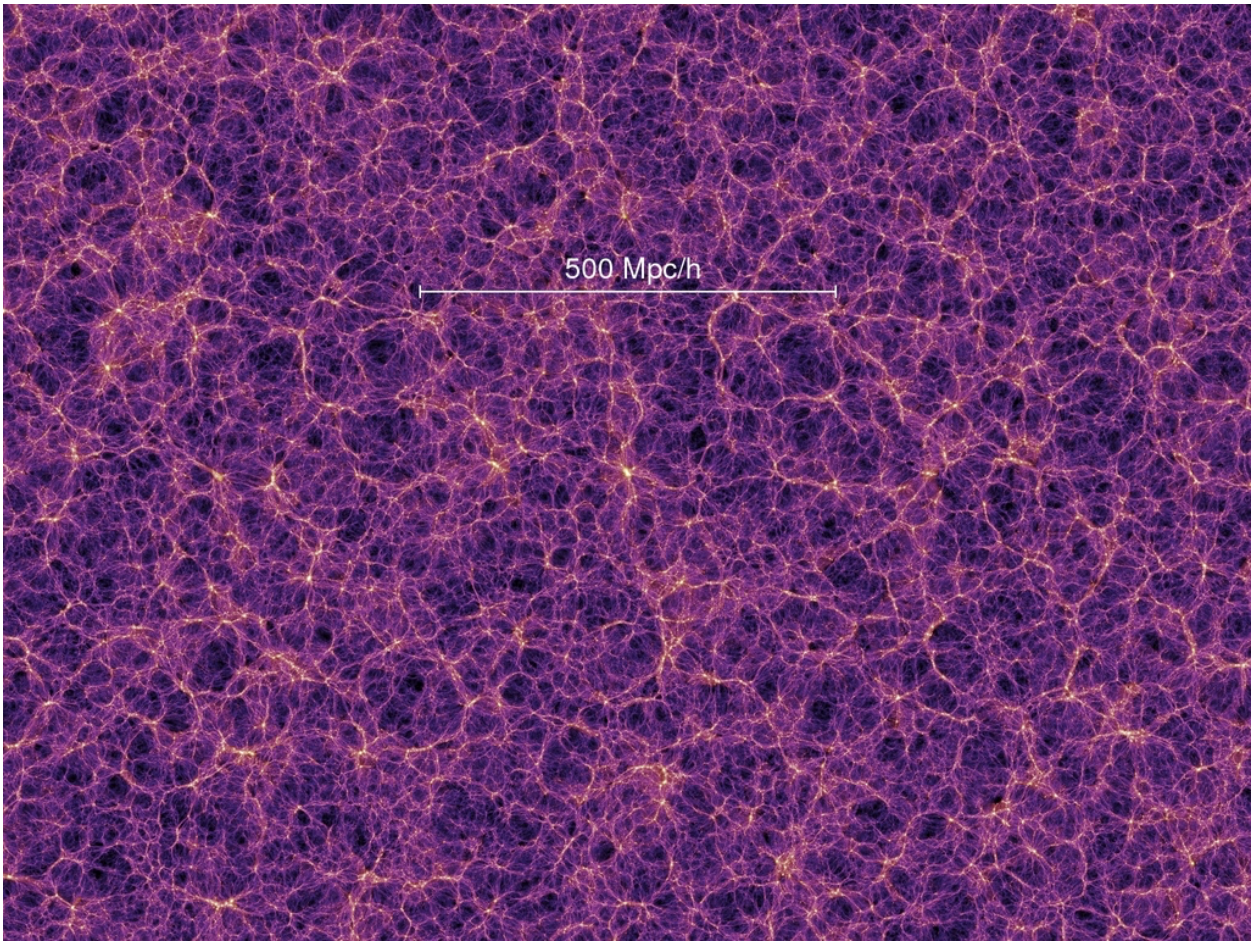
z=5.7



z=1.4



z=0 (today!)



As we go forward in time, we see that the fluctuations grow from small perturbations on all scales to large, nonlinear fluctuations. However, even today, if we average over a large enough volume, we will find that the perturbations can be made small (this is a consequence of our assumption of the homogeneity of the Universe on the largest scales), and linear perturbation theory is a good description. This will be our immediate focus.

Just to summarize the regime of linear perturbation theory : early times and/or large scales.

15.2. Fluid Dynamics

We will treat the matter as a fluid with a mass density ρ, a pressure $P \ll \rho$ and a velocity $\mathbf{u} \ll c$. We'll start by ignoring gravity and the expansion of the Universe, adding these in turn.

The two primary equations governing this fluid are the continuity equation and the Euler equation. The continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

captures mass conservation - the change in mass in a given volume per unit time is determined by the flow of material in/out of the volume. The Euler equation follows from momentum conservation

and is written as

$$\rho \frac{D\mathbf{u}}{Dt} = \rho \left(\frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla \right) \mathbf{u} = -\nabla P$$

Note the appearance of the material or convective derivative D/Dt to track the a fluid element as it flows. Finally, we close this system with an equation of state to relate pressure to density - we assume a simple form

$$P = P(\rho)$$

An initial solution to this is a static fluid with a uniform density $\bar{\rho}$, a uniform pressure $\bar{P}$ and zero fluid velocity $\mathbf{u} = 0$. Let us consider perturbations around this solution

$$\rho = \bar{\rho} + \delta\rho$$
$$P = \bar{P} + \delta P$$

where the perturbations are assumed to functions of space and time. Plugging first into the continuity equation, we find

$$\frac{\partial \delta\rho}{\partial t} = -\bar{\rho} \nabla \cdot \mathbf{u}$$

where we are working to linear order and so ignore products of perturbations (note that $\mathbf{u}$ is a perturbation about the static solution). Similarly, the Euler equation simplifies to

$$\bar{\rho} \frac{\partial \mathbf{u}}{\partial t} = -\nabla \delta P$$

We can combine these two equations to get

$$\frac{\partial^2 \delta\rho}{\partial t^2} - \nabla^2 \delta P = 0$$

Finally, we can use the equation of state to eliminate δP:

$$\delta P = \frac{\partial P}{\partial \rho} \delta\rho = c_s^2 \delta\rho$$

where c_s is the speed of sound in the fluid. This gives us

$$\frac{\partial^2 \delta\rho}{\partial t^2} - c_s^2 \nabla^2 \delta\rho = 0$$

This is nothing but the wave equation. We can solve this using Fourier transforms :

$$\delta\rho(\mathbf{x}, t) = \int \frac{d^3k}{(2\pi)^3} e^{i\mathbf{k}\cdot\mathbf{x}} \delta\rho(\mathbf{k}, t)$$

where we explicitly re-introduce the space and time dependencies to show the Fourier transform. The Laplacian ∇^2 operator pulls down two factors of $i\mathbf{k}$ and we find

$$\int \frac{d^3k}{(2\pi)^3} \left[\frac{\partial^2 \delta\rho(\mathbf{k}, t)}{\partial t^2} + c_s^2 k^2 \rho(\mathbf{k}, t) = 0 \right]$$

This is nothing by a simple harmonic oscillator for each individual $\mathbf{k}$ mode, and so the solutions are

$$\delta\rho(\mathbf{k}, t) = A_{\mathbf{k}} e^{-i\omega_k t} + B_{\mathbf{k}} e^{i\omega_k t}$$

where $\omega_k = c_s k$. Putting this back, we see that

$$\delta\rho(\mathbf{x}, t) = \int \frac{d^3k}{(2\pi)^3} \left[A_\mathbf{k} e^{-i(\omega_k t - \mathbf{k}\cdot\mathbf{x})} + B_\mathbf{k} e^{i(\omega_k t + \mathbf{k}\cdot\mathbf{x})}\right]$$

It is convenient to rewrite this as

$$\delta\rho(\mathbf{x}, t) = \int \frac{d^3k}{(2\pi)^3} \left[A_\mathbf{k} e^{-i(\omega_k t - \mathbf{k}\cdot\mathbf{x})} + B_{-\mathbf{k}} e^{i(\omega_k t - \mathbf{k}\cdot\mathbf{x})}\right]$$

Furthermore, if we require $\delta\rho$ to be real, we require $B_{-\mathbf{k}} = A^*_\mathbf{k}$ to get

$$\delta\rho(\mathbf{x}, t) = \int \frac{d^3k}{(2\pi)^3} \left[A_\mathbf{k} e^{-i(\omega_k t - \mathbf{k}\cdot\mathbf{x})} + A^*_\mathbf{k} e^{i(\omega_k t - \mathbf{k}\cdot\mathbf{x})}\right]$$

which we recognize as the sum of sine and cosine waves.

This Fourier technique will form the basis of many of our following solutions, and we will often just consider the evolution of a each Fourier mode individually.

15.3. Adding Gravity

The gravitational potential Φ modifies the Euler equation

$$\rho\left(\frac{\partial}{\partial t} + \mathbf{u}\cdot\nabla\right)\mathbf{u} = -\nabla P - \rho\nabla\Phi$$

and introduces the Poisson equation to relate density to the potential

$$\nabla^2\Phi = 4\pi G\rho$$

As before, we consider our uniform background solution, but we hit a snag with the Poisson equation since

$$\nabla^2\bar{\Phi} = 0 \neq 4\pi G\bar{\rho}$$

Let us just ignore this for now (it will get fixed when we introduce expansion) and just introduce a perturbation to the gravitational potential

$$\Phi = \bar{\Phi} + \delta\Phi$$

As before, we can linearize to get

$$\begin{aligned}
\frac{\partial\delta\rho}{\partial t} &= -\bar{\rho}\nabla\cdot\mathbf{u} \\
\bar{\rho}\frac{\partial\mathbf{u}}{\partial t} &= -\nabla\delta P - \bar{\rho}\nabla\delta\Phi \\
\nabla^2\delta\Phi &= 4\pi G\delta\rho \\
\delta P &= c_s^2\delta\rho
\end{aligned}$$

We can combine these the same way to get a single equation

$$\left(\frac{\partial^2}{\partial t^2} - c_s^2\nabla^2 - 4\pi G\bar{\rho}\right)\delta\rho = 0$$

or in Fourier space

$$\left(\frac{\partial^2}{\partial t^2} + c_s^2 k^2 - 4\pi G\bar{\rho}\right)\delta\rho(\mathbf{k}, t) = 0$$

We notice there is a critical wavenumber k_J at which the frequency of oscillations is zero. This scale is known as the Jeans scale. At wavenumbers $k > k_J$, the equation behaves like a simple harmonic oscillator, while below this scale, the equation exponentially grows. Recalling that increasing wavenumbers correspond to decreasing physical scales, the above is just the statement that on small scales, pressure fluctuations dominate and you get sound waves, while on large scales, we get gravity dominating.

The Jeans scale is defined as

$$k_J^2 = \frac{4\pi G\bar{\rho}}{c_s^2}$$

For $k \ll k_J$, we have

$$\left(\frac{\partial^2}{\partial t^2} - 4\pi G\bar{\rho}\right)\delta\rho(\mathbf{k}, t) = 0$$

whose solutions are

$$\delta\rho \sim e^{\pm t/\tau}$$

where $\tau = 1/\sqrt{4\pi G\bar{\rho}}$. This is also known as the Jean's instability.

Part III.

Linear Theory : Tools/Numerics

16. Introduction to CLASS

In lecture, we sketched out the broad structure of the matter power spectrum in its simplest form. A complete analytic treatment of the full shape of the power spectrum is beyond our scope in this class. However, there are standard codes that allow one to compute these as a function of cosmological parameters.

The two most popular today are CAMB and CLASS. This notebook aims to give you a quick introduction to using CLASS to compute the matter power spectrum and related quantities. CLASS can also be used to compute the CMB power spectra, and we will return to this at the end of this class.

16.1. Installing CLASS

A full description of CLASS, including installation instructions, can be found at the CLASS website. If all you need is the Python interface, you could install it via

```
pip install classy
```

or a similar equivalent option. As you get more advanced, you might want to install the full CLASS package. (Note that if you are taking ASTR6000, this will already be installed in your Jupyter environment.)

The website also has links to a number of tutorials/course material that go into significantly more detail than we can here. Much of these notes are derived from these.

16.2. Explanatory Input

CLASS has many parameters that can be set, and the simplest documentation of these that I know of is the explanatory input file that it ships with.

You can find this here and I would suggest browsing it.

16.3. A First Example: Matter Power Spectrum

Let's compute and plot the matter power spectrum at redshift z=0 using CLASS. Start by importing the necessary libraries.

Next, we create an instance of the CLASS class and set the cosmological parameters. The final step will be to call the `compute()` method to run CLASS.

```
Cosmological parameters set in CLASS:
------------------------------
Omega_b: 0.04565
Omega_cdm: 0.24490
Omega_m: 0.29055
h: 0.70
sigma_8: 0.83
n_s: 0.9649
------------------------------
```

We now have the matter power spectrum computed and can access it via the `pk()` method. This method takes two arguments: the wavenumber `k` in units of 1/Mpc, and the redshift `z`.

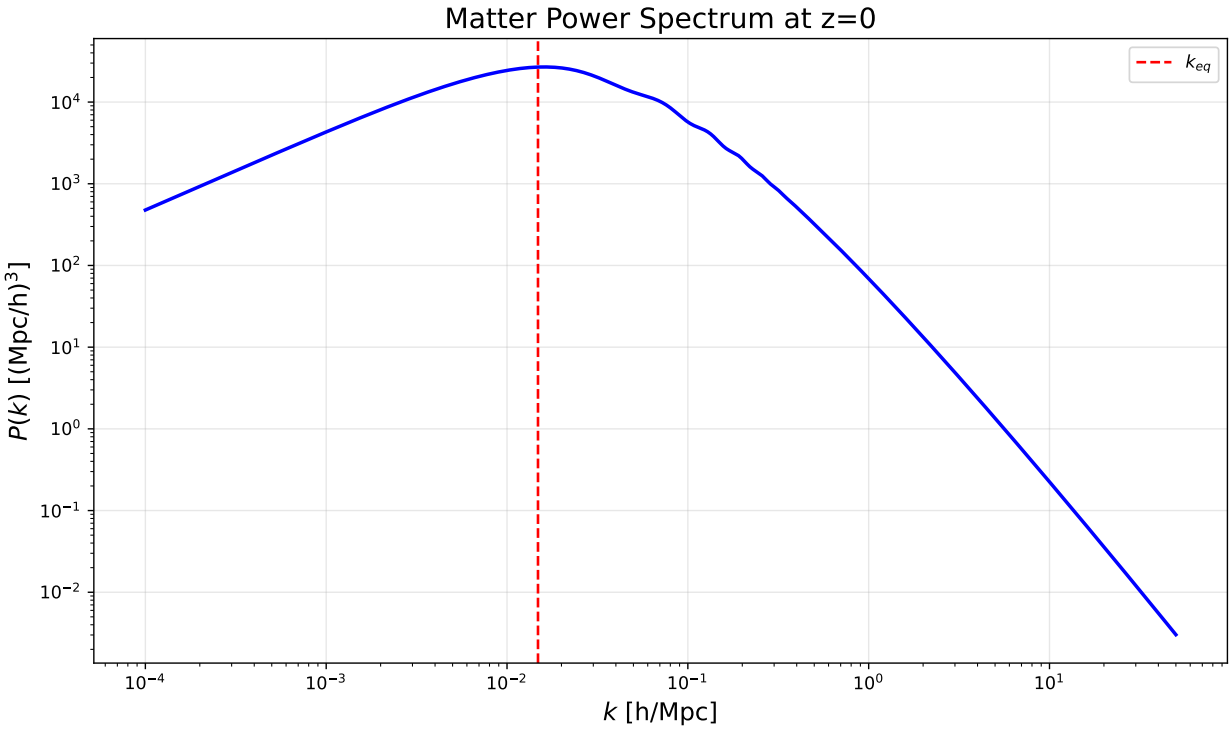


Figure 16.1.: Matter power spectrum at z=0

16.4. Parameter Variations

With this in hand, we can easily explore how the matter power spectrum changes as we vary cosmological parameters. We'll do some simple examples here, paralleling the discussion in lecture.

16.4.1. Varying the primordial amplitude A_s

The simplest variation is to change the amplitude of the primordial power spectrum, A_s. This will simply scale the matter power spectrum up or down without changing its shape.

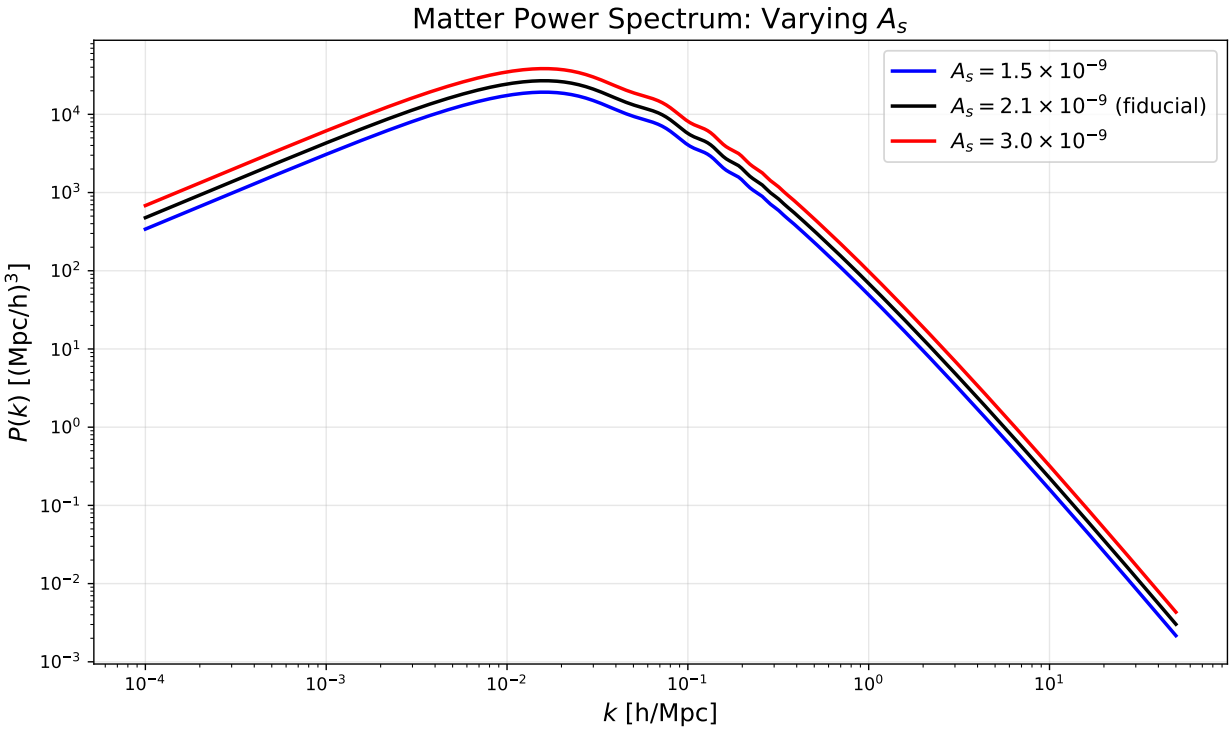


Figure 16.2.: Effect of varying the primordial amplitude A_s

16.4.2. Varying the primordial slope n_s

Changing the spectral index will change the overall tilt of the power spectrum on both sides of the turnover.

We also plot our expectations for the slope in the two regimes for the reference model :

- For $k \ll k_{eq}$, we expect $P(k) \propto k^{n_s}$.
- For $k \gg k_{eq}$, we expect $P(k) \propto k^{n_s-4}$.
- We also show a variant $P(k) \propto k^{n_s-4} \ln^2(k/k_{eq})$ anchored at $k = 5\,h/\mathrm{Mpc}$.

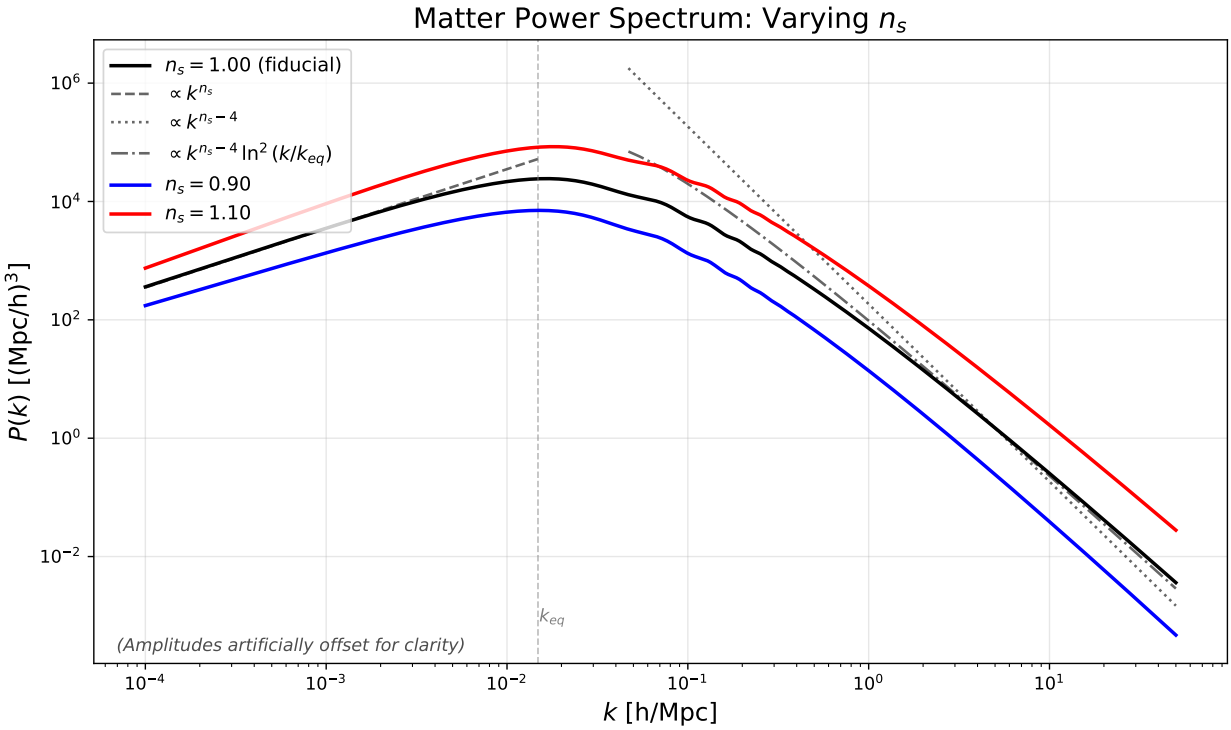


Figure 16.3.: Effect of varying the spectral index n_s

16.4.3. Varying k_{eq}

Let's now consider varying the matter-radiation equality scale k_{eq} and therefore shift the location of the turnover in the power spectrum. This is determined by the matter density $\Omega_m h^2$ and the radiation density (which is fixed by the CMB temperature).

Recall from lecture that the turnover scale is given by

$$k_{eq} = H_0 \sqrt{\frac{2\Omega_m^2}{\Omega_r}}$$

where Ω_r is the radiation density parameter.

Physically, the temperature of the CMB is very well measured, so the photon density is fixed. We could varying the number of relativistic species (e.g. neutrinos) to change the radiation density, but for simplicity we will just vary $\Omega_m h^2$ to shift k_{eq}. We will do so, but keep the ratio of baryons to CDM fixed to avoid changing the shape of the power spectrum in other ways.

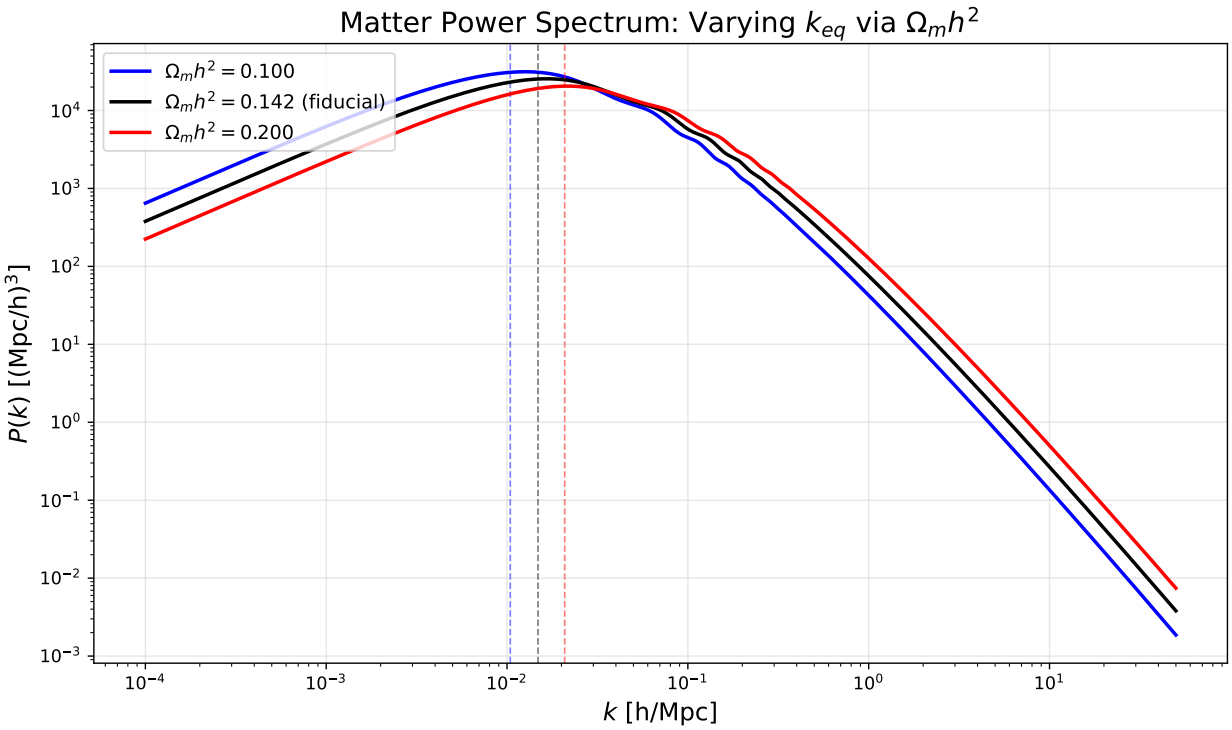


Figure 16.4.: Effect of shifting k_{eq} by varying $\Omega_m h^2$

17. The Correlation Function

The correlation function $\xi(r)$ is the 3D Fourier transform of the power spectrum $P(k)$:

$$\xi(r) = \frac{1}{2\pi^2} \int_0^\infty k^2 P(k) \frac{\sin(kr)}{kr} dk$$

A word on the limits of integration here. While the limits formally run from 0 to ∞, in practice, we often do not have the power spectrum defined at arbitrarily small or large k. For a LCDM like power spectrum, the low-k limit is often not a concern, since the power is dropping off at large scales. The high-k limit is more tricky, since it is easy to get ringing artifacts in the correlation function if we truncate at too low a k value or too abruptly. In practice, we avoid this by smoothing the correlation function by truncating the integral with a Gaussian of scale $R \sim 0.1 - 1\,h^{-1}\,\mathrm{Mpc}$. This does not affect the correlation function at large scales.

17.1. Code to compute the correlation function

To compute the correlation function, we can use the `scipy.integrate.quad` function to perform the integral. Note that we need to do this for each value of r we want to compute $\xi(r)$ at, which can be computationally expensive. (There are more efficient methods to do this [e.g. look up FFTLog]), but we'll stick to the direct integration for simplicity.

We'll also assume that we have a function `P(k)` that returns the power spectrum at a given k.

Finally, we include a Gaussian smoothing factor in the integral to avoid ringing artifacts.

17.2. Examples

Let's explore examples here.

17.2.1. LCDM correlation function with a BAO feature

We can now use our `xi_of_r` function to compute the correlation function for a LCDM power spectrum calculated using CLASS from the previous notebook.

17.2.2. Compute $P(k)$ with CLASS

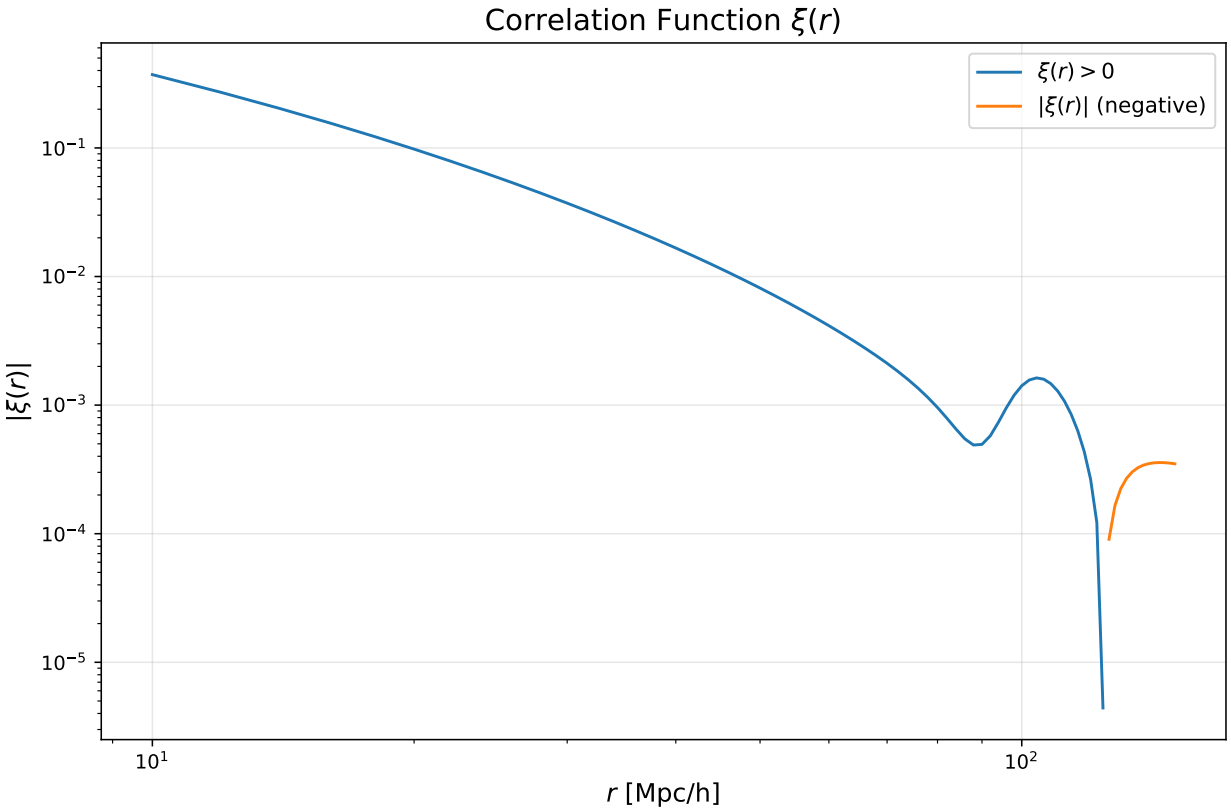
We first compute a fiducial LCDM power spectrum at $z = 0$, and define a wrapper that returns $P(k)$ in units of $(\mathrm{Mpc}/h)^3$ for k in h/Mpc.

17.2.3. Compute $\xi(r)$

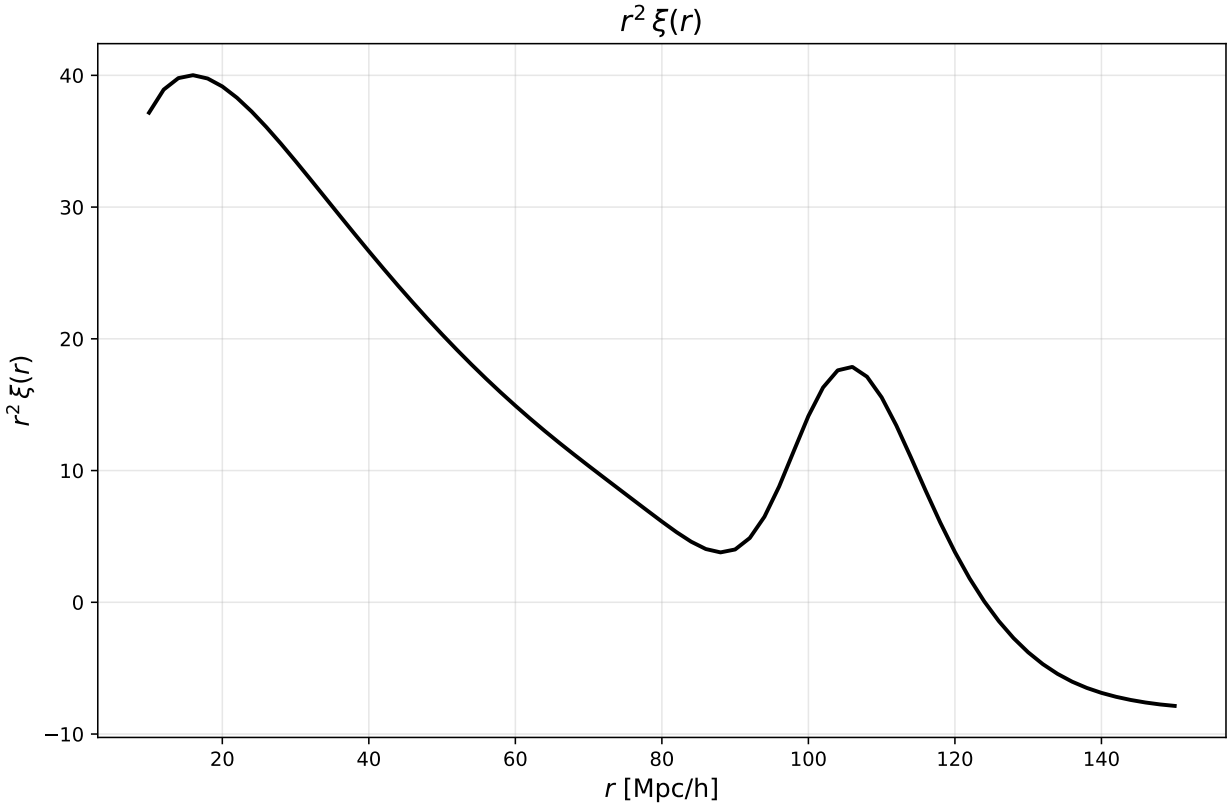
We evaluate $\xi(r)$ from $r = 10$ to $150\,\mathrm{Mpc}/h$ in $\sim 2\,\mathrm{Mpc}/h$ bins.

17.2.4. Plot $\xi(r)$ on a log-log scale

We split the curve by sign so that the log scale remains well-defined.



17.2.5. Plot $r^2\,\xi(r)$ on a linear scale

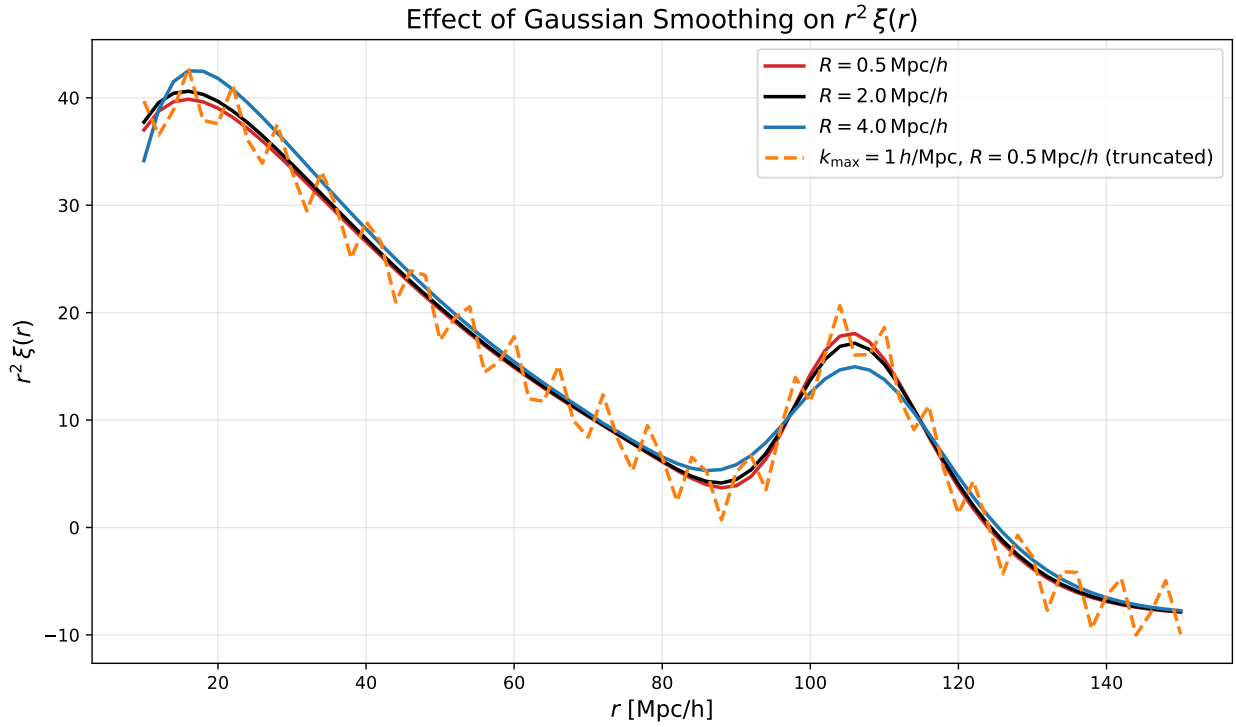


17.2.6. Demonstration of smoothing/ringing artifacts from truncation

To see the effect of truncating the integral at too low a k value, we can compute $\xi(r)$ with a much lower $k_{\rm max}$ and compare to our previous result.

We compare three standard cases with different smoothing scales: $R = 0.5\,\mathrm{Mpc}/h$, $2.0\,\mathrm{Mpc}/h$, and $4.0\,\mathrm{Mpc}/h$, all computed with the full $k_{\rm max} = 50\,h/\mathrm{Mpc}$.

We also show a pathological case where we truncate the integral at $k_{\rm max} = 1\,h/\mathrm{Mpc}$ with modest smoothing ($R = 0.5\,\mathrm{Mpc}/h$). This demonstrates the characteristic oscillations that appear when the truncation is too aggressive—these are the ringing artifacts we discussed. In practice, this shows why careful choice of $k_{\rm max}$ and smoothing scale is important when computing the correlation function from observationally-limited power spectrum measurements.

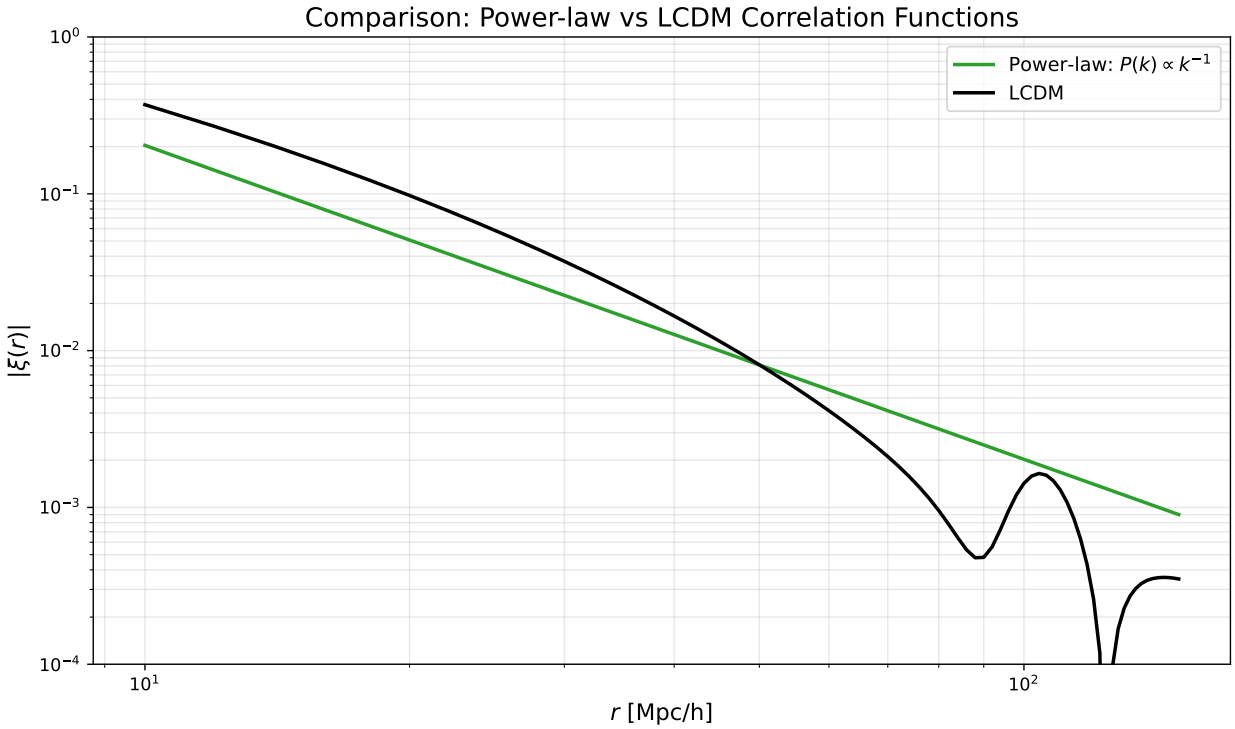


17.2.7. Power-law spectrum example

Let's also compute the correlation function for a simple power-law spectrum $P(k) = A\,k^{-n}$. This serves as a toy model that's easier to understand analytically (though the real LCDM spectrum has more structure). We'll compare this directly to the LCDM case.

Note that for the power-law case, the correlation function will also be a power-law in r

$$\xi(r) \propto r^{n-3}$$


Comparison: Power-law vs LCDM Correlation Functions
Power-law: $P(k) \propto k^{-1}$
LCDM
$|\xi(r)|$
r [Mpc/h]
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^1
10^2

18. Comparing Linear Theory to N-body Simulations

Our goal here is to compare the power spectra from the Quijote simulations to the predictions of linear theory.

You can find the details of the Quijote simulations here.

We are using a subset of the Quijote simulations for this analysis; these will be made available on the cluster (the directory will be shared separately). The notebook here uses a directory structure set up on my laptop/desktop, so you will need to adjust these if you want to run this notebook.

18.1. Loading the Power Spectrum Data

Let us start by loading in one simulation from the Quijote suite and plotting the power spectrum at $z = 0$ and compare it to the input linear theory power spectrum.

In the plot above, observe the following features:

- On large scales (small k), the simulation power spectrum appears to match the linear theory prediction.
- There is significant scatter on the largest scales; this is "sample variance" due to the finite size of the box.
- On small scales (large k), the simulation power spectrum shows a clear enhancement relative to linear theory, due to nonlinear gravitational clustering. The ratio plot in the lower panel makes this enhancement more evident.
- The simulation does not extend to very large scales due to the finite box size, which is why we see a cutoff at low k.

Quijote has a very large number of simulations. Let us take 10 of these and average them together to see how the scatter reduces on large scales.

18.2. Redshift Evolution

We can now turn to the redshift evolution of the power spectrum. Quijote provides power spectra at multiple redshifts, so we can compare how the simulation evolves with redshift to the linear theory prediction.

As an initial comparison, we'll look at a single simulation. We will estimate the growth factor directly from the simulation by directly taking the ratio of the first power spectrum point in k to the value at $z = 0$, and then apply this to the linear theory power spectrum.

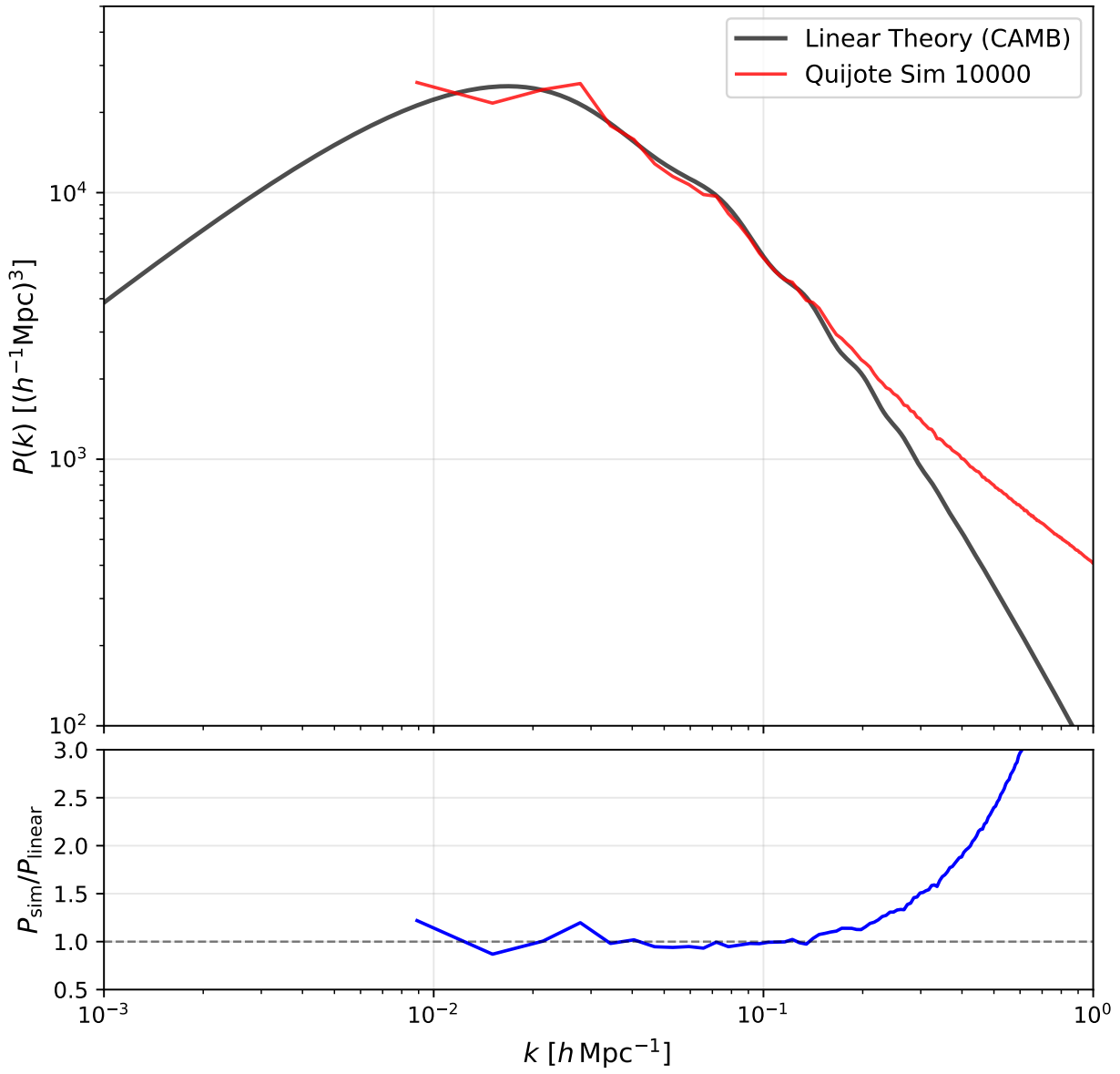


Figure 18.1.: Comparison of power spectrum from Quijote N-body simulation (z=0) with linear theory prediction from CAMB. The upper panel shows the power spectra, while the lower panel shows the ratio P(k)_sim / P(k)_linear, highlighting the nonlinear enhancement on small scales.

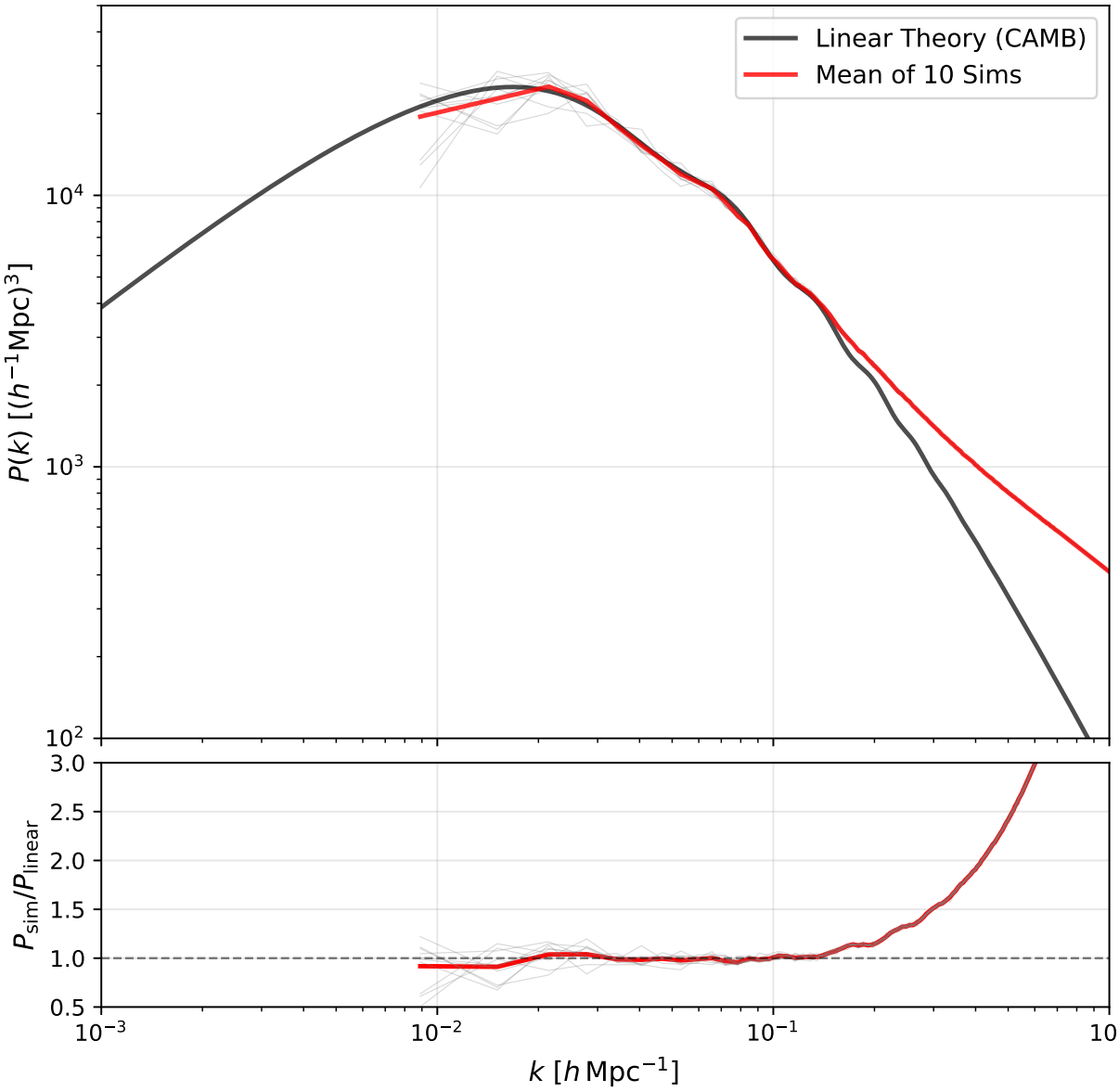


Figure 18.2.: Comparison of averaged power spectrum from 10 Quijote simulations with linear theory. Individual simulations are shown as faint gray lines, the mean is in red, and linear theory is in black. Averaging reduces the sample variance on large scales.

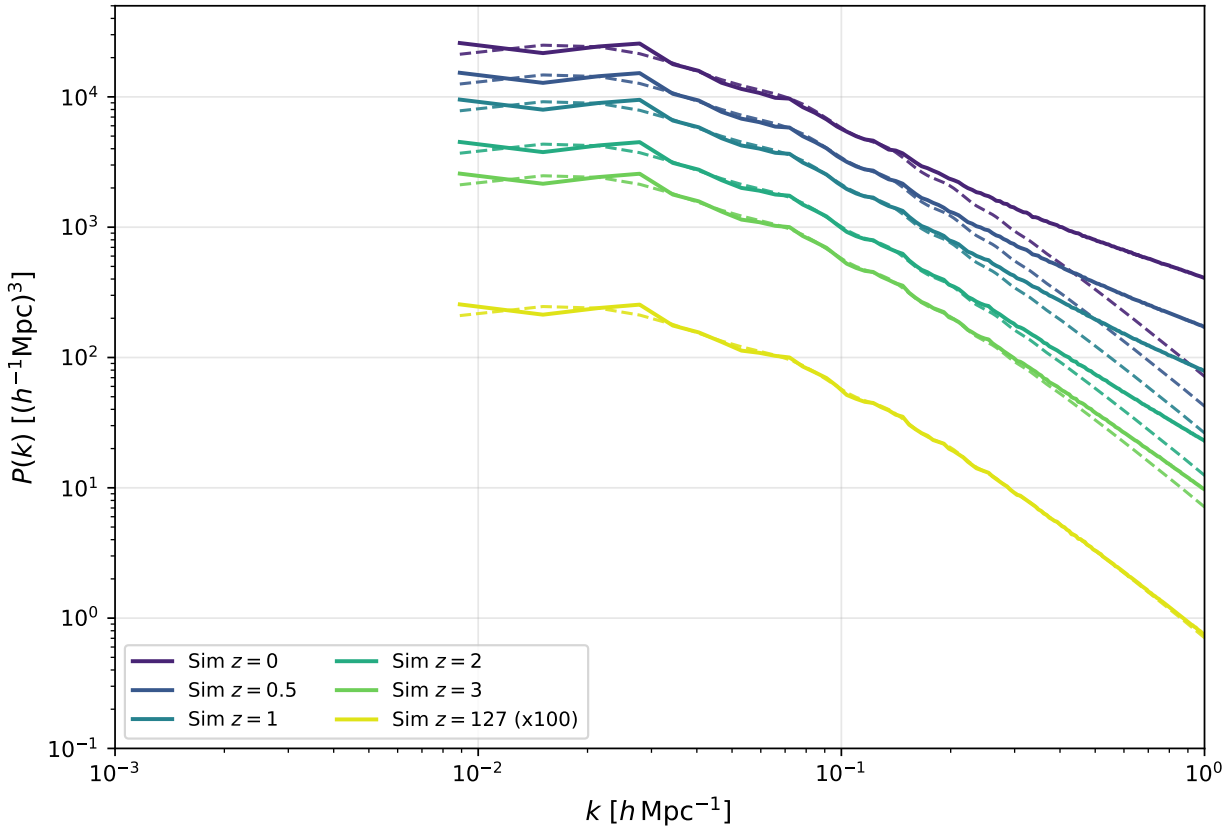


Figure 18.3.: Power spectra at multiple redshifts for Quijote simulation 10000 (solid) and corresponding growth-scaled linear theory predictions (dashed). The z=127 curves are multiplied by 100 for visual comparison in the same y-range.

The figure above shows the growth of fluctuations clearly: as time progresses toward lower redshift, the overall amplitude of $P(k)$ increases across all scales.

Comparing each simulation curve to its growth-scaled linear prediction (same color, dashed) shows where linear theory begins to fail. The disagreement is strongest at large k, where nonlinear mode coupling boosts power above linear expectations.

In the next figure, we compare exact k-modes directly in ratio form. This removes much of the broad amplitude evolution and makes two effects easier to see: (i) reduced apparent scatter when matching the same modes across redshifts, and (ii) scale-dependent departures from linear evolution.

18.3. The Growth Function

We can now directly compare the growth function estimated from the simulations to the theoretical prediction from linear theory.

To compute the theoretical growth function, we'll use the formula in the HW, valid for a flat Universe with a cosmological constant (which matches the Quijote cosmology).

$$D(a) = \frac{5\Omega_m H(a)}{2} \int_0^a \frac{da'}{(a' H(a'))^3}$$

where the factors are chosen so that $D(a) \approx a$ at early times when matter dominates.

We won't make this into a full-fledged function here, since we'll only need it once in this notebook, but you can easily turn this into a reusable function if you want to compute growth factors for redshifts/cosmologies. Note that this only works for a cosmological constant.

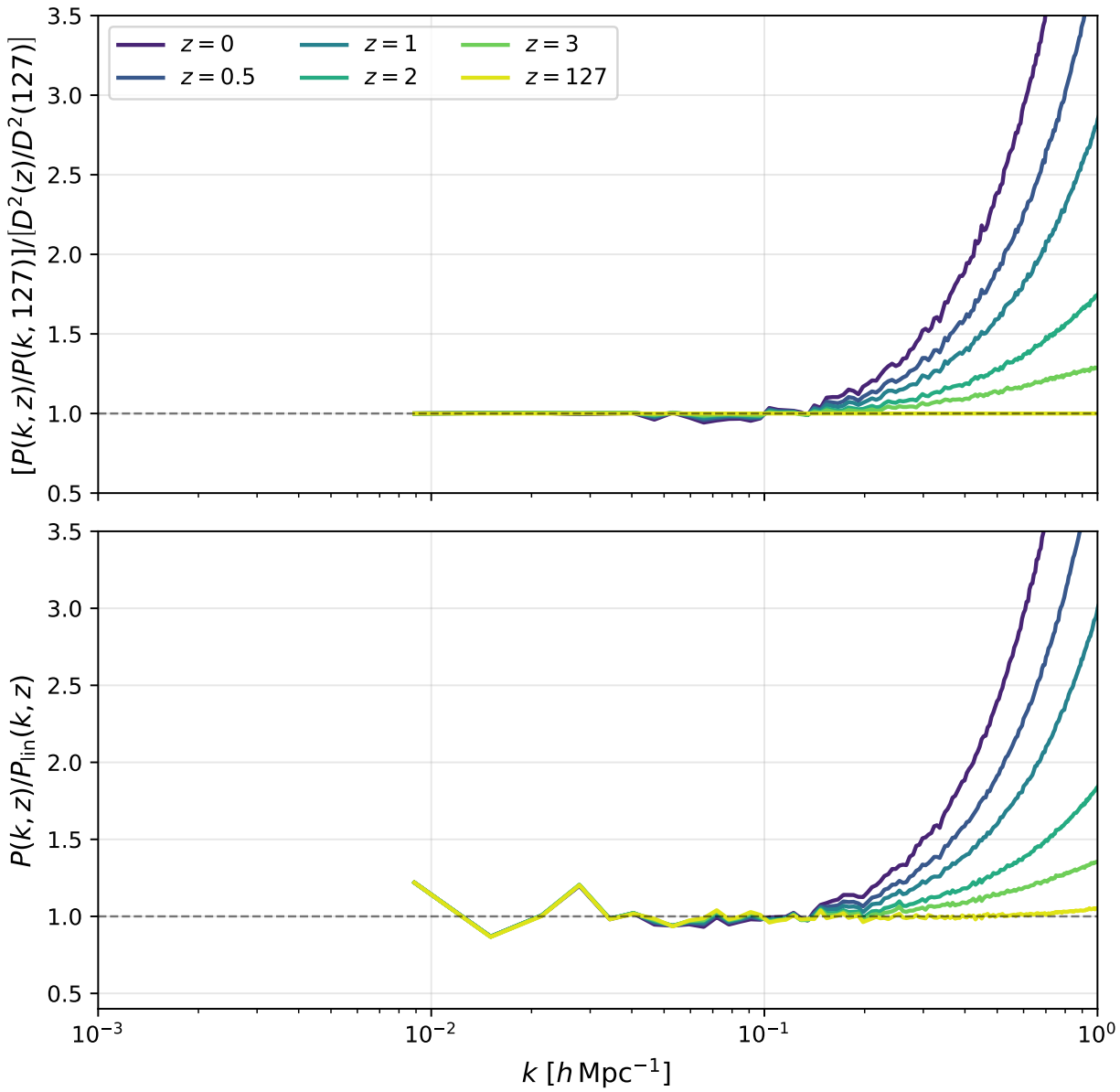


Figure 18.4.: Redshift evolution ratios for Quijote simulation 10000. Top: growth-scaled ratio relative to z=127, [P(k,z)/P(k,127)]/[D$^{2(z)/D}$2(127)], with linear y-axis. Bottom: P(k,z)/P_lin(k,z), where P_lin(k,z)=D(z)^2 P_lin(k,z=0) and D(z)^2 is estimated from the largest-scale mode.

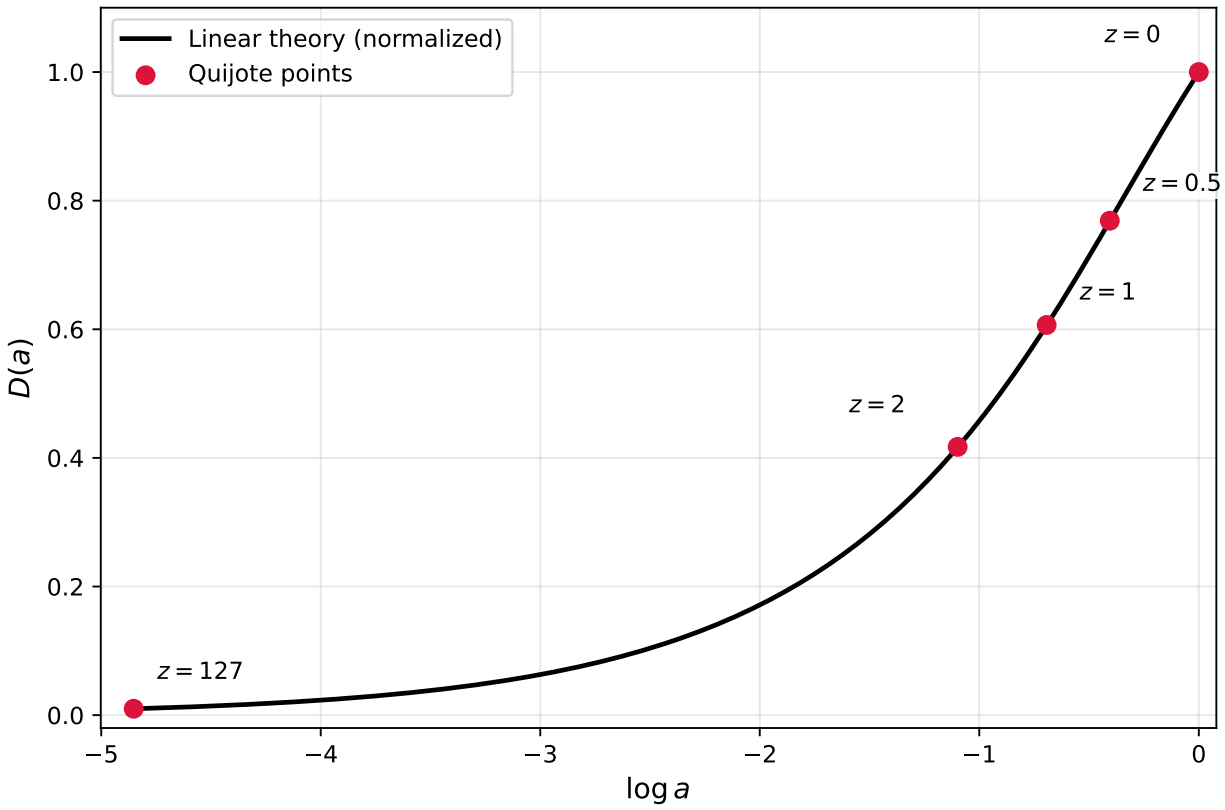


Figure 18.5.: Linear-theory growth factor D(a), normalized so D(1)=1, compared with Quijote growth estimates from the same large-scale mode at four redshifts.

Part IV.

Nonlinear Structure Formation I

19. Computing the Quijote Mass Function

Our goal in this notebook is to compute and plot the Quijote mass function.

You can find the details of the Quijote simulations here.

We are using a subset of the Quijote simulations for this analysis; these will be made available on the cluster (the directory will be shared separately). The notebook here uses a directory structure set up on my laptop/desktop, so you will need to adjust these if you want to run this notebook.

19.1. Loading the Halo Catalogs

Unlike our previous analyses of the power spectra, we will need to actually load and process the halo catalogs. These are stored as binary files in the Quijote data directory, and so, we will use a simple helper function to read these in (for now, we'll just need the halo masses).

The halo catalogs are organized by simulation number and snapshot number, where the snapshot number corresponds to different redshifts. In the Quijote simulations, snapshot 4 corresponds to $z = 0$.

```
Reading halo catalog from /Users/npadmana/data/teaching/ASTR6000-SP26/Quijote-
Halos-HR/0/group_tab_004.0
Number of groups in file: 2946947
Read 2946947 halo masses. Min mass: 1.64e+12 M_sun/h, Max mass: 5.38e+15 M_sun/h
```

19.2. Computing the Mass Function

The mass function is nothing but a histogram of halo masses. Let us set up the mass bins so that we can accumulate the halo mass functions for multiple simulations. We'll use logarithmic bins from 5×10^{12} to $5 \times 10^{15}\, M_\odot/h$ with 30 bins in total.

We'll test this out for a single simulation first.

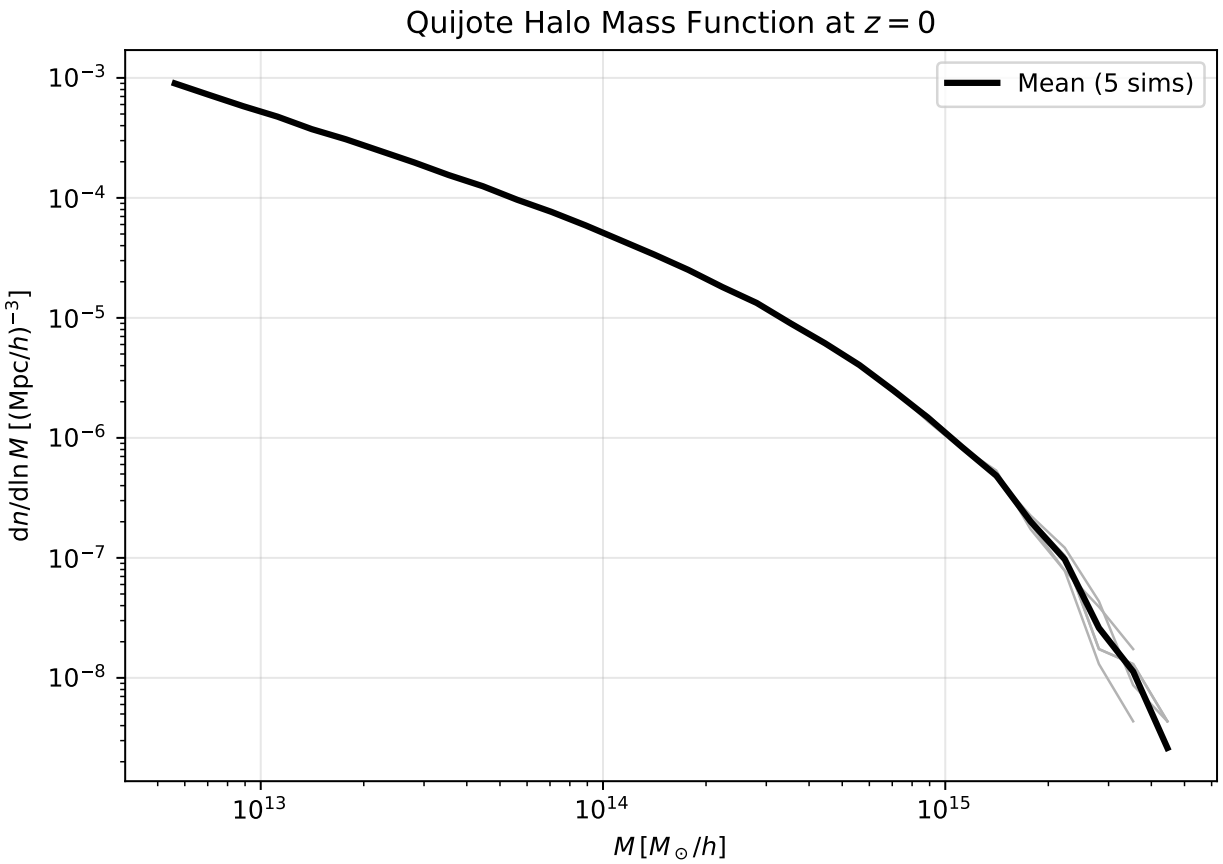
Finally, we'll define overall normalizations, including the volume and the bin widths, so that what we get is dn/dlnM in units of $(\mathrm{Mpc/h})^{-3}$.

```
Mass bin edges (log10 M_sun/h): [12.69897 12.79897 12.89897 12.99897 13.09897 13.19897 13.2989
 13.49897 13.59897 13.69897 13.79897 13.89897 13.99897 14.09897 14.19897
 14.29897 14.39897 14.49897 14.59897 14.69897 14.79897 14.89897 14.99897
 15.09897 15.19897 15.29897 15.39897 15.49897 15.59897 15.69897]
Mass function (counts per bin): [207251 165954 133980 109590  86609  70920  56822  45131  3573
```

```
  22181   17621   13443   10425    7755    5786    4203    2987    2137    1404
    932     546     357     191     123      40      18       9       4       0]
```

We can now compute and plot the mass function for the first 5 Quijote simulations. We'll plot the individual mass functions, but also the average mass function.

```
Reading halo catalog from /Users/npadmana/data/teaching/ASTR6000-SP26/Quijote-
Halos-HR/0/group_tab_004.0
Number of groups in file: 2946947
Read 2946947 halo masses. Min mass: 1.64e+12 M_sun/h, Max mass: 5.38e+15 M_sun/h
Reading halo catalog from /Users/npadmana/data/teaching/ASTR6000-SP26/Quijote-
Halos-HR/1/group_tab_004.0
Number of groups in file: 2946445
Read 2946445 halo masses. Min mass: 1.64e+12 M_sun/h, Max mass: 3.97e+15 M_sun/h
Reading halo catalog from /Users/npadmana/data/teaching/ASTR6000-SP26/Quijote-
Halos-HR/2/group_tab_004.0
Number of groups in file: 2945544
Read 2945544 halo masses. Min mass: 1.64e+12 M_sun/h, Max mass: 4.39e+15 M_sun/h
Reading halo catalog from /Users/npadmana/data/teaching/ASTR6000-SP26/Quijote-
Halos-HR/3/group_tab_004.0
Number of groups in file: 2945565
Read 2945565 halo masses. Min mass: 1.64e+12 M_sun/h, Max mass: 4.01e+15 M_sun/h
Reading halo catalog from /Users/npadmana/data/teaching/ASTR6000-SP26/Quijote-
Halos-HR/4/group_tab_004.0
Number of groups in file: 2945565
Read 2945565 halo masses. Min mass: 1.64e+12 M_sun/h, Max mass: 4.01e+15 M_sun/h
```



19.3. Press-Schechter Comparison

We can compare these results to the Press-Schechter mass function. Recall that the Press-Schechter mass function is given by:

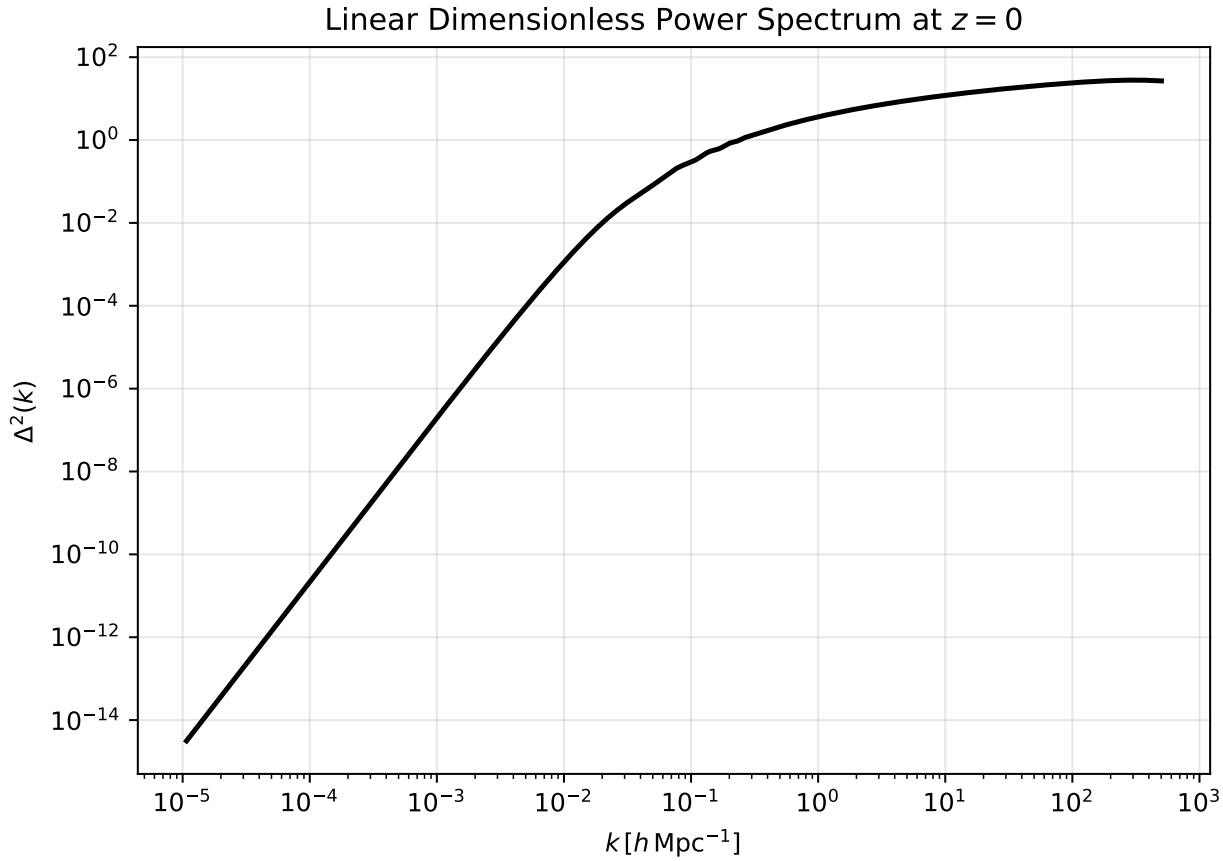
$$\frac{\mathrm{d}n}{\mathrm{d}\ln M} = \sqrt{\frac{2}{\pi}}\frac{\bar{\rho}}{M}\nu e^{-\nu^2/2}\left|\frac{\mathrm{d}\ln\sigma}{\mathrm{d}\ln M}\right|$$

where $\nu = \delta_c/\sigma(M)$, $\delta_c \approx 1.686$, and $\sigma(M)$ is the variance of the linear density field smoothed on scale M.

There are a few steps to compute this:

1. Compute the linear matter power spectrum at $z = 0$. We could use CLASS for this, but since Quijote has already provided the linear power spectrum, we will just load it in from their data files.
2. Compute $\sigma(M)$ by integrating the power spectrum with a top-hat filter. Since we'll need this at a number of scales, we'll precompute on a grid of masses and then interpolate $\ln\sigma$ as a function of $\ln M$.
3. Compute $\mathrm{d}\ln\sigma/\mathrm{d}\ln M$. Since we have interpolated $\ln\sigma$, this can be directly calculated from the interpolation.
4. Compute the Press-Schechter mass function and plot it alongside the Quijote results.

Let's go through these steps one by one.



Next, we compute $\sigma(M)$ by integrating the power spectrum with a top-hat filter. We'll set up a grid of masses from 10^{12} to $10^{16}\,M_\odot/h$, compute the corresponding radii, and then perform the integration for each mass.

We'll use the spherical top-hat window function in Fourier space:

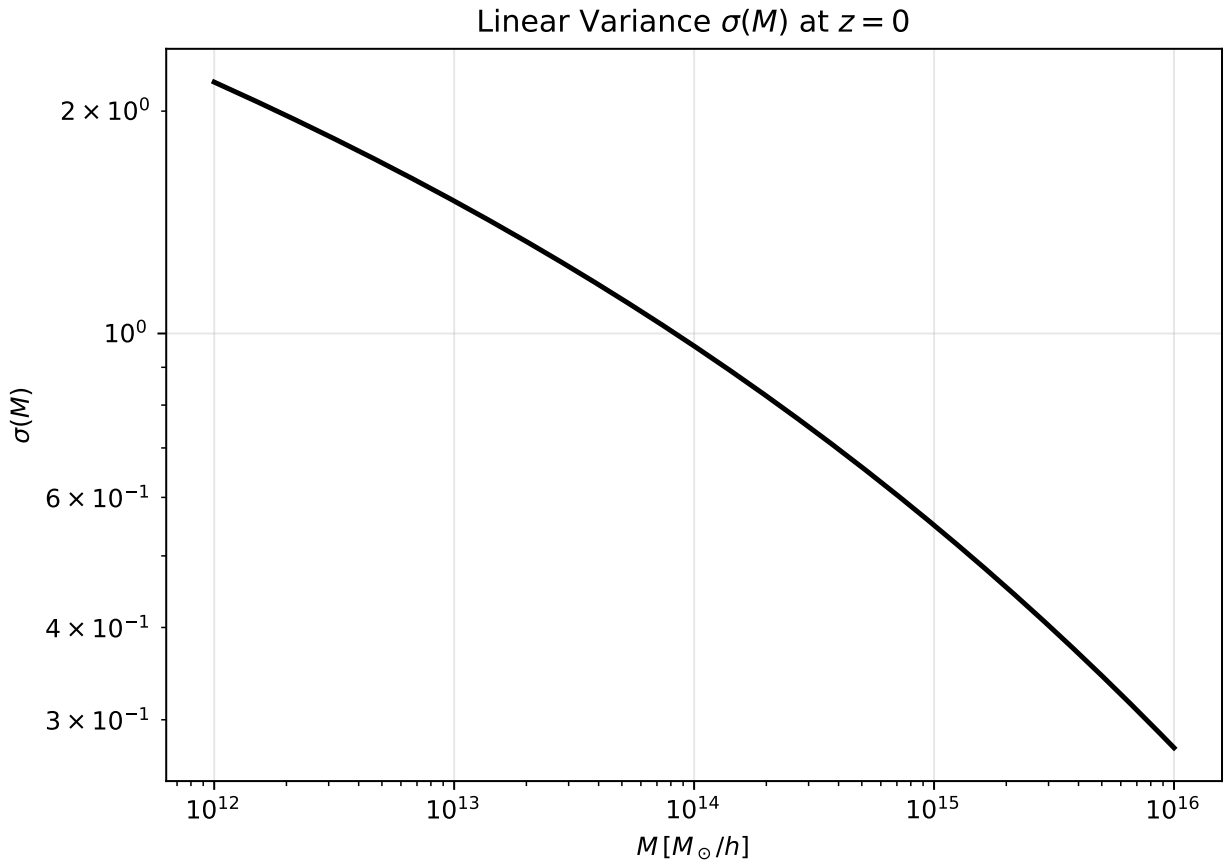
$$W(kR) = \frac{3}{(kR)^3}\left[\sin(kR) - kR\cos(kR)\right]$$

For the integration, we'll use the trapezoidal rule from `numpy` on a uniform grid in $\ln k$, integrating the equivalent form

$$\sigma^2(R) = \int \Delta^2(k)\, W^2(kR)\, \mathrm{d}\ln k.$$

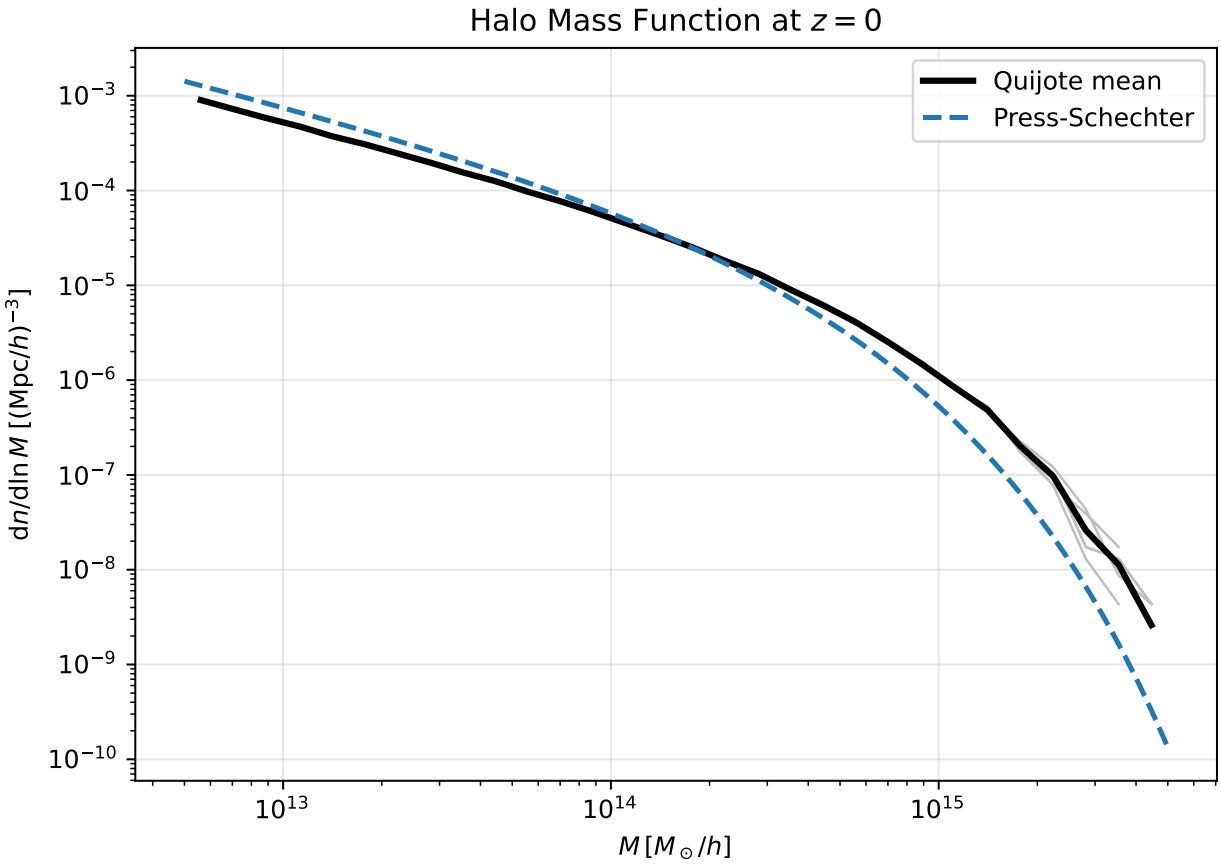
And then, we'll interpolate $\ln\sigma$ as a function of $\ln M$ for later use.

As a test, we'll also compute $\sigma(R = 8\,\mathrm{Mpc}/h)$, i.e. σ_8. The expected value for the Quijote cosmology is 0.834.




```
sigma(R=8 Mpc/h) = 0.8340
```

We can now compute the mass function using the Press-Schechter formula and then compare with the Quijote results.



19.4. Sheth-Tormen Extension

The Press-Schechter multiplicity function is

$$f_{\mathrm{PS}}(\nu) = \sqrt{\frac{2}{\pi}}\,\nu\, e^{-\nu^2/2}.$$

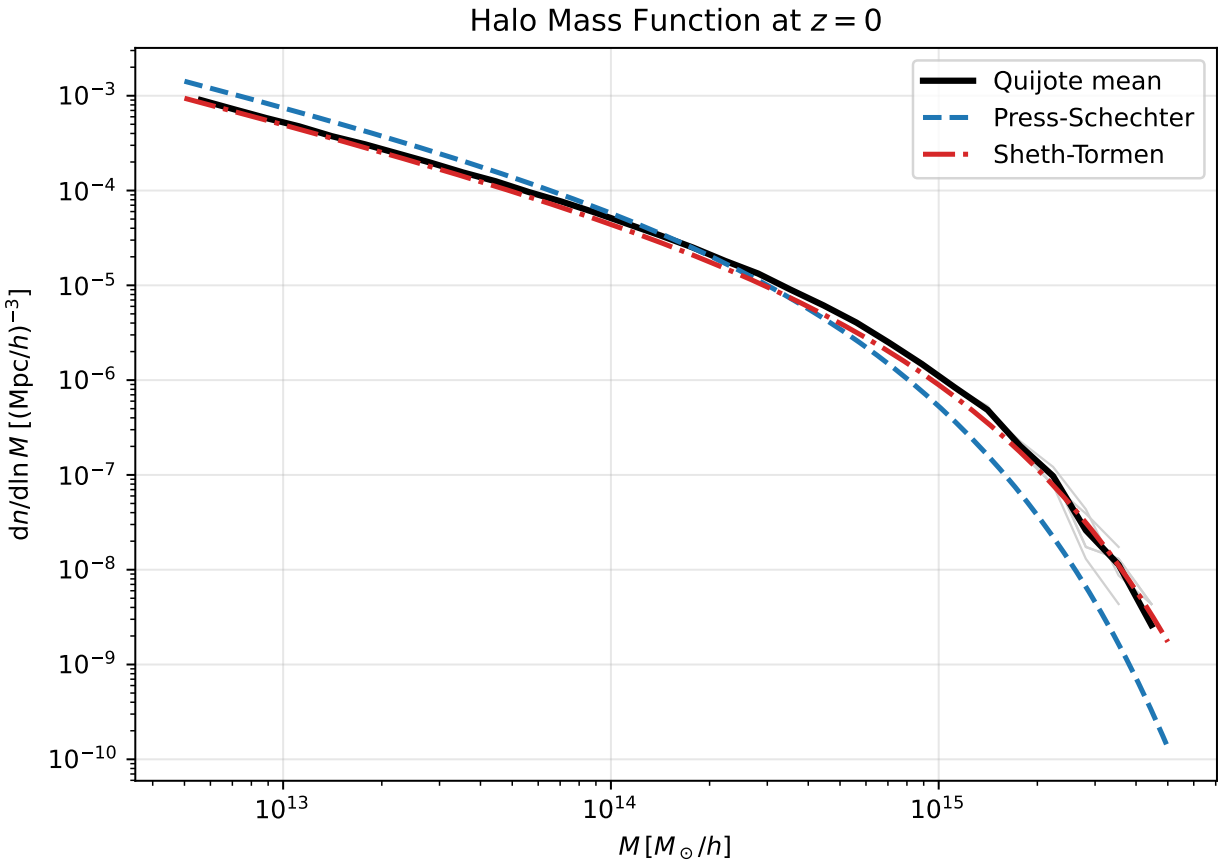
The Sheth-Tormen form modifies this to better match simulation-based halo abundances:

$$f_{\mathrm{ST}}(\nu) = A\sqrt{\frac{2a}{\pi}}\,\nu\,[1 + (a\nu^2)^{-p}]\,e^{-a\nu^2/2},$$

with typical parameters $A = 0.3222$, $a = 0.707$, and $p = 0.3$.

Using either multiplicity function,

$$\frac{\mathrm{d}n}{\mathrm{d}\ln M} = \frac{\bar{\rho}}{M} f(\nu)\left|\frac{\mathrm{d}\ln\sigma}{\mathrm{d}\ln M}\right|.$$


Halo Mass Function at $z = 0$
Quijote mean
Press-Schechter
Sheth-Tormen
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
$\mathrm{d}n/\mathrm{dln}\,M\,[(\mathrm{Mpc}/h)^{-3}]$
10^{13}
10^{14}
10^{15}
$M\,[M_\odot/h]$

20. Cosmological Simulations II: The PM Algorithm

20.1. Recap: The Equations of Motion

In the previous lecture, we started from the collisionless Boltzmann (Vlasov) equation coupled to the Poisson equation — the Vlasov–Poisson system — which governs the evolution of collisionless dark matter in comoving coordinates $\mathbf{x}$ (with $\mathbf{r} = a(t)\mathbf{x}$). Since the 6D phase space is too large to simulate directly, we Monte-Carlo sample the distribution function with N particles in a periodic box of comoving side L, each carrying a mass

$$m_p = \Omega_m \bar{\rho} \frac{L^3}{N}$$

We then derived the equations of motion for each particle. Starting from the full Lagrangian $L = \frac{1}{2}m|\dot{a}\mathbf{x} + a\dot{\mathbf{x}}|^2 - m\varphi$, we decomposed the gravitational potential as $\varphi = \varphi_{\rm bg} + \Phi$ where $\varphi_{\rm bg} = \frac{2\pi}{3}G\bar{\rho}a^2|\mathbf{x}|^2$. The background and cross terms cancel exactly (using the Friedmann equation $\ddot{a}/a = -4\pi G\bar{\rho}/3$), leaving the **peculiar Lagrangian**:

$$\mathcal{L}_{\rm pec} = \frac{1}{2}ma^2\dot{\mathbf{x}}^2 - m\Phi(\mathbf{x}, t)$$

where Φ is the peculiar potential satisfying the **Poisson equation**:

$$\nabla^2\Phi = 4\pi G\bar{\rho}a^2\delta$$

The conjugate momentum is $\mathbf{p} = ma^2\dot{\mathbf{x}}$, and the Euler–Lagrange equation gives

$$\ddot{\mathbf{x}} + 2\frac{\dot{a}}{a}\dot{\mathbf{x}} = -\frac{1}{a^2}\nabla_x\Phi$$

with the familiar Hubble drag term $2H\dot{\mathbf{x}}$.

We also saw the **total derivative trick**: adding $d/dt(\frac{1}{2}ma\dot{a}|\mathbf{x}|^2)$ to $\mathcal{L}_{\rm pec}$ gives $\mathcal{L}'$ with conjugate momentum $\mathbf{p}' = ma\dot{\mathbf{r}}$ (proportional to the total velocity), and the simpler equations $d\mathbf{x}/dt = \mathbf{p}'/(ma^2)$, $d\mathbf{p}'/dt = -m\nabla_x\Phi$ — no Hubble drag, but the momentum now includes the Hubble flow.

The task is now to integrate these equations efficiently and stably over cosmological timescales — which brings us to symplectic methods.

20.2. Symplectic Integration

20.2.1. Why Symplectic?

The equations of motion derived from $\mathcal{L}_{\text{pec}}$ are Hamiltonian, and Hamiltonian systems have a fundamental geometric property: they preserve the symplectic form on phase space. Loosely speaking, this means that the flow of trajectories preserves phase-space volume (Liouville's theorem) and the area of any canonical (x_i, p_i) pair.

A generic ODE integrator (e.g., Runge–Kutta) does not respect this structure. Over many timesteps, such integrators introduce systematic drift in conserved quantities — the energy "random walks" away from its true value. For cosmological simulations, where we may take 10^3–10^4 timesteps, this secular drift is unacceptable.

Symplectic integrators are constructed to exactly preserve the symplectic structure at each step. The key consequence: while the energy at each step has an $\mathcal{O}(\Delta t^n)$ error (where n is the order of the integrator), this error is **bounded** — it oscillates rather than drifts. There is no secular growth.

> **!** No Conserved Energy in an Expanding Universe
>
> In cosmology, the Hamiltonian is explicitly time-dependent (through $a(t)$), so there is no conserved energy. Symplecticity is still valuable, however: it preserves phase-space structure, prevents artificial dissipation or heating of particle orbits, and maintains the correct topology of trajectories. The composition of symplectomorphisms is itself a symplectomorphism — this is the mathematical property that makes the whole approach work.
> For testing purposes, there exists a **Layzer–Irvine equation** that predicts dE/dt given the expansion history — one can integrate this alongside the simulation as a consistency check. We will not derive it here.

20.2.2. The Splitting Method

Consider a Hamiltonian of the form

$$H = A(\mathbf{p}) + B(\mathbf{x})$$

Each piece alone generates a trivially solvable flow:

- $H = A(\mathbf{p})$: momenta are constant, positions evolve as $\dot{\mathbf{x}} = \partial A/\partial \mathbf{p}$
- $H = B(\mathbf{x})$: positions are constant, momenta evolve as $\dot{\mathbf{p}} = -\partial B/\partial \mathbf{x}$

The idea of **symplectic splitting** is to approximate the full evolution by alternating these sub-steps. Each sub-step is an exact symplectomorphism (it solves a Hamiltonian system exactly), and the composition of symplectomorphisms is a symplectomorphism. So the overall integrator is automatically symplectic, regardless of the step size.

Splitting and the Baker–Campbell–Hausdorff Formula

The splitting approximation can be understood through the BCH formula. For operators $\hat{A}$ and $\hat{B}$,

$$e^{\hat{A}\Delta t}\, e^{\hat{B}\Delta t} = e^{(\hat{A}+\hat{B})\Delta t + \frac{1}{2}[\hat{A},\hat{B}]\Delta t^2 + \cdots}$$

The leading error is $\mathcal{O}(\Delta t^2)$, giving a first-order integrator. By symmetrizing — taking a half-step, a full step, and another half-step — the odd-order error terms cancel, yielding second-order accuracy.

20.2.3. KDK Leapfrog

The most common choice for cosmological simulations is the **Kick-Drift-Kick (KDK)** leapfrog:

1. **Kick** (half step): update momenta by a half timestep using the force
2. **Drift** (full step): update positions by a full timestep using the (now updated) momenta
3. **Kick** (half step): update momenta by another half timestep using the force at the new positions

This is a second-order symplectic integrator: the position and force errors are $\mathcal{O}(\Delta t^2)$ per step.

Why KDK Instead of DKD?

The alternative **Drift-Kick-Drift (DKD)** ordering is equally valid and has the same formal accuracy. KDK is preferred in practice because after each full step, the positions and momenta are synchronized at the same time — making it easy to output snapshots and diagnostics. In DKD, positions lead momenta by a half step, requiring an extra synchronization step for output.

20.3. Equations of Motion in $\ln a$

The peculiar Lagrangian uses coordinate time t, which leads to explicit time-dependence through $a(t)$ and introduces Hubble drag terms in the equation of motion. We can eliminate this by choosing a better time variable.

20.3.1. Changing the Time Variable

Define

$$\eta \equiv \ln a$$

so that $d\eta = da/a = (da/dt)/a\, dt = H\, dt$, or equivalently, $dt = d\eta/H$. This is a monotonic reparametrization of time (as long as $H > 0$, i.e., the universe is expanding).

We now re-derive the Lagrangian, conjugate momentum, and Hamiltonian in this new time variable.

20.3.2. The η-Lagrangian

The action is invariant under reparametrization of time. Starting from

$$S = \int \left[\frac{1}{2}ma^2\dot{\mathbf{x}}^2 - m\Phi\right] dt$$

we substitute $dt = d\eta/H$ and $\dot{\mathbf{x}} = (d\mathbf{x}/d\eta)(d\eta/dt) = H\mathbf{x}'$ where primes denote $d/d\eta$:

$$S = \int \left[\frac{1}{2}ma^2H^2\mathbf{x}'^2 - m\Phi\right] \frac{d\eta}{H} = \int \left[\frac{1}{2}ma^2H\,\mathbf{x}'^2 - \frac{m\Phi}{H}\right] d\eta$$

Reading off the η-Lagrangian:

$$\tilde{\mathcal{L}} = \frac{1}{2}ma^2H\,\mathbf{x}'^2 - \frac{m\Phi}{H}$$

20.3.3. Conjugate Momentum

The conjugate momentum with respect to η is

$$\mathbf{p} = \frac{\partial\tilde{\mathcal{L}}}{\partial\mathbf{x}'} = ma^2H\,\mathbf{x}'$$

Since $\mathbf{x}' = d\mathbf{x}/d\eta = \dot{\mathbf{x}}/H$, this gives

$$\mathbf{p} = ma^2\dot{\mathbf{x}}$$

This is the same conjugate momentum as in the original (t-time) Lagrangian — it is proportional to the peculiar velocity $\mathbf{v}_{\rm pec} = a\dot{\mathbf{x}}$:

$$\mathbf{p} = ma\,\mathbf{v}_{\rm pec}$$

> **i** Connection to Redshift-Space Distortions
>
> This momentum is convenient for computing redshift-space positions. If $\hat{\mathbf{n}}$ is the line-of-sight direction, the redshift-space coordinate is
>
> $$\mathbf{s} = \mathbf{x} + \hat{\mathbf{n}}\frac{p_\parallel}{ma^2H} = \mathbf{x} + \hat{\mathbf{n}}\frac{v_{\rm pec,\parallel}}{aH}$$
>
> Having $\mathbf{p} = ma^2\dot{\mathbf{x}}$ directly available from the integrator makes this trivial to evaluate.

20.3.4. The Hamiltonian

To construct the Hamiltonian, we perform the Legendre transform:

$$\tilde{H} = \mathbf{p}\cdot\mathbf{x}' - \tilde{\mathcal{L}}$$

First, express $\mathbf{x}'$ in terms of $\mathbf{p}$:

$$\mathbf{x}' = \frac{\mathbf{p}}{ma^2H}$$

Then:

$$\mathbf{p} \cdot \mathbf{x}' = \frac{\mathbf{p}^2}{ma^2 H}$$

$$\tilde{\mathcal{L}} = \frac{1}{2}\frac{\mathbf{p}^2}{ma^2 H} - \frac{m\Phi}{H}$$

Therefore:

$$\tilde{H} = \frac{\mathbf{p}^2}{ma^2 H} - \frac{1}{2}\frac{\mathbf{p}^2}{ma^2 H} + \frac{m\Phi}{H}$$

$$\boxed{\tilde{H} = \underbrace{\frac{\mathbf{p}^2}{2ma^2 H}}_{A(\mathbf{p},\eta)} + \underbrace{\frac{m\Phi(\mathbf{x})}{H}}_{B(\mathbf{x},\eta)}}$$

This is the key result: **the Hamiltonian is separable** into a kinetic term A depending only on $\mathbf{p}$ (and η) and a potential term B depending only on $\mathbf{x}$ (and η). This separability is exactly what we need for the symplectic splitting to work.

> **!** Separability is the Key
>
> The explicit time-dependence through $H(\eta)$ and $a(\eta)$ does not spoil the splitting — what matters is that for a *fixed* value of η, the Hamiltonian separates into a function of $\mathbf{p}$ alone plus a function of $\mathbf{x}$ alone. The time-dependence of the coefficients simply means that the drift and kick factors vary from step to step.

20.3.5. Hamilton's Equations

From $\tilde{H}$, Hamilton's equations in the η variable are:

$$\frac{d\mathbf{x}}{d\eta} = \frac{\partial \tilde{H}}{\partial \mathbf{p}} = \frac{\mathbf{p}}{ma^2 H}$$

$$\frac{d\mathbf{p}}{d\eta} = -\frac{\partial \tilde{H}}{\partial \mathbf{x}} = -\frac{m\nabla\Phi}{H}$$

These equations have no Hubble drag term — the friction that appeared in the coordinate-time formulation has been absorbed into the choice of time variable and conjugate momentum.

20.3.6. Drift and Kick Factors

In the KDK scheme, the drift and kick sub-steps require integrating Hamilton's equations with either $B = 0$ (drift) or $A = 0$ (kick) over an interval in η. Since A and B have explicit η-dependence through a and H, these sub-steps involve integrals.

Drift step (from a_1 to a_2, with $\mathbf{p}$ fixed):

$$\Delta\mathbf{x} = \frac{\mathbf{p}}{m}\int_{a_1}^{a_2} \frac{d\eta}{a^2 H} = \frac{\mathbf{p}}{m}\int_{a_1}^{a_2} \frac{da}{a^3 H(a)}$$

where we used $d\eta = da/a$. We define the **drift factor**:

$$\boxed{D(a_1, a_2) \equiv \int_{a_1}^{a_2} \frac{da}{a^3 H(a)}}$$

Kick step (from a_1 to a_2, with $\mathbf{x}$ fixed):

$$\Delta\mathbf{p} = -m\nabla\Phi \int_{a_1}^{a_2} \frac{d\eta}{H} = -m\nabla\Phi \int_{a_1}^{a_2} \frac{da}{aH(a)}$$

We define the **kick factor**:

$$\boxed{K(a_1, a_2) \equiv \int_{a_1}^{a_2} \frac{da}{aH(a)}}$$

> **i** Why Integrals over a Rather Than η?
>
> We converted from $d\eta$ to da/a for practical reasons: the Hubble parameter $H(a)$ is a known function of a (from the Friedmann equation), making the integrands explicit. In the η variable, one would need H as a function of $\eta = \ln a$, which is the same information in a slightly less convenient form.

20.3.7. Einstein–de Sitter Case

For an Einstein–de Sitter (EdS) universe ($\Omega_m = 1$, $\Omega_\Lambda = 0$), the Friedmann equation gives

$$H(a) = H_0\, a^{-3/2}$$

The drift and kick factors then have closed-form expressions.

Drift factor:

$$D(a_1, a_2) = \int_{a_1}^{a_2} \frac{da}{a^3 \cdot H_0 a^{-3/2}} = \frac{1}{H_0} \int_{a_1}^{a_2} \frac{da}{a^{3/2}} = \frac{1}{H_0} \left[-\frac{2}{\sqrt{a}} \right]_{a_1}^{a_2}$$

$$D_{\text{EdS}}(a_1, a_2) = \frac{2}{H_0} \left(\frac{1}{\sqrt{a_1}} - \frac{1}{\sqrt{a_2}} \right)$$

Kick factor:

$$K(a_1, a_2) = \int_{a_1}^{a_2} \frac{da}{a \cdot H_0 a^{-3/2}} = \frac{1}{H_0} \int_{a_1}^{a_2} a^{1/2}\, da = \frac{1}{H_0} \left[\frac{2}{3} a^{3/2} \right]_{a_1}^{a_2}$$

$$K_{\text{EdS}}(a_1, a_2) = \frac{2}{3H_0} \left(a_2^{3/2} - a_1^{3/2} \right)$$

20.3.8. General ΛCDM

For a general flat ΛCDM cosmology:

$$H(a) = H_0\sqrt{\Omega_m\, a^{-3} + \Omega_\Lambda}$$

The drift and kick integrals become:

$$D(a_1, a_2) = \int_{a_1}^{a_2} \frac{da}{a^3 H_0\sqrt{\Omega_m a^{-3} + \Omega_\Lambda}}$$

$$K(a_1, a_2) = \int_{a_1}^{a_2} \frac{da}{a H_0\sqrt{\Omega_m a^{-3} + \Omega_\Lambda}}$$

These generally do not have closed-form solutions and must be evaluated numerically.

> **i** Production Codes
>
> In practice, production N-body codes precompute $D(a)$ and $K(a)$ as lookup tables at startup (evaluated from some initial a_i to a grid of a values using numerical quadrature). During the simulation, the drift and kick factors for any interval $[a_1, a_2]$ are obtained by table lookup and subtraction: $D(a_1, a_2) = D(a_2) - D(a_1)$.

20.4. Comparison of Formulations

The three natural Lagrangian formulations for cosmological N-body simulations differ in their choice of time variable and conjugate momentum:

	$\mathcal{L}_{\rm pec}$	$\mathcal{L}'$ (total deriv.)	$\tilde{\mathcal{L}}$ ($\ln a$ time)
Time variable	t	t	$\eta = \ln a$
Conjugate momentum	$ma^2\dot{\mathbf{x}}$	$ma\dot{\mathbf{r}}$	$ma^2\dot{\mathbf{x}}$
Physical meaning of p	$\propto \mathbf{v}_{\rm pec}$	$\propto$ total velocity	$\propto \mathbf{v}_{\rm pec}$
Hubble drag in EOM	Yes	No	No
RSD convenience	Direct	Requires subtracting $H\mathbf{x}$	Direct
Hamiltonian separable?	Yes	Yes	Yes
Common use	Textbook derivations	Analytical work	Simulation codes

The $\tilde{\mathcal{L}}$ formulation with $\eta = \ln a$ is preferred for simulations: the absence of Hubble drag simplifies the integrator, the conjugate momentum directly gives the peculiar velocity, and η provides a natural logarithmic time-stepping (equal steps in η correspond to equal expansion factors, giving finer resolution at early times when structures are forming rapidly).

20.5. The KDK Step

Bringing it all together, a single KDK leapfrog step advancing from a_n to a_{n+1} (with midpoint $a_{n+1/2}$) proceeds as:

Step 1 — Half Kick:

$$\mathbf{p}_{n+1/2} = \mathbf{p}_n - m\nabla\Phi(\mathbf{x}_n)\, K(a_n, a_{n+1/2})$$

Step 2 — Full Drift:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{\mathbf{p}_{n+1/2}}{m}\, D(a_n, a_{n+1})$$

Step 3 — Half Kick:

Compute $\nabla\Phi(\mathbf{x}_{n+1})$ from the particle positions at $\mathbf{x}_{n+1}$ (this is where the Poisson solver enters — to be discussed in later sections), then:

$$\mathbf{p}_{n+1} = \mathbf{p}_{n+1/2} - m\nabla\Phi(\mathbf{x}_{n+1})\, K(a_{n+1/2}, a_{n+1})$$

> **!** The Expensive Step
>
> The force evaluation (computing $\nabla\Phi$ from the particle distribution) is by far the most expensive part of each timestep. The KDK ordering means we need exactly **one force evaluation per step** — the force at the end of one step provides the first half-kick of the next step. This is a crucial practical advantage. How we compute this force efficiently using the Particle-Mesh method (CIC interpolation + FFT-based Poisson solver) is the subject of the next sections.

20.6. Setting Up

Before diving into the Particle-Mesh algorithm, let us set up our computational environment. We will use JAX for the simulation code (to be developed in later sections), but for now we only need NumPy and Matplotlib.

20.7. The Particle-Mesh Method: Mass Assignment

The KDK integrator needs the gravitational force $\nabla\Phi$ at every particle position, which requires solving the Poisson equation

$$\nabla^2\Phi = 4\pi G\bar{\rho}a^2\delta$$

The **Particle-Mesh (PM)** method solves this on a regular grid using FFTs, which gives $\mathcal{O}(N_g \log N_g)$ scaling (where N_g is the number of grid cells) rather than the $\mathcal{O}(N^2)$ cost of direct particle-particle summation.

The PM pipeline has three steps:

1. **Mass assignment**: deposit particle masses onto the grid to get $\delta(\mathbf{x}_{\text{grid}})$
2. **Poisson solve**: use FFTs to get $\Phi(\mathbf{x}_{\text{grid}})$ from $\delta(\mathbf{x}_{\text{grid}})$

3. **Force interpolation**: interpolate $\nabla\Phi$ from the grid back to particle positions

Steps 1 and 3 require a scheme for transferring quantities between particles (at arbitrary positions) and grid points (at fixed locations). The choice of scheme controls the accuracy, smoothness, and isotropy of the forces.

20.7.1. Nearest Grid Point (NGP)

The simplest approach: assign each particle's entire mass to the single nearest grid point.

For a 1D grid with spacing h and grid points at $x_j = jh$, a particle at position x_p contributes to the grid as:

$$\rho_j \mathrel{+}= m_p \cdot \begin{cases} 1 & \text{if } |x_p - x_j| < h/2 \\ 0 & \text{otherwise} \end{cases}$$

This is just a zeroth-order (piecewise constant) interpolation. It is fast but produces a very noisy density field — the density jumps discontinuously every time a particle crosses a cell boundary, and these discontinuities propagate into discontinuous forces.

20.7.2. Cloud-in-Cell (CIC)

Cloud-in-Cell is a first-order (piecewise linear) scheme that spreads each particle's mass across the 2^d nearest grid points (where d is the number of dimensions), weighted by the overlap volume.

In 1D, a particle at x_p contributes to the two bracketing grid points x_j and x_{j+1} with weights:

$$W(x_p, x_j) = \begin{cases} 1 - |x_p - x_j|/h & \text{if } |x_p - x_j| < h \\ 0 & \text{otherwise} \end{cases}$$

The weight is just a triangle (tent) function — linearly interpolating between the two nearest grid points. In d dimensions, the weight factorizes:

$$W(\mathbf{x}_p, \mathbf{x}_j) = \prod_{i=1}^{d} W(x_{p,i}, x_{j,i})$$

so in 3D each particle contributes to $2^3 = 8$ grid points.

> **!** CIC for Both Assignment and Interpolation
>
> A crucial requirement: **the same scheme must be used for mass assignment (particles → grid) and force interpolation (grid → particles)**. Using different schemes breaks Newton's third law — a particle would not feel the same force it exerts — leading to self-forces and momentum non-conservation. This is sometimes called the "transpose" requirement: the interpolation operator must be the transpose of the assignment operator.

20.7.3. Visualizing CIC in 1D

To build intuition, let us visualize how CIC works. We place a few particles on a 1D grid and show how each particle's mass is split between neighboring grid points.

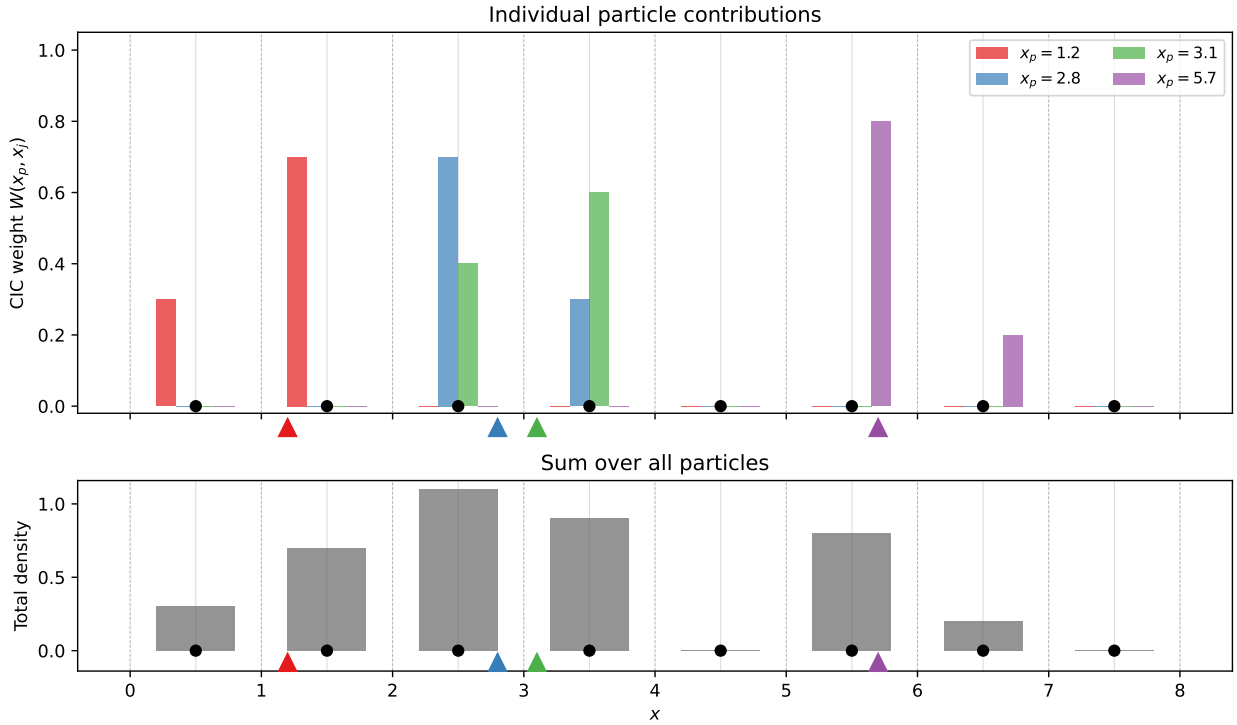


Figure 20.1.: CIC mass assignment in 1D. Particles (red triangles) deposit mass onto grid points (black dots) using linear weights. The blue bars show the weight assigned to each grid point from each particle, and the gray bars show the total deposited density. The triangle/tent function ensures smooth, continuous density as particles move.

20.7.4. Visualizing CIC in 2D

In 2D, each particle deposits mass onto the four corners of its enclosing cell. The weight at each corner is the product of the 1D weights in each direction — geometrically, this is the area of the rectangle diagonally opposite to that corner.

20.7.5. Why CIC?

CIC is the standard choice for PM codes because it strikes a good balance:

Scheme	Order	Grid points per particle	Density continuity	Force continuity
NGP	0	1	Discontinuous	Discontinuous
CIC	1	2^d	Continuous (C^0)	Discontinuous
TSC	2	3^d	C^1	Continuous

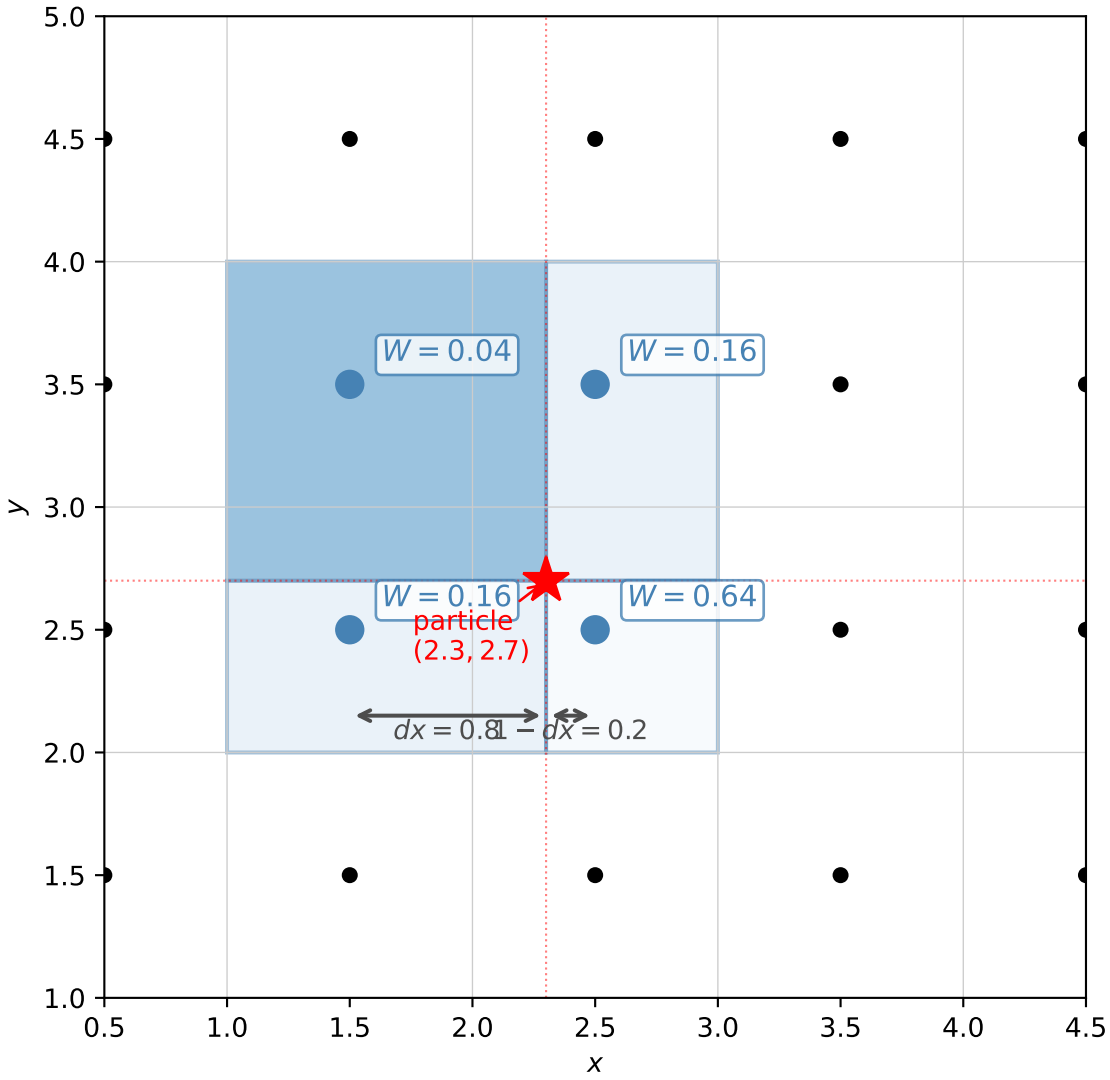


Figure 20.2.: CIC mass assignment in 2D for a single particle (red star). The particle sits inside a grid cell and deposits mass to the four surrounding grid points. Each weight equals the area of the shaded rectangle diagonally opposite to that grid point — this is the product of the 1D tent-function weights in x and y.

Higher-order schemes (TSC = Triangular Shaped Cloud, and beyond) give smoother forces but spread each particle over more grid points, increasing the cost of the assignment and interpolation steps. CIC is almost universally used in production PM and TreePM codes — the force discontinuity is at the grid scale and is subdominant to the grid-scale force resolution limit inherent in the PM method anyway.

> **i** CIC as a Convolution
>
> There is a clean mathematical way to think about the full NGP → CIC → TSC hierarchy. NGP assigns mass using a top-hat window of width h (the "zeroth-order B-spline"). CIC convolves this top-hat with itself, producing the triangle/tent function — the first-order B-spline. TSC convolves once more, producing the second-order B-spline (a piecewise quadratic). Each successive convolution gains one order of smoothness. In Fourier space, convolution becomes multiplication, so the CIC window function is $\tilde{W}(k) = \mathrm{sinc}^2(kh/2)$, compared to $\mathrm{sinc}(kh/2)$ for NGP — the CIC window suppresses small-scale power more aggressively, which is the price of smoother forces.

20.7.6. Force Interpolation

Once we have solved for Φ on the grid (next section), we need the force $\mathbf{F} = -m\nabla\Phi$ at each particle position. As emphasized above, we must use the **same CIC weights** to interpolate the force back to the particles:

$$\mathbf{F}(\mathbf{x}_p) = -m \sum_j W(\mathbf{x}_p, \mathbf{x}_j)\, \nabla\Phi(\mathbf{x}_j)$$

where the sum runs over the 2^d grid points surrounding particle p, and the gradient $\nabla\Phi$ on the grid is computed by finite differences (or, equivalently, by multiplying by $i\mathbf{k}$ in Fourier space before transforming back — more on this in the next section).

This completes the assignment/interpolation infrastructure. We now turn to the middle step of the PM pipeline: solving the Poisson equation on the grid using FFTs.

20.8. The Poisson Solver

20.8.1. The Idea

We need to solve

$$\nabla^2\Phi = 4\pi G\bar{\rho}a^2\delta$$

on a periodic grid. Periodicity is the key simplification: the eigenfunctions of ∇^2 on a periodic domain are plane waves $e^{i\mathbf{k}\cdot\mathbf{x}}$, so the Poisson equation becomes algebraic in Fourier space.

20.8.2. DFT Conventions

Before proceeding, we need to be precise about our Fourier transform conventions. On a grid of N_g points in each dimension with spacing $h = L/N_g$, the discrete Fourier transform (DFT) and its inverse are:

$$\hat{f}_{\mathbf{k}} = \sum_{\mathbf{n}} f_{\mathbf{n}}\, e^{-2\pi i\, \mathbf{k}\cdot\mathbf{n}/N_g}$$

$$f_{\mathbf{n}} = \frac{1}{N_g^d} \sum_{\mathbf{k}} \hat{f}_{\mathbf{k}}\, e^{+2\pi i\, \mathbf{k}\cdot\mathbf{n}/N_g}$$

where $\mathbf{n}$ indexes grid points and $\mathbf{k}$ indexes wavenumber bins. This is the convention used by NumPy's `np.fft.fftn` and `np.fft.ifftn` (and JAX's `jnp.fft.fftn`), with the $1/N_g^d$ normalization on the inverse transform.

20.8.3. From Continuous to Discrete Fourier Transforms

In physics we work with the continuous Fourier transform pair (in 1D for clarity):

$$\hat{f}(k) = \int_{-\infty}^{\infty} f(x)\, e^{-ikx}\, dx \qquad f(x) = \int_{-\infty}^{\infty} \hat{f}(k)\, \frac{dk}{2\pi}\, e^{+ikx}$$

On a periodic box of length L with N_g grid points at positions $x_n = nh$ (where $h = L/N_g$), the integral becomes a Riemann sum:

$$\hat{f}(k) \approx \sum_{n=0}^{N_g-1} f(x_n)\, e^{-ikx_n}\, h$$

The allowed wavenumbers are $k = 2\pi m/L$ for integer m. Substituting $x_n = nh = nL/N_g$ and $k = 2\pi m/L$:

$$\hat{f}(k_m) \approx h \sum_{n=0}^{N_g-1} f_n\, e^{-2\pi i\, mn/N_g} = h\, \hat{f}_m^{\mathrm{DFT}}$$

So the **DFT output differs from the continuous transform by a factor of** h (the grid spacing, which is the quadrature weight of the Riemann sum).

Similarly, for the inverse transform, the integral over $dk/(2\pi)$ becomes a sum over discrete modes with spacing $\Delta k = 2\pi/L$:

$$f(x_n) \approx \sum_m \hat{f}(k_m)\, e^{+ik_m x_n}\, \frac{\Delta k}{2\pi} = \frac{1}{L} \sum_m \hat{f}(k_m)\, e^{+2\pi i\, mn/N_g}$$

Substituting $\hat{f}(k_m) = h\, \hat{f}_m^{\mathrm{DFT}}$:

$$f_n = \frac{h}{L} \sum_m \hat{f}_m^{\mathrm{DFT}}\, e^{+2\pi i\, mn/N_g} = \frac{1}{N_g} \sum_m \hat{f}_m^{\mathrm{DFT}}\, e^{+2\pi i\, mn/N_g}$$

which is exactly the DFT inverse transform convention above.

In d dimensions, the volume element $d^d x \to h^d$ gives:

$$\boxed{\hat{f}(\mathbf{k}) = h^d \, \hat{f}^{\mathrm{DFT}}_{\mathbf{k}}} \qquad \boxed{f(\mathbf{x_n}) = h^{-d} \, f^{\mathrm{IDFT}}_{\mathbf{n}}}$$

where $h = L/N_g$ is the grid spacing. Here $\hat{f}^{\mathrm{DFT}}$ denotes the output of `fftn`, and f^{IDFT} denotes the output of `ifftn` when fed $\hat{f}(\mathbf{k})$ (the continuous transform values). The forward relation says: multiply the `fftn` output by the volume element h^d to get the continuous transform. The inverse says: if you feed continuous Fourier coefficients into `ifftn`, multiply the result by h^{-d} to get the field values. The two factors are inverses of each other, as they must be.

! When Do These Factors Matter?

If your equation is entirely in Fourier space (like the Poisson equation: divide $\hat{\delta}$ by $\hat{L}$ to get $\hat{\Phi}$, then differentiate), the h^d factors cancel between the forward and inverse transforms and you can work directly with the DFT output — no extra factors needed.
The factors matter when you need to relate DFT output to a physical quantity with specific units. For example, the power spectrum:

$$P(k) = |\hat{\delta}(k)|^2 = h^{2d} \, |\hat{\delta}^{\mathrm{DFT}}_{\mathbf{k}}|^2$$

or Parseval's theorem relating a real-space integral to a Fourier-space sum. We will encounter this when setting up initial conditions from a power spectrum.

! Physical vs Grid Wavenumbers

The DFT indices $\mathbf{k}$ run over integers. The corresponding **physical wavenumbers** are

$$\mathbf{k}_{\mathrm{phys}} = \frac{2\pi \mathbf{k}}{L}$$

The Nyquist frequency is $k_{\mathrm{Ny}} = \pi N_g / L = \pi / h$. Modes above this frequency are aliased. In NumPy, after `np.fft.fftn`, the wavenumber indices are ordered as $k = 0, 1, \ldots, N_g/2, -N_g/2+1, \ldots, -1$; use `np.fft.fftfreq(Ng, d=h)` to get the physical frequencies directly.

20.8.4. Solving Poisson in Fourier Space

Taking the DFT of the Poisson equation:

$$\widehat{\nabla^2 \Phi}_{\mathbf{k}} = (4\pi G \bar{\rho} a^2) \, \hat{\delta}_{\mathbf{k}}$$

The left-hand side requires the Fourier-space representation of ∇^2. On a continuous domain, $\nabla^2 e^{i\mathbf{k}\cdot\mathbf{x}} = -k^2 e^{i\mathbf{k}\cdot\mathbf{x}}$, so one might use $-k^2$. However, on a discrete grid we should use the Laplacian that is **consistent with the finite-difference stencil** we use for the gradient. This matters for maintaining consistency between the potential and the force.

Continuous (spectral) Laplacian:

$$\hat{L}(\mathbf{k}) = -k_{\text{phys}}^2 = -\left(\frac{2\pi}{L}\right)^2 |\mathbf{k}|^2$$

Discrete (finite-difference) Laplacian:

For a second-order centered finite difference, $\nabla^2 f \approx (f_{j+1} - 2f_j + f_{j-1})/h^2$ in each dimension. The Fourier-space representation of this operator is:

$$\hat{L}(\mathbf{k}) = -\sum_{i=1}^{d} \frac{4}{h^2} \sin^2\left(\frac{\pi k_i}{N_g}\right)$$

At low k (long wavelengths), $\sin(\pi k/N_g) \approx \pi k/N_g$ and the two agree. They differ near the Nyquist frequency, where the discrete version correctly captures the grid-scale behavior.

> **i** Which Laplacian to Use?
>
> Either choice works. The discrete Laplacian is more self-consistent if you also compute gradients by finite differences in real space. The continuous (spectral) Laplacian is simpler and gives slightly better long-wavelength accuracy. Many production PM codes use the discrete version for strict consistency; for a pedagogical code the difference is minor. We will use the **discrete Laplacian** for consistency.

The solution is then:

$$\hat{\Phi}_{\mathbf{k}} = \frac{4\pi G\bar{\rho}a^2}{\hat{L}(\mathbf{k})}\,\hat{\delta}_{\mathbf{k}} \qquad (\mathbf{k} \neq 0)$$

The $\mathbf{k} = 0$ mode is the mean potential, which is unphysical in a periodic box (it corresponds to the force from a uniform infinite sheet). We set $\hat{\Phi}_{\mathbf{k}=0} = 0$.

20.8.5. Computing the Force

We need $\nabla\Phi$ on the grid to kick the particles. There are two approaches:

Option A: Finite differences in real space. Transform $\hat{\Phi}_{\mathbf{k}}$ back to real space to get $\Phi(\mathbf{x}_j)$, then compute the gradient with centered differences:

$$(\nabla_i \Phi)_j = \frac{\Phi_{j+1} - \Phi_{j-1}}{2h}$$

This requires one inverse FFT (to get Φ) plus a simple stencil operation.

Option B: Differentiation in Fourier space. Multiply $\hat{\Phi}_{\mathbf{k}}$ by the Fourier-space gradient operator, then inverse-transform each component separately:

$$\widehat{(\nabla_i \Phi)}_{\mathbf{k}} = \hat{D}_i(\mathbf{k})\,\hat{\Phi}_{\mathbf{k}}$$

where $\hat{D}_i$ is the Fourier representation of the gradient in direction i.

For consistency with the discrete Laplacian, we use the **discrete gradient**:

$$\hat{D}_i(k_i) = \frac{i}{h} \sin\left(\frac{2\pi k_i}{N_g}\right)$$

which is the Fourier transform of the centered-difference operator $(f_{j+1} - f_{j-1})/(2h)$.

i Combining the Solve and Gradient

In practice, we can combine the Poisson solve and the gradient into a single Fourier-space operation. Define the **Green's function with gradient**:

$$\hat{G}_i(\mathbf{k}) = \frac{\hat{D}_i(\mathbf{k})}{\hat{L}(\mathbf{k})}$$

Then the force field in direction i is:

$$\widehat{F_i}(\mathbf{k}) = -m \cdot (4\pi G\bar{\rho}a^2)\, \hat{G}_i(\mathbf{k})\, \hat{\delta}_{\mathbf{k}}$$

This requires one forward FFT (to get $\hat{\delta}$) and d inverse FFTs (one per force component) — no intermediate real-space potential is needed.

20.8.6. The Full PM Force Pipeline

Putting it all together, the force calculation at each timestep proceeds as:

1. **CIC deposit**: assign particle masses to grid $\to \rho(\mathbf{x}_j)$
2. **Compute overdensity**: $\delta_j = \rho_j/\bar{\rho} - 1$
3. **Forward FFT**: $\delta_j \to \hat{\delta}_{\mathbf{k}}$
4. **Green's function multiply**: $\hat{F}_{i,\mathbf{k}} = -m \cdot (4\pi G\bar{\rho}a^2)\, \hat{G}_i(\mathbf{k})\, \hat{\delta}_{\mathbf{k}}$ for each direction i
5. **Inverse FFT**: $\hat{F}_{i,\mathbf{k}} \to F_i(\mathbf{x}_j)$ for each direction
6. **CIC interpolate**: interpolate $F_i(\mathbf{x}_j)$ to particle positions

Steps 3–5 cost $\mathcal{O}(N_g^d \log N_g)$ via the FFT. Steps 1 and 6 cost $\mathcal{O}(N)$ where N is the number of particles. The total cost per timestep is $\mathcal{O}(N_g^d \log N_g + N)$.

! PM Force Resolution

The PM method resolves forces only down to the grid scale $\sim h = L/N_g$. Structures smaller than $\sim 2h$ are smoothed out. This is adequate for following the large-scale evolution of the density field, but not for resolving internal halo structure. Production codes address this with hybrid methods:

- **P^{3}M**: PM for long-range forces + direct particle-particle (PP) for short-range corrections
- **TreePM**: PM for long-range + tree algorithm for short-range
- **AMR-PM**: Adaptive mesh refinement in high-density regions

For our purposes, a pure PM code is sufficient — we will test it against the Zel'dovich approximation, which operates well above the grid scale.

20.9. Implementation

We now implement the PM N-body code in JAX. We will work in 3D throughout, in a periodic box of comoving side L with N_g^3 grid cells and N_p^3 particles.

20.9.1. Code Units

Before writing any code, let us set up a unit system that simplifies the equations. We make three choices:

Quantity	Code value	Physical value
Hubble constant H_0	1	H_0
Box size L	1	L_{phys} (comoving Mpc/h)
Particle mass m	1	$m_p = \Omega_m \bar{\rho} L^3/N$

With $H_0 = 1$, time is measured in units of $1/H_0$ and the Friedmann equation becomes $H(a) = a^{-3/2}$ (for EdS). With $L = 1$, comoving positions run from 0 to 1 and wavenumbers are $k = 2\pi n$ for integer n. With $m = 1$, the density deposited onto the grid is simply a particle count per cell.

The key derived relation: $4\pi G\bar{\rho} = \frac{3}{2}\Omega_m H_0^2$, so the Poisson source $4\pi G\bar{\rho}a^2\delta$ becomes $\frac{3}{2}\Omega_m \delta/a$ in code units.

Converting to physical units. A code velocity v_{code} (which has dimensions of $L/t_{\text{code}} = L \cdot H_0$) maps to a physical velocity as:

$$v_{\text{phys}} = v_{\text{code}} \times H_0 L_{\text{phys}}$$

Similarly, code positions map as $x_{\text{phys}} = x_{\text{code}} \times L_{\text{phys}}$, and code times as $t_{\text{phys}} = t_{\text{code}}/H_0$. The conjugate momentum $p = a^2\dot{x}$ in code units becomes $p_{\text{phys}} = p_{\text{code}} \times m_p H_0 L_{\text{phys}}$.

> **i** Why These Units?
>
> This is the standard approach in N-body codes: factor out the overall scales (H_0, L, m_p) so the code only deals with dimensionless numbers of order unity. The physics is entirely in the cosmological functions ($H(a)$, $D(a_1, a_2)$, $K(a_1, a_2)$) and the Poisson prefactor $\frac{3}{2}\Omega_m/a$. To run a "different" simulation (different box size, particle number, or H_0), you only change the conversion factors at output time — the code itself is identical.

20.9.2. Periodic Boundary Conditions

As discussed in the previous lecture, we simulate a cubic box of comoving side L with **periodic boundary conditions**: a particle leaving one face re-enters from the opposite face, and every particle sees an infinite periodic tiling of the box in all directions.

This is both a physical assumption and a computational convenience:

- **Physical**: on scales much smaller than the horizon, the universe is (statistically) homogeneous. A periodic box is the natural way to enforce this — there are no boundaries, no edge effects, and the mean density is exactly $\bar{\rho}$ by construction.
- **Computational**: periodicity means the density field is a periodic function, so we can use the **discrete Fourier transform** (FFT) to solve the Poisson equation. This is what makes the PM method fast.

Periodicity also means we can only resolve modes with wavelengths $\lambda \leq L$, i.e., wavenumbers $k \geq 2\pi/L$. There is no information about fluctuations on scales larger than the box — these are implicitly assumed to vanish. This sets a minimum wavenumber ("fundamental mode") $k_f = 2\pi/L$.

In the code, periodicity enters in two places: (1) particle positions are wrapped via `pos % L` after every drift step, and (2) the CIC stencil uses modular arithmetic `(idx + 1) % Ng` so that particles near the boundary correctly deposit mass across the periodic boundary.

20.9.3. Imports and Setup

20.9.4. Cosmological Functions (Einstein–de Sitter)

For an EdS universe ($\Omega_m = 1$), the Hubble parameter and drift/kick factors have simple closed-form expressions (derived in Section 1). Using these avoids any numerical integration.

> **i** Generalizing Beyond EdS
>
> For a general ΛCDM cosmology, replace these three functions with numerical quadrature (e.g., `jnp.trapezoid` over a fine grid in a) or precomputed lookup tables. The rest of the code is unchanged — this is one of the advantages of the $\ln a$ formulation.

20.9.5. CIC Mass Assignment

We now implement the CIC deposit. For each particle, we find the two nearest grid points in each dimension and distribute mass with linear weights.

The key subtlety is **periodicity**: particles near the box boundary wrap around, so their CIC stencil may span the periodic boundary. We handle this with modular arithmetic on the grid indices.

> **i** CIC and JAX Tracing
>
> The triple loop over the 8 CIC corners is unrolled at trace time by JAX — it produces 8 `.at[].add()` operations in the computation graph. This is fine because 8 is small and fixed. The actual particle loop is vectorized (all N particles are processed simultaneously via array operations). When we `jit` the `pm_force` function below, these CIC functions get compiled as part of the larger computation graph.

20.9.6. CIC Force Interpolation

The transpose operation: interpolate a grid-based field to particle positions using the same CIC weights.

20.9.7. Green's Function

We precompute the Fourier-space Green's function that combines the Poisson solve and gradient in one step, as described in Section 3. This only needs to be computed once for a given grid.

20.9.8. PM Force Calculation

Now we assemble the full force pipeline: CIC deposit → FFT → Green's function → IFFT → CIC interpolate.

Units Check

Let us verify the units. The Poisson equation in our $H_0 = 1$ units is:

$$\hat{L}(\mathbf{k})\hat{\Phi} = \frac{3}{2}\Omega_m \frac{1}{a}\hat{\delta}$$

The Green's function $\hat{G}_i = \hat{D}_i/\hat{L}$ has dimensions of $1/[\text{length}]$ (since $\hat{D}_i \sim 1/h$ and $\hat{L} \sim 1/h^2$). So $\nabla\Phi = (3\Omega_m/2a) \cdot G_i * \delta$ has dimensions of $H_0^2 \times \text{length}$, which is an acceleration — correct for the kick equation $\Delta p = -m\nabla\Phi \cdot K(a_1, a_2)$ where K has dimensions of time.

20.9.9. The KDK Integrator

Finally, we assemble the KDK leapfrog step. Each step advances the system from a_n to a_{n+1}.

Force Evaluation Timing

Notice that the force in the first half-kick is evaluated at a_{start} and the force in the second half-kick at a_{end}. Strictly, the kick integral $K(a_1, a_2)$ assumes the force is constant over that interval, so the "correct" scale factor to evaluate at is somewhere in between. Using the endpoint values is the standard approximation — the error is absorbed into the $\mathcal{O}(\Delta a^2)$ truncation error of the leapfrog scheme.
In a more careful implementation, the second half-kick of step n and the first half-kick of step $n+1$ share the same force evaluation (since both use the force at a_{n+1}). We have not optimized for this here, computing the force twice. We will fix this in the simulation loop below.

20.9.10. The Simulation Loop

We now write the main simulation loop that advances the system from an initial scale factor a_i to a final scale factor a_f over N_{steps} equal steps in $\ln a$.

> **!** JIT-Compiled Simulation Loop
>
> The simulation is compiled end-to-end using `@jit` and `jax.lax.fori_loop`. The Python `for` loop is replaced by XLA's loop primitive, which avoids Python overhead and enables hardware acceleration.
> The merged-kick optimization is implemented by precomputing the kick factors: for step i, the merged kick uses $K(a_{i-1/2}, a_{i+1/2})$, except the final step which closes with $K(a_{N-1/2}, a_N)$. This halves the number of force evaluations compared to a naive KDK implementation.
> The `n_steps` and `Ng` arguments are marked `static` — changing them triggers recompilation, but within a run the loop executes as a single compiled computation. The first call incurs a compilation cost; subsequent calls with the same `n_steps` and `Ng` reuse the compiled code.

20.9.11. Utility: Uniform Particle Grid

20.10. Test: Single Sine Wave vs Zel'dovich

We now test our PM code against the **Zel'dovich approximation** — an exact solution to the equations of motion in the limit of small perturbations. This is the standard first test for any N-body code.

20.10.1. The Zel'dovich Approximation

The Zel'dovich approximation describes the displacement of particles from a uniform grid under a growing-mode perturbation. In Lagrangian coordinates $\mathbf{q}$ (the initial grid positions), the comoving position at time a is:

$$\mathbf{x}(\mathbf{q}, a) = \mathbf{q} + D_+(a)\, \Psi(\mathbf{q})$$

where $D_+(a)$ is the linear growth factor and $\Psi(\mathbf{q})$ is the displacement field, related to the initial overdensity by $\nabla \cdot \Psi = -\delta_0(\mathbf{q})$.

The velocity (conjugate momentum) follows from differentiating:

$$\mathbf{p} = ma^2 \dot{\mathbf{x}} = ma^2 \dot{D}_+(a)\, \Psi(\mathbf{q})$$

where $\dot{D}_+ = dD_+/dt$.

For an Einstein–de Sitter universe, the growing mode is simply $D_+(a) = a$, and using $H = H_0 a^{-3/2}$ we get $\dot{D}_+ = \dot{a} = Ha = H_0 a^{-1/2}$, so:

$$\mathbf{p} = ma^2 H_0 a^{-1/2} \Psi = m H_0 a^{3/2} \Psi$$

In our code units ($H_0 = 1$, $m = 1$):

$$\mathbf{x}(\mathbf{q}, a) = \mathbf{q} + a\, \Psi(\mathbf{q}) \qquad \mathbf{p}(\mathbf{q}, a) = a^{3/2}\, \Psi(\mathbf{q})$$

20.10.2. Single Plane-Wave Perturbation

The simplest test case is a single sinusoidal perturbation along one axis. Take the displacement field:

$$\Psi(\mathbf{q}) = -\frac{A}{k}\sin(kq_x)\,\hat{\mathbf{x}}$$

where $k = 2\pi n/L$ for some integer mode number n, and A is the amplitude of the initial overdensity:

$$\delta_0(\mathbf{q}) = -\nabla\cdot\Psi = A\cos(kq_x)$$

The Zel'dovich solution is exact (for this 1D perturbation) until **shell crossing** — the moment when particle trajectories intersect. To see when this happens, consider the Jacobian of the Lagrangian-to-Eulerian map:

$$\frac{\partial x}{\partial q_x} = 1 + a\frac{d\Psi_x}{dq_x} = 1 - aA\cos(kq_x)$$

Mass conservation requires $\bar{\rho}\,dq = \rho\,dx$ (the mass in a Lagrangian volume element is conserved), so the Eulerian density is

$$\rho = \frac{\bar{\rho}}{|\partial x/\partial q_x|}$$

When $\partial x/\partial q_x \to 0$, the density diverges — neighboring particles that started at q and $q + dq$ have been compressed to the same Eulerian position x. Their trajectories cross, forming a **caustic**. After this point, multiple particle streams overlap at the same location and the single-valued Zel'dovich map breaks down.

The first crossing occurs when $\cos(kq_x) = 1$ and $aA = 1$, i.e., at $a_{\text{cross}} = 1/A$. We will choose A and the final scale factor such that we stay well before shell crossing.

> **i** Why This Is an Exact Test
>
> For a single Fourier mode along $\hat{x}$, the Zel'dovich approximation is actually the **exact** solution to the full nonlinear equations of motion (before shell crossing), not just a linear approximation. The physical picture is illuminating. A perturbation purely along $\hat{x}$ means particles only move in x — they form infinite 2D sheets (uniform in y and z) that slide back and forth along $\hat{x}$. The gravitational force on a sheet depends only on how many sheets are to its left vs. right (just like parallel plates in electrostatics). In a periodic box, this count is determined purely by the ordering of the sheets, which is preserved until shell crossing. So as long as no sheets pass through each other, each sheet feels a force that depends only on its Lagrangian label q_x — not on the detailed positions of other sheets. The sheets evolve independently, and the Zel'dovich solution describes their motion exactly.
> Mathematically: the Poisson equation is linear, so the force from a sinusoidal δ is sinusoidal, and the resulting motion preserves the sinusoidal structure. Any disagreement between the simulation and the Zel'dovich prediction (before shell crossing) is a bug.

20.10.3. Setting Up the Test

```
Sanity check: 64^3 = 262144 particles on 64^3 grid
  Density: min=1.0000, max=1.0000 (expected 1.0000)

Mode: n=2, k=12.5664
Amplitude: A=0.5
Shell crossing at a = 2.0
Running from a=0.1 to a=1.5 (100 steps)
Final displacement amplitude: a*A/k = 0.0597 (in units of L)
```

20.10.4. Zel'dovich Initial Conditions

20.10.5. Zel'dovich Prediction at Final Time

20.10.6. Running the Simulation

20.10.7. Time Evolution

Before comparing in detail at the final time, let us visualize how the displacement evolves. We run to several intermediate scale factors and compare with the Zel'dovich prediction at each epoch.

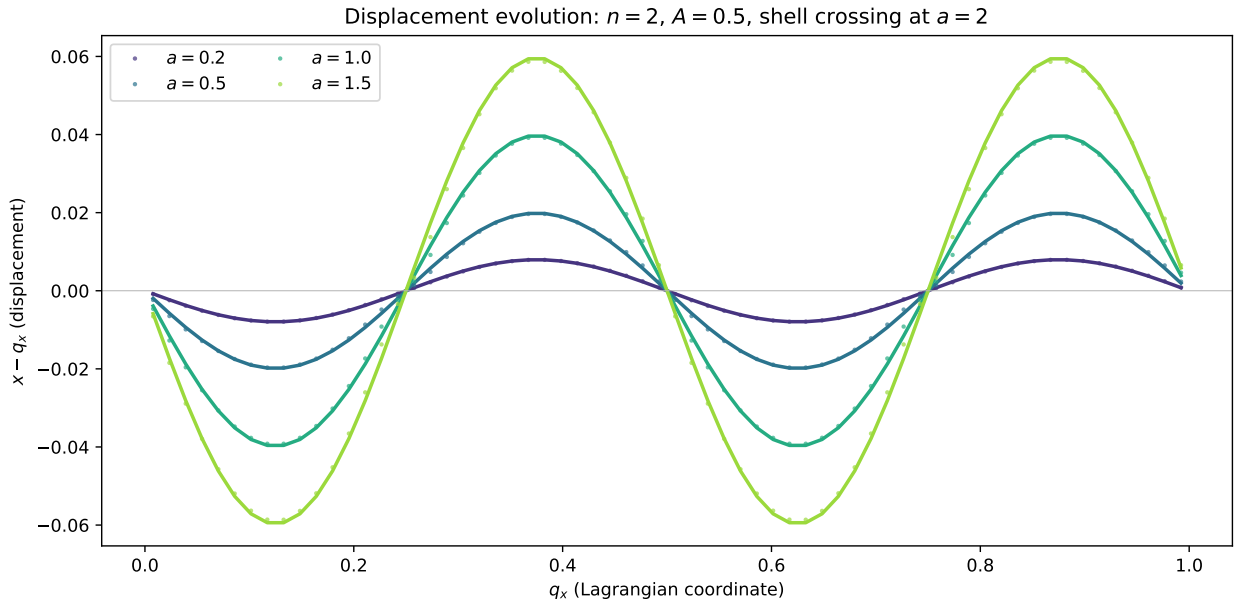


Figure 20.3.: Time evolution of the x-displacement for a single sine-wave perturbation. Solid lines show the exact Zel'dovich solution; dots show the PM simulation. The displacement grows linearly with a (since $D_+ = a$ in EdS) and the sinusoidal shape steepens as the system approaches shell crossing at $a = 1/A = 2$.

20.10.8. Comparing Results at Final Time

We compare the simulation output against the Zel'dovich prediction. Since the perturbation is along x only, we look at the x-displacement and x-momentum as functions of the Lagrangian coordinate q_x.

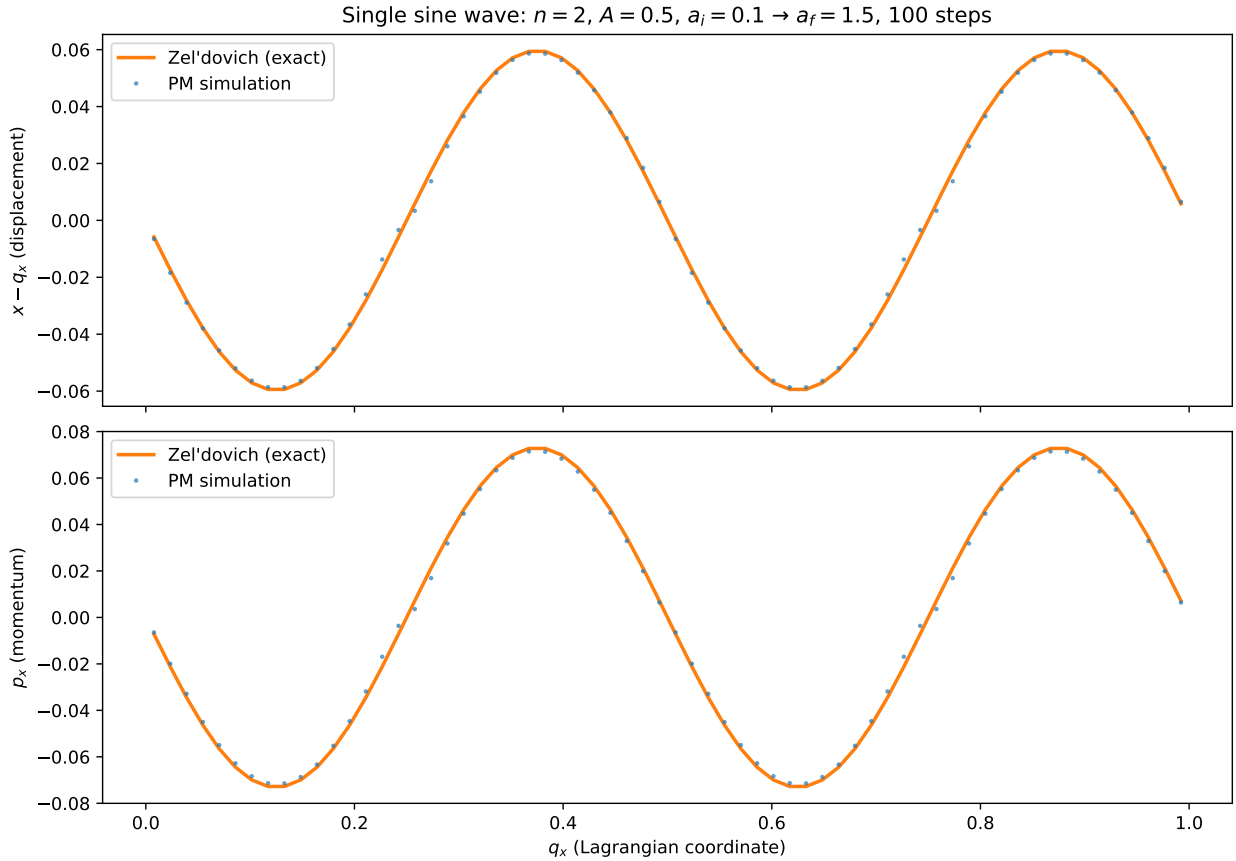


Figure 20.4.: Comparison of the PM simulation (blue dots) against the exact Zel'dovich solution (orange line) for a single sine-wave perturbation. Top: x-displacement from the grid. Bottom: x-momentum. The agreement validates the CIC deposit, Poisson solver, force interpolation, and KDK integrator.

```
Displacement: max |error| = 3.58e-03
Momentum:     max |error| = 4.30e-03
```

The errors are at the $\sim 10^{-3}$ level in both displacement and momentum. These errors have two sources: (1) time-stepping error from the KDK integrator, and (2) spatial error from the PM force calculation (CIC window function, discrete grid). We disentangle these below.

20.10.9. Convergence with Timestep

Let us verify the second-order convergence explicitly. A subtlety: if we compare against the exact Zel'dovich solution, the error is dominated by the **spatial** PM error (CIC window, discrete particle

Figure 20.5.: Absolute error in displacement and momentum relative to the exact Zel'dovich solution. The errors combine both time-stepping error and spatial PM error (CIC window, discrete grid). See the convergence tests below to disentangle these contributions.

sampling) rather than the **temporal** error from the integrator. These spatial errors are independent of timestep count, creating a floor.

To isolate the time-stepping error, we compare each run against a high-resolution reference run (1000 steps) using the **same PM code**. The spatial errors cancel in the difference, leaving only the integrator error.

20.10.10. Convergence with Grid Resolution

The timestep convergence test above isolates the integrator error by comparing against a reference run with the **same spatial resolution**. But how large is the spatial PM error itself? We can test this by running the same problem at increasing grid resolution $N_g = N_p$ (keeping 1 particle per cell and using enough timesteps that the time-stepping error is negligible) and comparing against the exact Zel'dovich solution.

```
Ng=Np= 32: max |Δx| = 7.50e-03, max |Δp| = 9.23e-03
Ng=Np= 64: max |Δx| = 3.58e-03, max |Δp| = 4.30e-03
Ng=Np=128: max |Δx| = 1.77e-03, max |Δp| = 2.11e-03
```

The spatial error decreases as $\sim N_g^{-1}$, not N_g^{-2}. This may seem surprising since the discrete gradient and Laplacian operators are both second-order accurate. The bottleneck is **CIC itself**: CIC is a first-order (piecewise linear) interpolation scheme, producing a density field that is C^0 but not C^1. The resulting force — the gradient of the potential — is **discontinuous** at cell boundaries, with $\mathcal{O}(h)$ jumps. This limits the maximum force error at particle positions to $\mathcal{O}(h) = \mathcal{O}(N_g^{-1})$, regardless of how accurate the Poisson solver is.

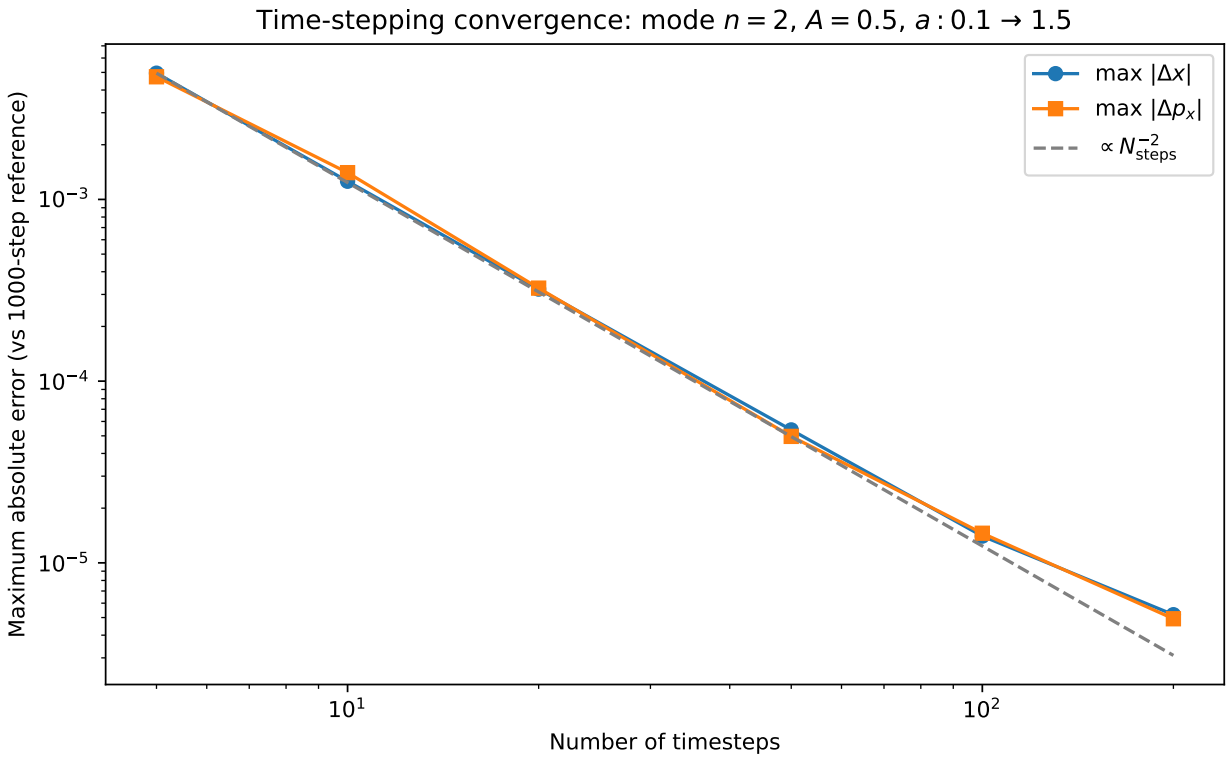


Figure 20.6.: Convergence of the KDK leapfrog integrator. Errors are measured relative to a high-resolution reference run (1000 steps) to isolate the time-stepping error from the spatial PM error. Both displacement and momentum errors decrease as N_{steps}^{-2}, confirming second-order convergence.

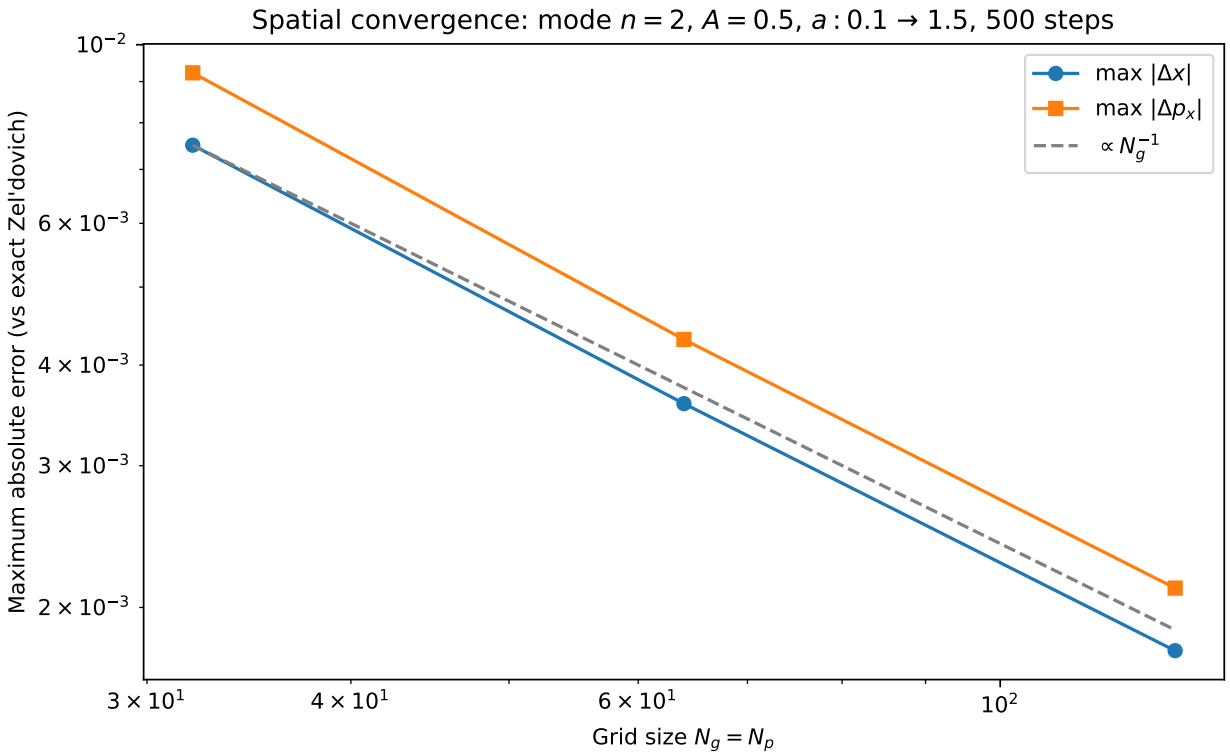


Figure 20.7.: Spatial convergence of the PM force. Each run uses 500 timesteps (so time-stepping error is negligible) at increasing grid resolution. The error against the exact Zel'dovich solution decreases with resolution, demonstrating that the PM spatial error is the dominant contribution to the total error at coarse grids.

Higher-order mass assignment schemes improve this: TSC (second-order B-spline) gives continuous forces and $\mathcal{O}(N_g^{-2})$ spatial convergence.

20.11. Cosmological Initial Conditions

We now put the pieces together and run a cosmological simulation with realistic initial conditions drawn from the matter power spectrum.

20.11.1. The Recipe

To generate cosmological ICs we need:

1. **A matter power spectrum** $P(k)$ at the initial redshift — we use CLASS to compute this.
2. **A Gaussian random field** $\delta(\mathbf{k})$ on the grid, with variance set by $P(k)$: each Fourier mode is drawn as
$$\hat{\delta}(\mathbf{k}) = \sqrt{P(k)/V}\,(g_1 + i\,g_2)/\sqrt{2}$$
where g_1, g_2 are independent standard normals and $V = L^3$ is the box volume. The factor $1/V$ converts the continuous power spectrum (dimensions of volume) to the variance of the discrete Fourier mode.
3. **Zel'dovich displacements** from the density field: $\hat{\Psi}_i(\mathbf{k}) = -ik_i/k^2\,\hat{\delta}(\mathbf{k})$, then $\mathbf{x} = \mathbf{q} + D_+(a_{\rm init})\Psi$ and $\mathbf{p} = a_{\rm init}^{3/2}\Psi$ (EdS).

20.11.2. Power Spectrum from CLASS

```
CLASS cosmology: Omega_m = 0.2906, h = 0.700, sigma8 = 0.8300
```

> **i** EdS Approximation
>
> Our PM code uses Einstein–de Sitter dynamics ($\Omega_m = 1$). We use the CLASS power spectrum for the **shape** of $P(k)$ (which encodes the transfer function, BAO, etc.) but evolve under EdS gravity. This is a common simplification in pedagogical codes — the large-scale structure looks qualitatively correct, but the growth rate and normalization differ from ΛCDM. A production code would use the correct $H(a)$ and growth factor for the target cosmology.

20.11.3. Generating Gaussian Initial Conditions

```
Box: L = 200.0 Mpc/h, Ng = Np = 128
Starting at z = 49.0 (a = 0.0200)
Fundamental mode: k_f = 0.0314 h/Mpc
Nyquist: k_Ny = 2.01 h/Mpc

Particles: 2,097,152
Initial rms displacement: 0.1098 Mpc/h
```

20.11.4. Running the Cosmological Simulation

```
Running 128^3 simulation: a = 0.0200 → 1.0
  100 steps, Ng = 128, L = 200.0 Mpc/h
Done!
```

20.11.5. The Cosmic Web

We visualize the result by depositing particles onto the grid and showing a thin slice through the density field.

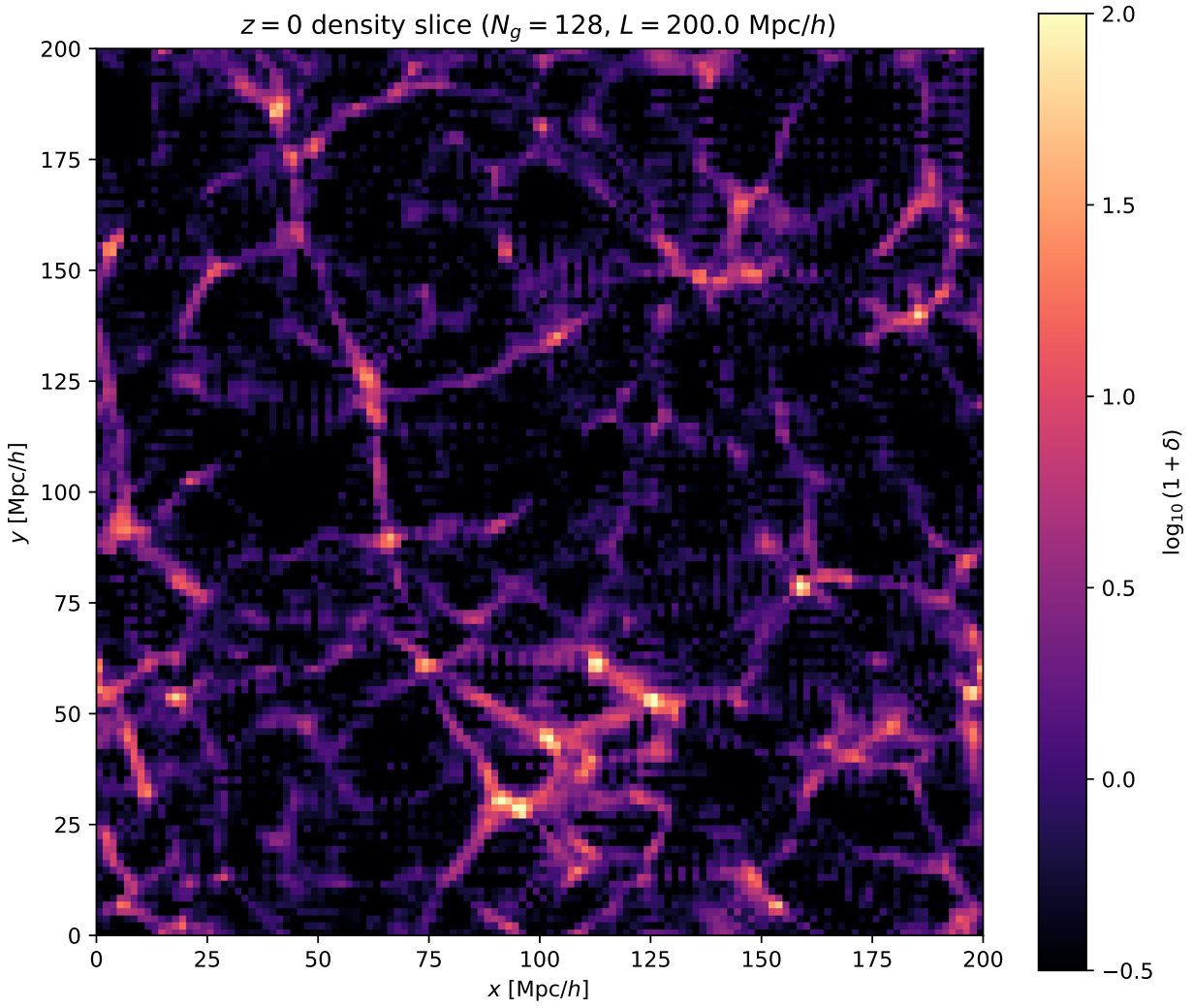


Figure 20.8.: Slice through the evolved density field at $z = 0$ from our 128^3 PM simulation with cosmological initial conditions. The familiar features of the cosmic web — filaments, nodes, and voids — are clearly visible. The color scale shows $\log_{10}(1 + \delta)$.

```
Density field statistics at z=0:
```

```
min(delta) = -1.00
max(delta) = 494.1
rms(delta) = 3.68
```

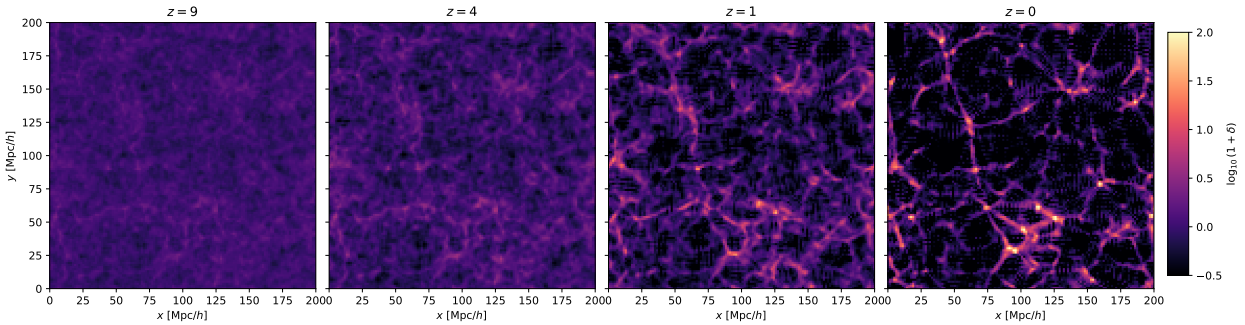


Figure 20.9.: Evolution of the cosmic web from $z = 9$ to $z = 0$. Structure grows hierarchically: small-scale perturbations collapse first, then larger structures assemble through merging and accretion along filaments.

20.11.6. Animation

Finally, let us watch the cosmic web form in real time. We run the simulation to 50 intermediate epochs and assemble the density slices into a movie.

```
<IPython.core.display.Video object>
```

Figure 20.10.: Animation of the cosmic web forming from $z = 49$ to $z = 0$. The density field grows from nearly uniform initial conditions through gravitational instability alone — no hydrodynamics, no star formation, just dark matter and gravity on a 128^3 PM grid.

This is the cosmic web emerging from nearly uniform initial conditions through gravitational instability alone — no hydrodynamics, no star formation, just dark matter and gravity computed on a 128^3 grid with the PM algorithm we built from scratch in this lecture.

21. Cosmological Simulations III: Analysis

21.1. Introduction

In the previous lecture, we built a Particle-Mesh (PM) N-body code from scratch — implementing CIC mass assignment, an FFT-based Poisson solver, and a symplectic KDK leapfrog integrator. We tested it against the Zel'dovich approximation and ran a 128^3 cosmological simulation to see the cosmic web emerge.

In this lecture, we push the resolution to 256^3 and analyze the simulation output in detail: density fields, power spectra, growth factors, redshift-space distortions, and RSD multipoles. Along the way, we will connect the simulation results back to the linear theory we developed earlier in the course.

21.2. Setup

21.2.1. PM Code

All the PM functions from the previous lecture are collected below. This is the same code — `cic_deposit`, `cic_interp`, `make_green_function`, `pm_force`, `kdk_step`, `run_simulation`, and supporting utilities — copied verbatim so that this notebook is self-contained.

21.2.2. Imports and Parameters

```
Grid: 256^3 = 16,777,216 cells
Particles: 256^3 = 16,777,216
Box: L = 500.0 Mpc/h
Redshift: z = 49.0 → 0  (a = 0.0200 → 1.0)
Steps: 100 (log-spaced in a)
```

> **Choice of Timestep Count**
>
> We use 100 steps for this simulation. The KDK leapfrog is second-order, and with steps equally spaced in $\ln a$ the resolution is finest at early times when structures are forming most rapidly. For our EdS cosmology and the observables we measure in this lecture (power spectra, growth factors, RSD multipoles), we have verified that 100 steps gives sub-percent agreement with 200 steps across all scales.
> This should not be taken as a general rule. The required number of steps depends on the cosmology, the box size, the force resolution, and the observables of interest. Any production simulation requires its own convergence tests — the right approach is to run at two different

step counts and verify that your results are insensitive to the choice.

21.2.3. Power Spectrum from CLASS

We use CLASS to compute the linear power spectrum $P(k)$ at $z = 0$, which serves as input for the initial conditions and as a reference for comparison later.

```
CLASS cosmology: Omega_m = 0.2906, h = 0.700, sigma8 = 0.8300
Box: L = 500.0 Mpc/h
Fundamental mode: k_f = 0.0126 h/Mpc
Nyquist: k_Ny = 1.61 h/Mpc
```

21.2.4. Generating Initial Conditions

We generate Gaussian random initial conditions using the Zel'dovich approximation, exactly as in Lecture 20. The procedure: draw a white noise field, color it with $\sqrt{P(k)}$ to create the $z = 0$ density field δ_0, compute the displacement field Ψ via $\Psi_i(\mathbf{k}) = -ik_i/k^2\,\hat{\delta}_0(\mathbf{k})$, then displace particles from a uniform grid by $D_+(a_{\rm init})\,\Psi$.

21.2.5. Running the Simulation

We precompute the Green's function, then run the simulation from $z = 49$ to $z = 0$. We also run to several intermediate redshifts ($z = 9, 4, 1$) so that we can study the growth of structure over time in later sections. Each run uses a proportional number of timesteps, keeping the step size in $\ln a$ consistent.

Results (including the initial conditions and the $z = 0$ linear density field) are cached to disk as separate files so that subsequent renders skip the simulation entirely, and individual snapshots can be regenerated without invalidating the rest.

```
Loading cached ICs from /Users/npadmana/data/teaching/ASTR6000-SP26/021-pm-
analysis/ic.npz
Particles: 16,777,216
  z=9: loaded from cache
  z=4: loaded from cache
  z=1: loaded from cache
  z=0: loaded from cache

All snapshots loaded from cache.
```

Compilation Overhead

The first simulation triggers JAX's JIT compilation — tracing the entire computation graph and compiling it to optimized machine code via XLA. This is a one-time cost; subsequent runs

with the same `n_steps` and `Ng` reuse the cached compilation and execute much faster. Runs with different step counts trigger recompilation (since `n_steps` is a static argument), but each compilation is reused for any future call with the same count.

21.2.6. Analysis Utilities

We will repeatedly need to deposit particles onto the grid and measure power spectra throughout this lecture. Here we set up JIT-compiled and vectorized versions of these operations.

21.3. Linear vs Nonlinear Density Fields

Our simulation evolved 256^3 particles under full nonlinear gravity from $z = 49$ to $z = 0$. How does the result compare to what linear theory predicts?

The initial conditions were generated from the $z = 0$ linear density field $\delta_0(\mathbf{x})$, scaled back to $z = 49$ by multiplying by $D_+(a_{\rm init}) = a_{\rm init}$ (EdS). To get the linear prediction at $z = 0$, we simply take δ_0 — since $D_+(a = 1) = 1$ in our normalization. For the nonlinear field, we deposit the simulation particles at $z = 0$ onto the grid using CIC.

We use $\mathrm{arcsinh}(\delta/2)$ as the color mapping for all density field plots. Unlike $\log_{10}(1+\delta)$, this handles voids ($\delta < 0$) smoothly and behaves like $\log(\delta)$ at high density.

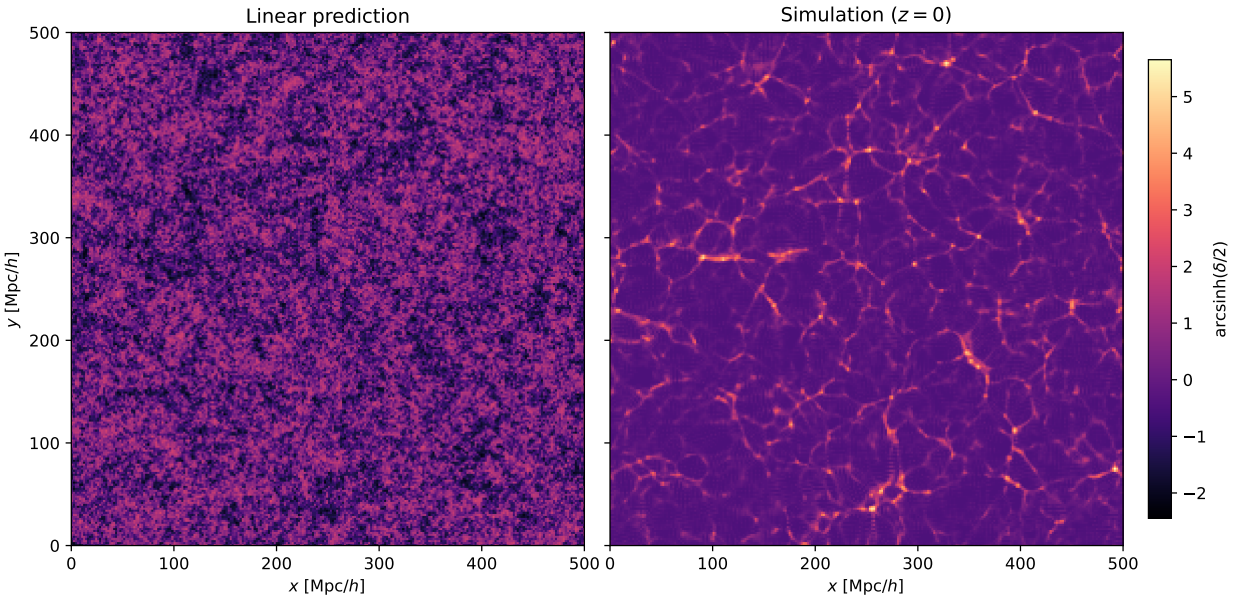


Figure 21.1.: Comparison of the linear density field (left) and the nonlinear simulation (right) at $z = 0$. The linear field is a Gaussian random field — smooth, with modest fluctuations around zero. The simulation has collapsed this into the cosmic web: thin filaments, dense nodes, and empty voids that have no counterpart in the linear field.

The contrast between the two panels is striking. The linear density field is a Gaussian random field: the fluctuations are modest ($|\delta| \lesssim 1$), the distribution is symmetric around zero, and there is no distinct "web-like" structure — just a random fluctuation field.

The simulation tells a very different story. Nonlinear gravitational evolution has reshaped the density field into the cosmic web:

- **Filaments** — thin, elongated structures where matter has streamed inward from both sides.
- **Nodes** — dense concentrations at the intersections of filaments, where halos form.
- **Voids** — large, nearly empty regions that have been evacuated as matter drains onto the surrounding filaments.

None of these features exist in the linear field. They are entirely the product of nonlinear gravitational collapse. The linear field and the simulation share the same Fourier phases — the *locations* of future structures are encoded in the initial conditions — but the *morphology* of the cosmic web is a nonlinear phenomenon.

The one-point distribution of δ makes this even clearer.

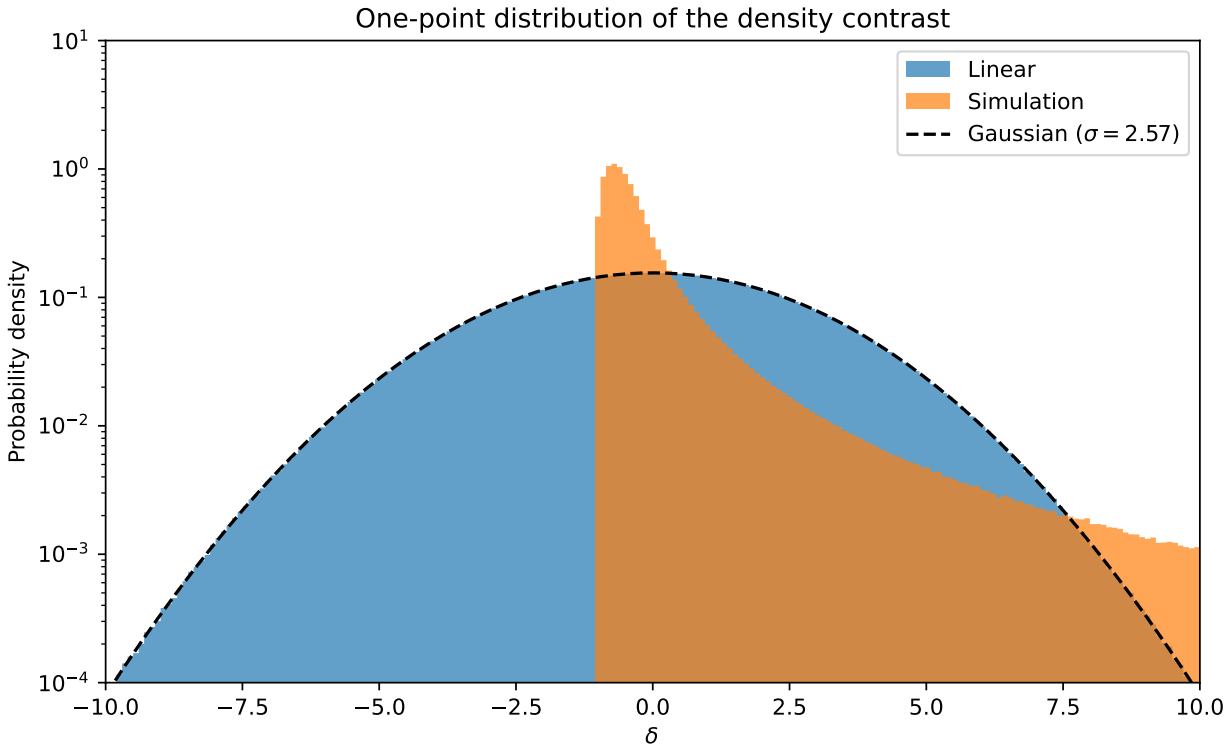


Figure 21.2.: Distribution of the overdensity δ for the linear field (blue) and the simulation (orange) at $z = 0$. The linear field is an almost perfect Gaussian centered at zero. The nonlinear field has developed a sharp cutoff near $\delta = -1$ (voids cannot go below $\rho = 0$) and a heavy positive tail from collapsed structures.

The linear field follows the Gaussian prediction almost exactly. The nonlinear distribution is strikingly different: it is bounded below by $\delta = -1$ (density cannot go negative), and the positive

tail extends to $\delta \gg 1$ from collapsed halos. This non-Gaussianity is generated entirely by gravity — the initial conditions were Gaussian by construction.

> **$\delta < -1$ in the Linear Field**
>
> The linear density field shows values below $\delta = -1$, which would correspond to negative physical density — clearly unphysical. This happens because the linear field is an *extrapolation*: we took the initial perturbations at $z = 49$ and scaled them to $z = 0$ using the linear growth factor $D_+(a) = a$, without accounting for the fact that nonlinear effects prevent real voids from reaching $\delta = -1$. The linear prediction is only valid where $|\delta| \ll 1$; by $z = 0$ the rms fluctuations are of order unity and linear theory has broken down.

A useful empirical observation: the nonlinear density field is approximately **lognormal** — that is, $\ln(1 + \delta)$ is roughly Gaussian distributed.

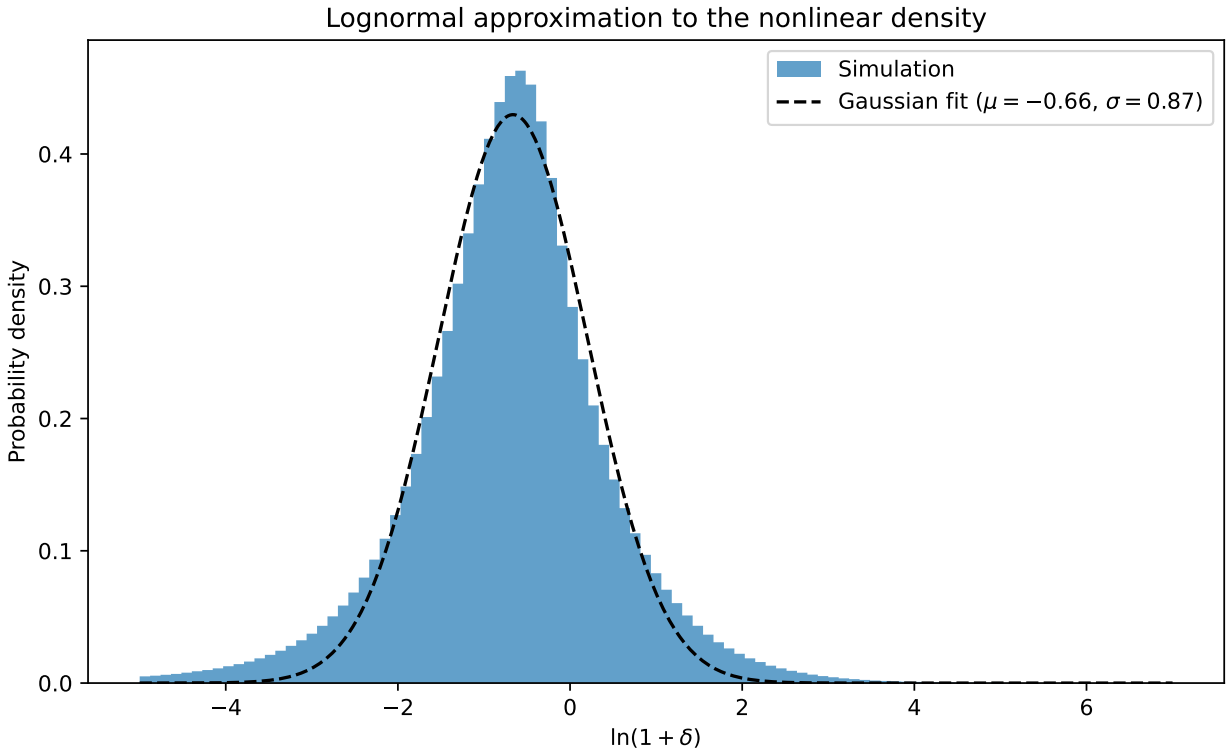


Figure 21.3.: Distribution of $\ln(1 + \delta)$ for the nonlinear density field at $z = 0$. The approximately Gaussian shape confirms that the nonlinear density is close to lognormally distributed. The dashed line shows a Gaussian fit.

21.4. Power Spectrum Measurement

The power spectrum $P(k)$ quantifies the amplitude of density fluctuations as a function of scale. We defined `measure_pk` above using the DFT conventions from Lecture 20: $\hat{\delta}_{\text{cont}}(\mathbf{k}) = h^3 \hat{\delta}_{\text{DFT}}(\mathbf{k})$, so

$$P(k) = \frac{1}{V}|\hat{\delta}_{\text{cont}}(\mathbf{k})|^2 = \frac{h^6}{V}|\hat{\delta}_{\text{DFT}}(\mathbf{k})|^2$$

averaged over spherical shells in k.

21.4.1. Validating the Initial Conditions

As a first check, we measure $P(k)$ from the linear density field used to generate the ICs and compare it to the input CLASS power spectrum. By construction, the field was built by coloring white noise with $\sqrt{P(k)}$ from CLASS, so the *expectation value* of the measured $P(k)$ equals CLASS. But any single realization fluctuates around this expectation — especially at low k where there are few modes per bin. This comparison validates both that `measure_pk` works correctly and gives a first look at sample variance.

21.4.2. Nonlinear Power Spectrum at $z = 0$

Now we compare the simulation's power spectrum at $z = 0$ to the linear field from the ICs.

The ratio plot reveals three regimes:

- **Large scales** ($k \lesssim 0.1\,h/\text{Mpc}$): the simulation tracks linear theory — perturbations on these scales are still in the linear regime at $z = 0$.
- **Intermediate scales** ($0.1 \lesssim k \lesssim 0.5\,h/\text{Mpc}$): the ratio exceeds unity — nonlinear mode coupling transfers power from large scales to small scales, boosting $P(k)$ above the linear prediction.
- **Small scales** ($k \gtrsim 0.5\,h/\text{Mpc}$): the ratio turns over and eventually drops below unity. This is not physical — it is the PM resolution limit. The CIC mass assignment and finite grid smooth the force on scales smaller than a few grid cells, suppressing small-scale power.

> **PM Resolution**
>
> The PM method resolves forces down to scales of order the grid spacing $h = L/N_g$. In Fourier space, the CIC window function and the discrete Laplacian both suppress power near the Nyquist frequency $k_{\text{Ny}} = \pi N_g/L$. Production codes use TreePM or P^{3}M methods to supplement the grid-based force with direct particle-particle interactions on small scales, eliminating this artificial suppression.

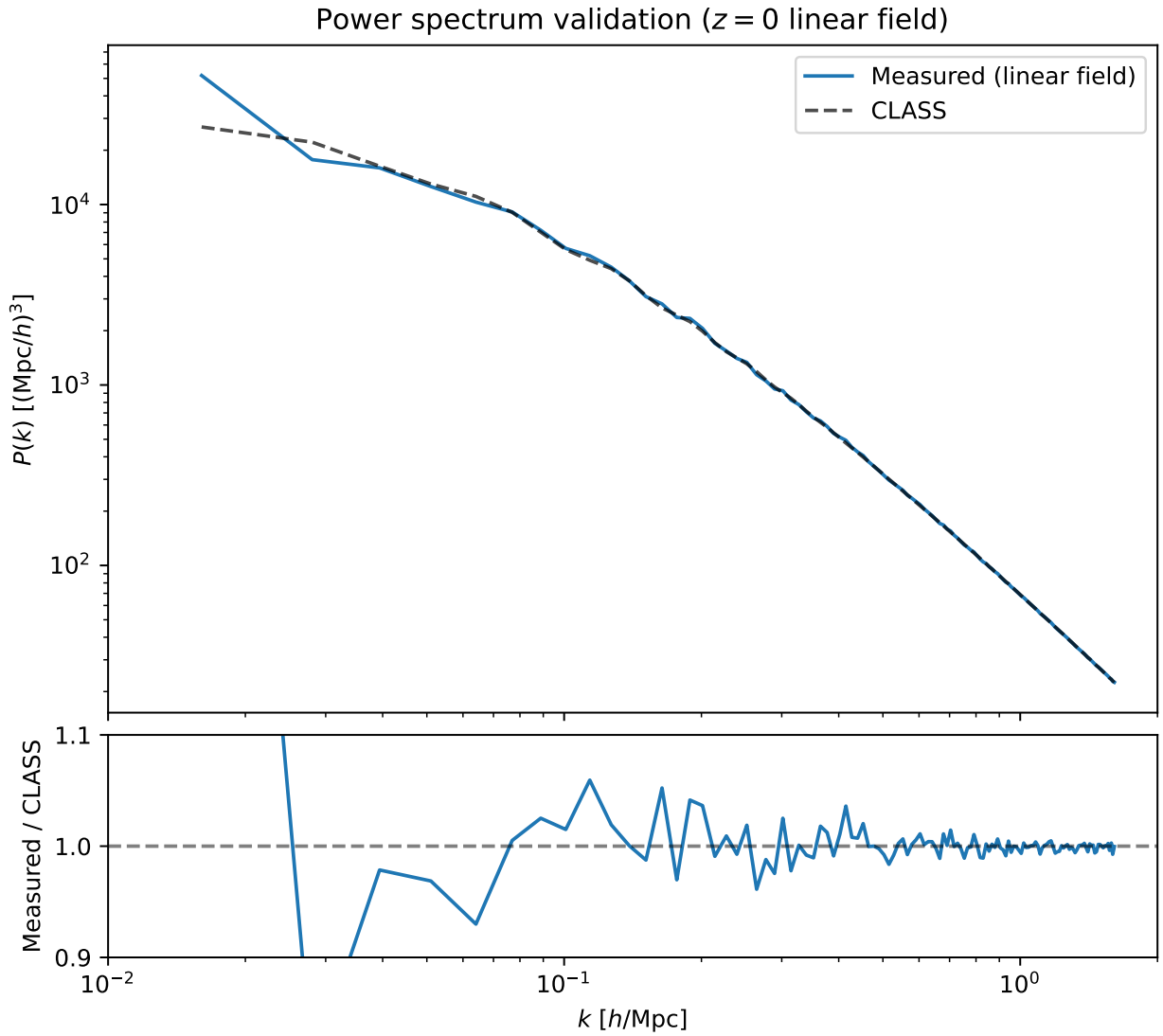


Figure 21.4.: Validation of the initial conditions and power spectrum measurement. The measured $P(k)$ from the $z = 0$ linear density field (blue) agrees with the input CLASS power spectrum (black dashed). The lower panel shows the ratio, confirming percent-level agreement across the resolved range.

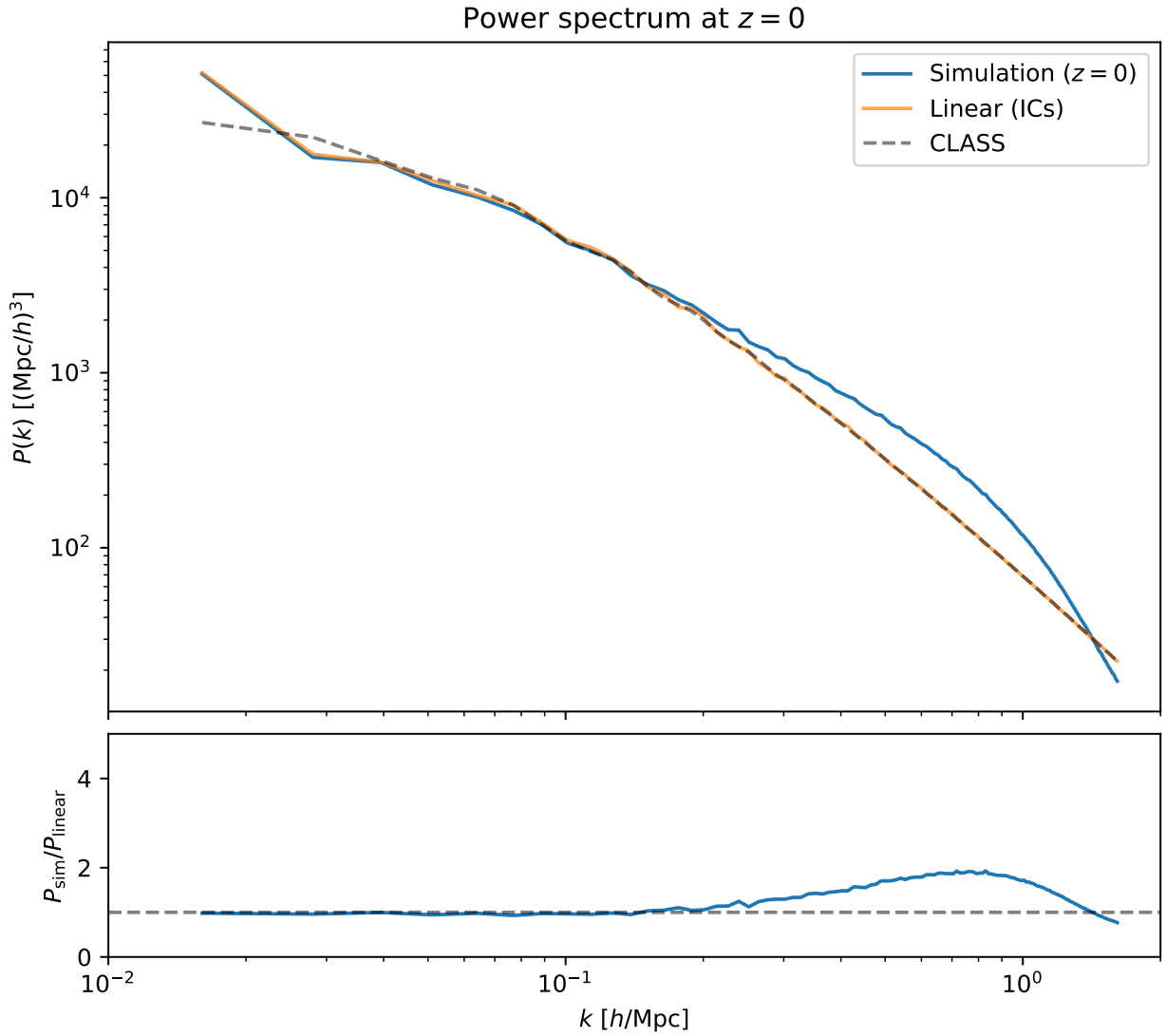


Figure 21.5.: Power spectrum at $z = 0$: simulation (blue) versus the linear IC field (orange) and CLASS (black dashed). The simulation tracks the linear prediction on large scales, shows nonlinear enhancement at intermediate scales, and falls below at high k due to the PM force resolution limit.

21.5. Growth Factor Evolution

In linear theory, all Fourier modes grow at the same rate: $P(k, a) = D_+(a)^2\, P(k, a_{\rm init})/D_+(a_{\rm init})^2$. For EdS, $D_+(a) = a$, so the ratio $P(k,a)/P(k,a_{\rm init})$ should equal $(a/a_{\rm init})^2$ for modes in the linear regime. Nonlinear modes grow faster than this prediction.

We measure $P(k)$ at each of our saved snapshots ($z = 9, 4, 1, 0$) and first show the full power spectra, then track the growth of individual k-bins.

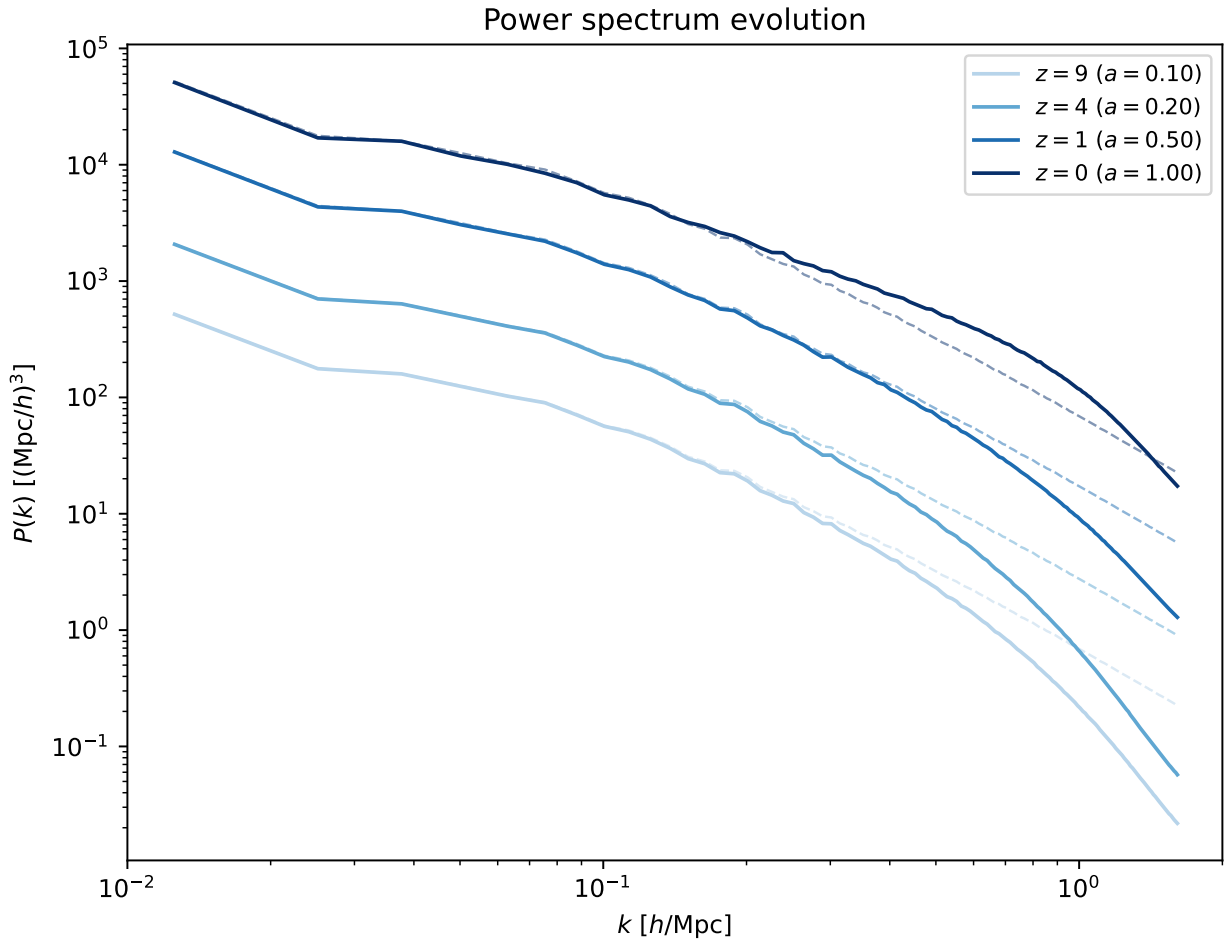


Figure 21.6.: Power spectra at four redshifts, from $z = 9$ (lightest) to $z = 0$ (darkest). Solid lines are the simulation; dashed lines show the linear prediction (IC field scaled by $(a/a_i)^2$). At early times the two agree; by $z = 0$ nonlinear growth boosts intermediate scales while PM resolution and the CIC window suppress high k.

Now we track the growth of specific k-bins as a function of scale factor, normalized by their initial values.

The large-scale modes ($k \lesssim 0.05\,h/\mathrm{Mpc}$) follow the linear prediction $(a/a_i)^2$ almost exactly — these modes are still in the linear regime even at $z = 0$. At higher k, the growth exceeds the linear prediction: nonlinear mode coupling (the transfer of power from large to small scales) accelerates

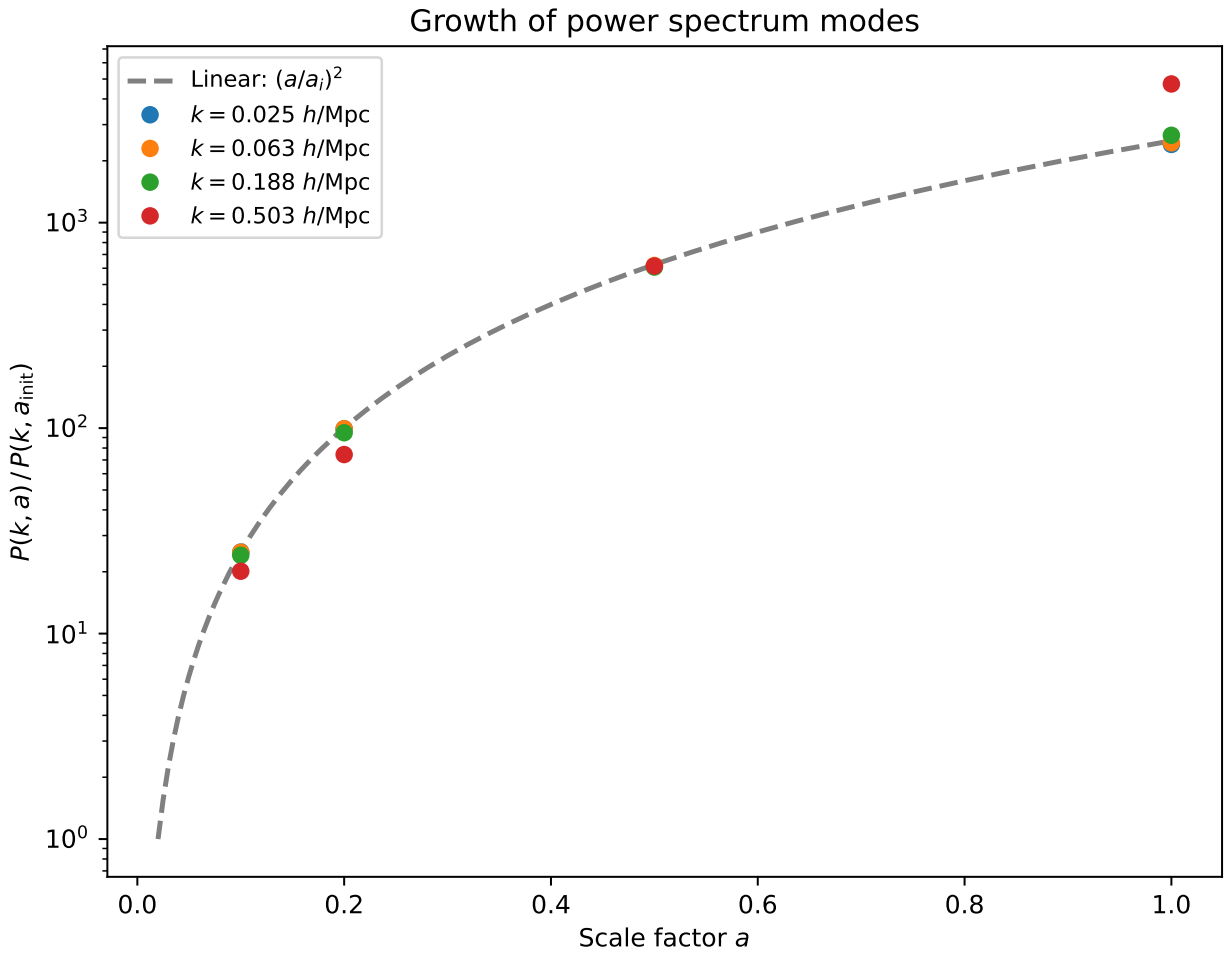


Figure 21.7.: Growth of the power spectrum with scale factor for several k-bins. The dashed line shows the EdS linear prediction $(a/a_i)^2$. Large-scale modes (low k) track linear theory closely, while small-scale modes grow faster due to nonlinear mode coupling.

the growth of small-scale power. This is the same nonlinear enhancement we saw in the $P(k)$ ratio plot, but now tracked as a function of time.

21.6. Redshift-Space Density Field

In observations, we do not measure the true (real-space) positions of galaxies. Instead, we infer distances from redshifts, which include a contribution from peculiar velocities. This maps the real-space position $\mathbf{x}$ to a **redshift-space position** $\mathbf{s}$:

$$\mathbf{s} = \mathbf{x} + \frac{v_{\mathrm{pec},\parallel}}{aH}\hat{\mathbf{n}}$$

where $\hat{\mathbf{n}}$ is the line-of-sight (LOS) direction and $v_{\mathrm{pec},\parallel}$ is the peculiar velocity component along it.

In the **plane-parallel approximation** (valid when the survey is far enough away that all lines of sight are nearly parallel), we pick a fixed LOS direction — here the z-axis. In our code units at $a = 1$ (EdS, $H_0 = 1$, $m = 1$): the momentum $\mathbf{p} = a^2\dot{\mathbf{x}}$ equals $a\,\mathbf{v}_{\mathrm{pec}}$, so at $a = 1$, $\mathbf{p} = \mathbf{v}_{\mathrm{pec}}$ and $aH = 1$, giving simply

$$s_z = x_z + p_z, \qquad s_x = x_x, \qquad s_y = x_y$$

21.6.1. Slices Parallel to the LOS

Since the RSD displacement is along the z-axis, we need a slice that includes z to see the effect. We show an x–z slice at fixed y, comparing real space and redshift space side by side, followed by a zoomed-in view of a dense region.

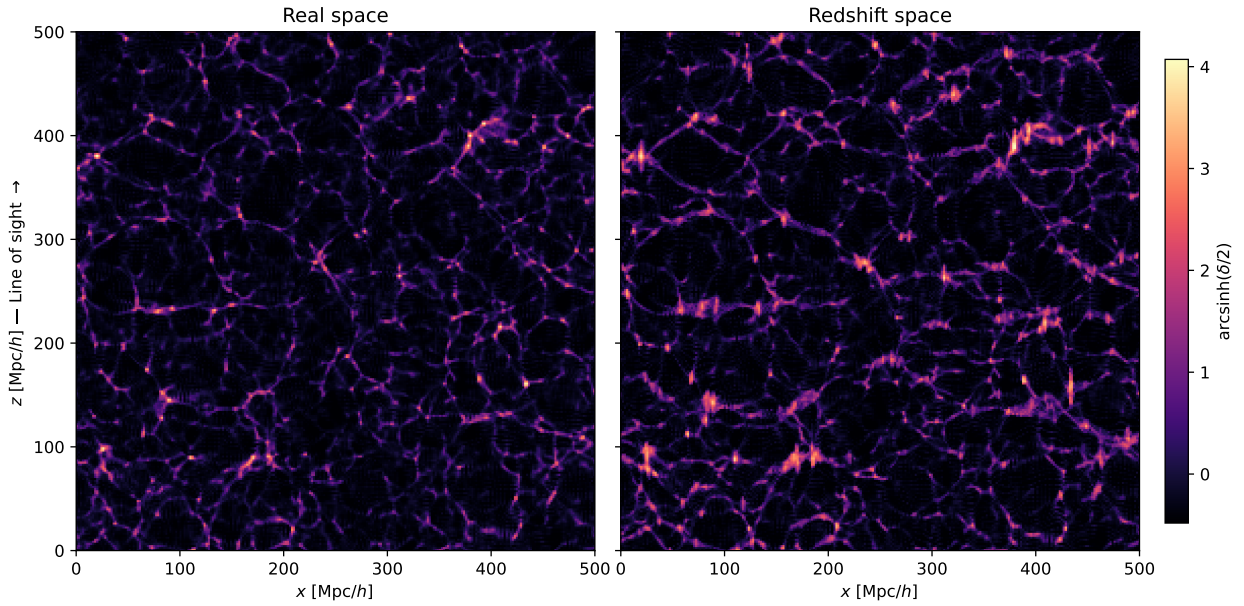


Figure 21.8.: Real-space (left) and redshift-space (right) density in an x–z slice. The vertical axis is the line of sight. Coherent infall squashes large-scale structures along z (Kaiser effect), while virialized motions in clusters elongate them (Fingers of God).

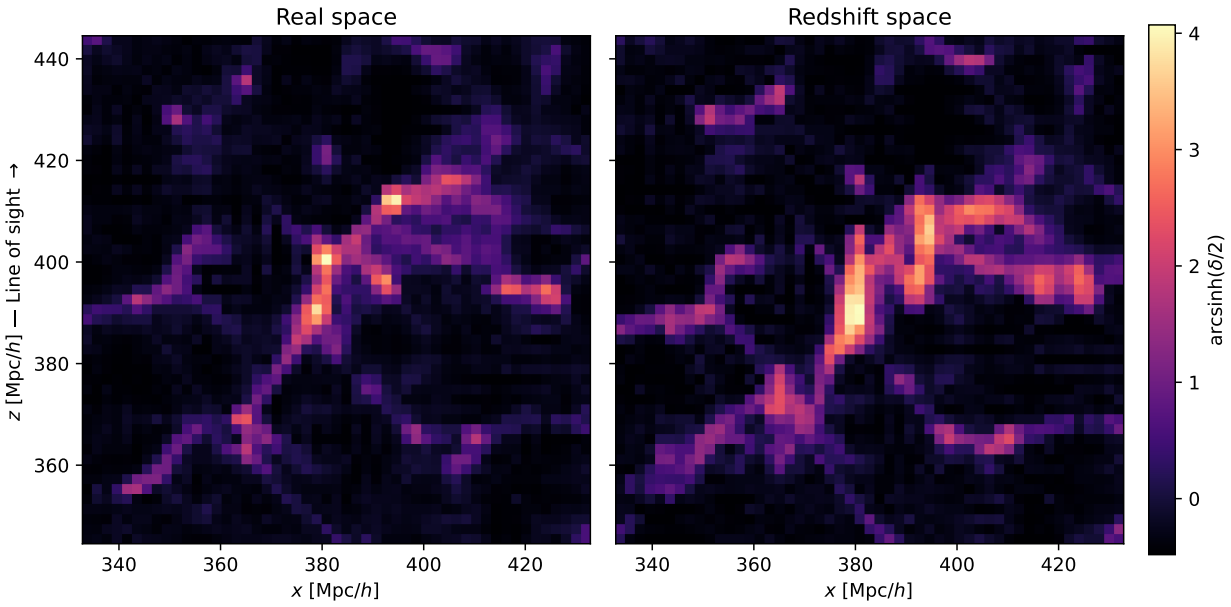


Figure 21.9.: Zoomed view of a 100×100 $(\mathrm{Mpc}/h)^2$ region in real space (left) and redshift space (right). Dense clusters that appear compact in real space are stretched along the line of sight (vertical) in redshift space — the classic Fingers of God.

Now we zoom in on a dense region to see the Fingers of God more clearly.

Two effects are at work in these slices:

- **Kaiser squashing**: large-scale overdensities appear compressed along the z-axis in redshift space. This is because matter is falling coherently toward the overdensity — the infalling material on the near side has a peculiar velocity *toward* us (reducing its redshift), while material on the far side moves *away* (increasing its redshift). Both shifts push the material closer together in redshift space.
- **Fingers of God**: dense clusters appear elongated along z. Inside a virialized halo, galaxies have large random velocities that scatter their redshifts, smearing the cluster along the line of sight.

The Kaiser effect is subtle in individual slices — it is a large-scale coherent compression that is difficult to see by eye. In the next section, we will measure the power spectrum *multipoles*, which quantify the anisotropy statistically and make the Kaiser signal unambiguous.

i Plane-Parallel Approximation

We assumed a single, fixed line of sight for the entire box. This is the plane-parallel (or distant-observer) approximation, valid when the box subtends a small angle on the sky. For a real survey covering a large fraction of the sky, the LOS varies across the volume and one must account for the changing projection direction — the so-called "wide-angle" effects.

21.6.2. 2D Correlation Function

A more quantitative view of the anisotropy comes from the two-point correlation function $\xi(r_\perp, r_\parallel)$, where $r_\parallel$ is the separation along the LOS and $r_\perp$ is perpendicular to it. We compute this by FFT: $\xi(\mathbf{r}) = \mathrm{IFFT}(|\hat{\delta}(\mathbf{k})|^2)/N_g^3$.

In real space, ξ depends only on $|\mathbf{r}|$ — the contours are circular. In redshift space, the Kaiser effect squashes the contours along $r_\parallel$ on large scales (coherent infall makes structures appear closer together along the LOS), while the Fingers of God elongate them on small scales.

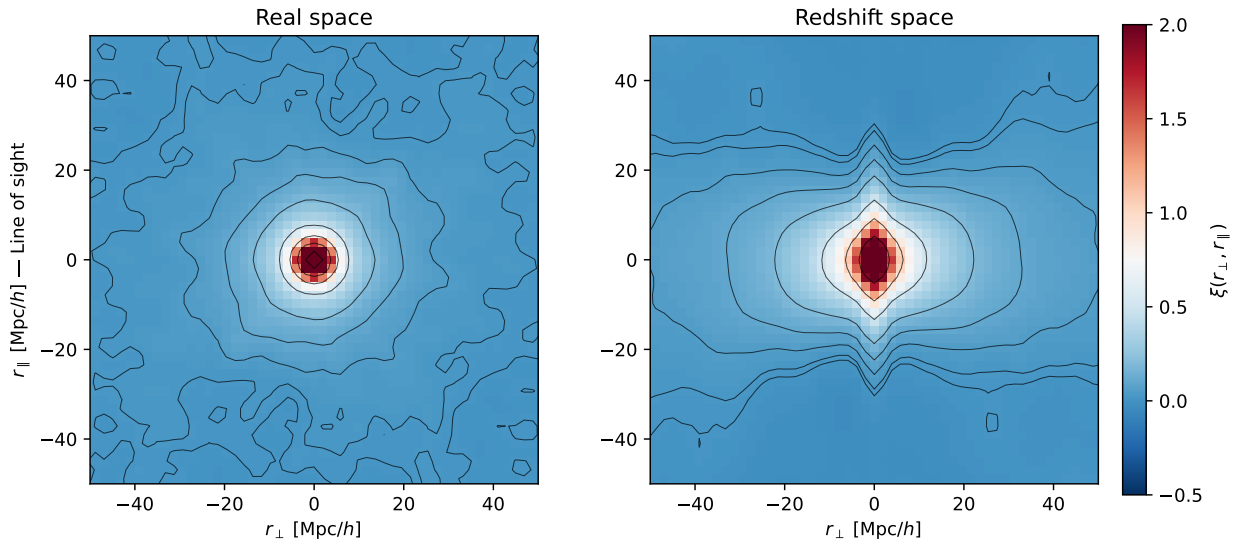


Figure 21.10.: Two-point correlation function $\xi(r_\perp, r_\parallel)$ in real space (left) and redshift space (right). In real space the contours are nearly circular (isotropic). In redshift space, the Kaiser effect squashes the contours along $r_\parallel$ at large separations, while the Fingers of God elongate them at small separations.

The contour plot makes the RSD anisotropy unmistakable. In real space, the contours of constant ξ are roughly circular — the clustering is isotropic. In redshift space, two distortions are visible:

- At **large separations**, the contours are squashed along $r_\parallel$ (they are wider in $r_\perp$ than in $r_\parallel$). This is the Kaiser effect: coherent infall compresses structures along the LOS.
- At **small separations**, the contours are elongated along $r_\parallel$. This is the Fingers of God: random velocities inside halos smear out the clustering along the LOS.

21.7. RSD Power Spectrum Multipoles

21.7.1. Theory: Kaiser Formula

In linear theory, the redshift-space power spectrum depends on the angle $\mu = \hat{\mathbf{k}} \cdot \hat{\mathbf{n}}$ between the wavevector and the line of sight:

$$P^s(k, \mu) = (1 + \beta\mu^2)^2 \, P(k)$$

where $\beta = f/b$, $f = d\ln D_+/d\ln a$ is the growth rate, b is the linear bias, and $P(k)$ is the real-space power spectrum. For EdS dark matter: $f = 1$ and $b = 1$, so $\beta = 1$.

Expanding in Legendre polynomials $\mathcal{L}_\ell(\mu)$, the multipoles are

$$P_\ell(k) = \frac{2\ell+1}{2}\int_{-1}^{1} P^s(k,\mu)\,\mathcal{L}_\ell(\mu)\,d\mu$$

For the Kaiser formula with $\beta = 1$:

$$P_0(k) = \left(1 + \frac{2}{3}\beta + \frac{1}{5}\beta^2\right)P(k) = \frac{28}{15}P(k)$$

$$P_2(k) = \left(\frac{4}{3}\beta + \frac{4}{7}\beta^2\right)P(k) = \frac{40}{21}P(k)$$

21.7.2. Measuring Multipoles

We measure $P(k,\mu)$ on a 2D grid, then integrate over μ with Legendre weights. The `kgrid` already stores the bin indices for both k and $|\mu|$.

21.7.3. Single Realization

We measure the monopole $P_0(k)$ and quadrupole $P_2(k)$ from the redshift-space density field of our single $z = 0$ realization, along with the real-space $P(k)$ for comparison with the Kaiser prediction.

The monopole is reasonably well described by the Kaiser formula on large scales ($k \lesssim 0.2\,h$/Mpc), but the quadrupole is much noisier. This is not surprising: the quadrupole is a higher-order statistic that requires more modes to average down the variance. At low k, there are very few modes per spherical shell, and the μ-dependence is poorly sampled.

Notice that P_2 actually *exceeds* P_0 — the Kaiser coefficients are $28/15 \approx 1.87$ for the monopole and $40/21 \approx 1.90$ for the quadrupole when $\beta = 1$. This is a peculiarity of EdS dark matter ($f = 1$, $b = 1$). In a realistic galaxy survey, both $f < 1$ (in ΛCDM, $f \approx \Omega_m^{0.55} \lesssim 1$) and $b > 1$ for typical tracers, so $\beta = f/b$ is well below unity and the quadrupole is much smaller than the monopole — making it harder to measure and more sensitive to systematic errors.

At small scales, both multipoles deviate from Kaiser. The Fingers of God suppress power along the LOS, reducing P_0 below the Kaiser prediction. The quadrupole P_2 actually goes negative at high k — the anisotropy reverses sign as the elongation from FoG dominates over the Kaiser squashing. (We plot $|P_2|$ on the log scale, so this sign change shows up as the ratio crossing zero in the lower panel.) The Kaiser formula, which assumes linear velocities, cannot capture this nonlinear effect.

> **i** Sample Variance
>
> The scatter in the multipoles — especially P_2 — is **sample variance** (also called cosmic variance). A single realization of a finite box contains a limited number of independent Fourier modes at each k. The measured power in each bin fluctuates around the true expectation value, and this scatter is worst at low k where there are fewest modes. In the next section, we will run an ensemble of simulations to beat down this variance and extract a clean signal.

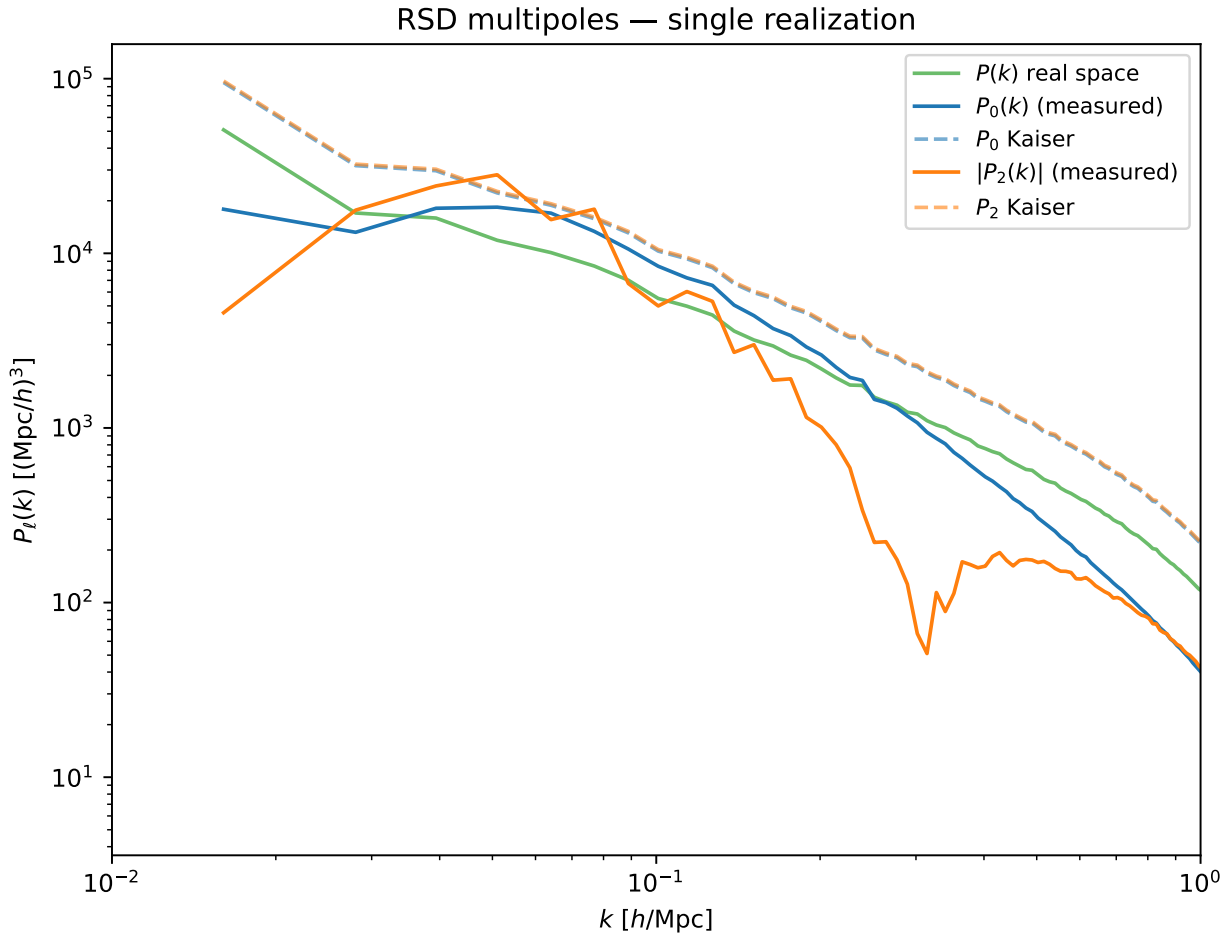


Figure 21.11.: RSD multipoles from a single realization: monopole $P_0(k)$, quadrupole $|P_2(k)|$, and real-space $P(k)$, compared to the Kaiser prediction (dashed). The quadrupole is noisy — a consequence of sample variance from a finite number of modes per k-bin.

21.7.4. Ensemble of Realizations

To reduce sample variance, we run 10 independent simulations with different random seeds (but the same cosmology and box parameters). For each realization, we measure the real-space $P(k)$, the redshift-space multipoles $P_0(k)$ and $P_2(k)$, and the 2D correlation function.

> **!** Precomputed Ensemble
>
> The ensemble simulations are too expensive to run inside the notebook. They were precomputed separately using `scripts/run_ensemble.py` and cached to `~/data/teaching/ASTR6000-SP26/021-pm-analysis/ensemble/`. To regenerate:
>
> ```
> pixi run python scripts/run_ensemble.py --n-real 10
> ```
>
> This takes ~15 minutes (10 simulations × ~90s each).

```
Loading 10 realizations from /Users/npadmana/data/teaching/ASTR6000-SP26/021-pm-analysis/ensemble
  k bins: 128, realizations: 10
```

Spaghetti Plots

First, we overlay all realizations to visualize the scatter.

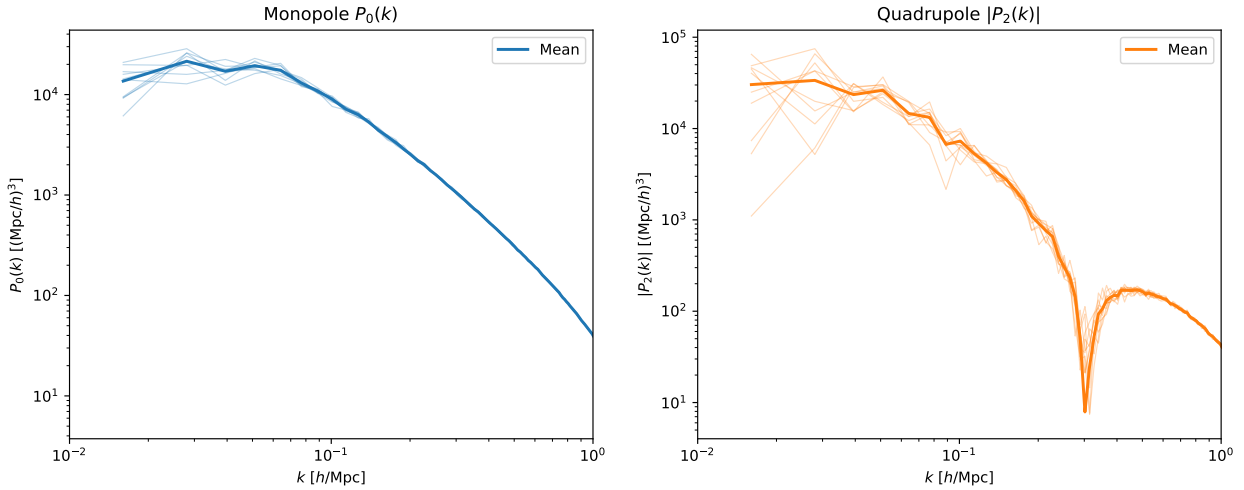


Figure 21.12.: All 10 realizations overplotted for the monopole $P_0(k)$ (left) and quadrupole $|P_2(k)|$ (right). The spread between realizations is sample variance — it is largest at low k where there are fewest independent modes.

The scatter is dramatically worse for the quadrupole — consistent with what we saw in the single realization. At $k \lesssim 0.03\,h/\mathrm{Mpc}$, the individual quadrupole measurements jump around wildly, while the monopole is relatively stable even at the lowest k.

Mean and Error Bars vs Kaiser

Finally, the ensemble mean with error bars, compared to the Kaiser prediction. We also show the mean real-space $P(k)$ compared to the IC linear prediction.

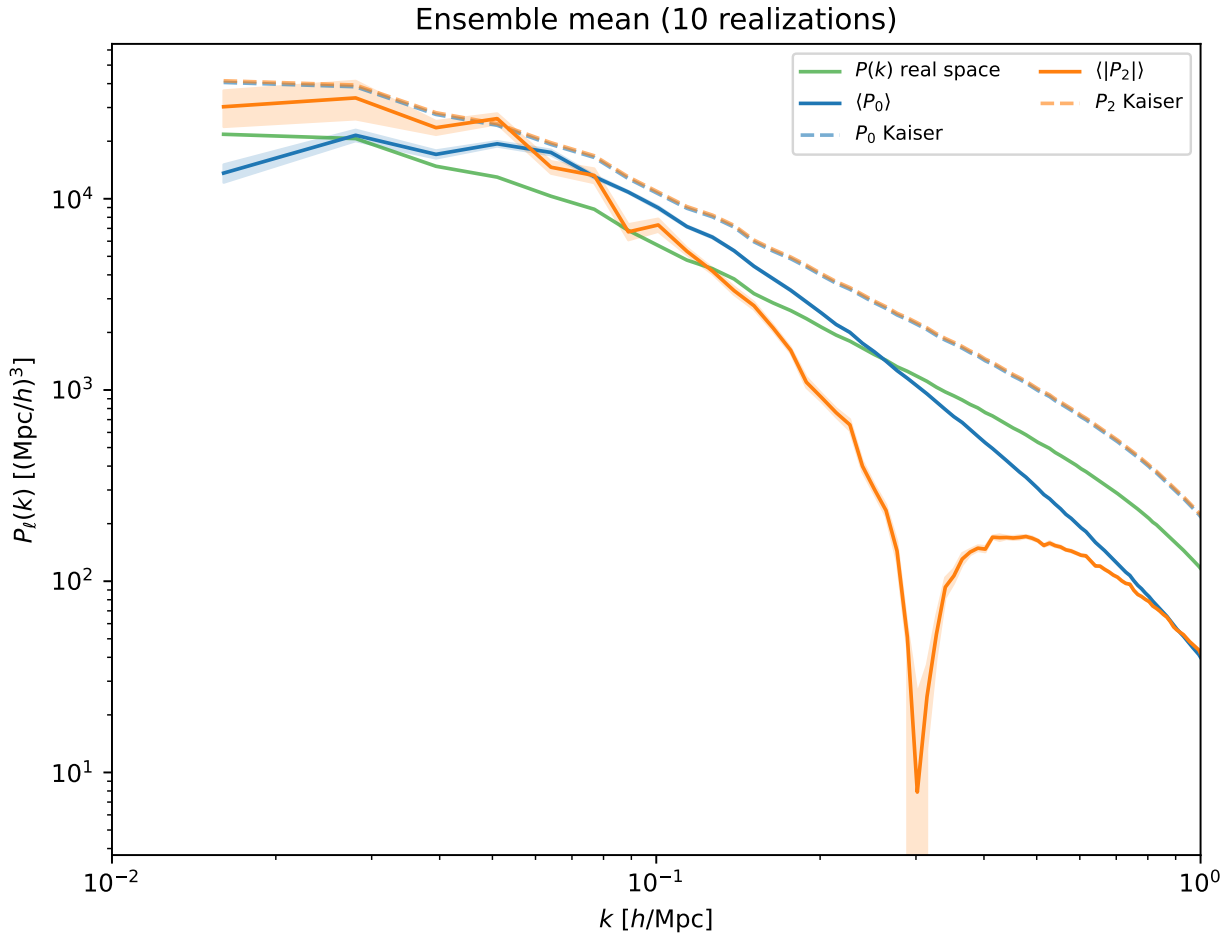


Figure 21.13.: Ensemble mean of $P_0(k)$ and $P_2(k)$ (with standard error shading) compared to the Kaiser prediction (dashed) and the real-space $P(k)$. The Kaiser formula captures the qualitative behavior; deviations at small scales are from Fingers of God and PM resolution.

Real-Space Power Spectrum vs Linear Theory

As a consistency check, we compare the ensemble-averaged real-space $P(k)$ to the CLASS linear prediction.

At intermediate and high k, the ensemble mean shows the expected nonlinear enhancement over CLASS. On large scales, where we expect agreement with linear theory, the ensemble mean sits a few percent below CLASS. This is a real effect: our finite time-stepping does not perfectly reproduce the exact linear growth rate, leading to a small systematic deficit in the evolved power. Production

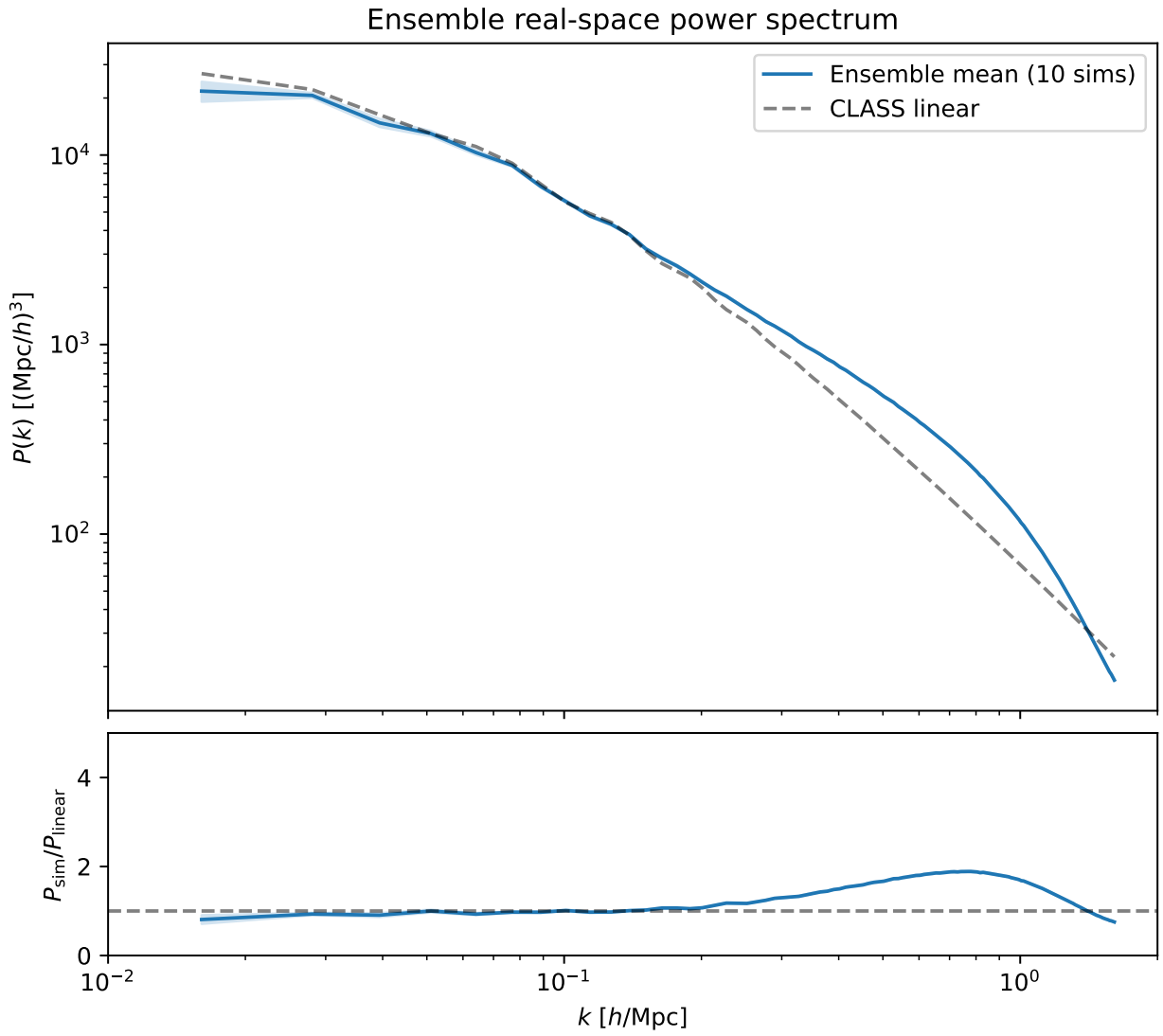


Figure 21.14.: Ensemble mean of the real-space $P(k)$ compared to the CLASS linear prediction. The error bars (standard error on the mean) are much smaller than the single-realization scatter, confirming that the ensemble average converges to the true power spectrum.

codes address this with more timesteps, higher-order integrators, or more careful treatment of the drift and kick factors.

The ensemble averaging also reveals the importance of large volumes. With a single realization (Section 2), the sample variance at low k was large enough to mask this systematic deficit — the noise was bigger than the signal we are trying to see. Only by averaging over 10 realizations could we beat down the scatter enough to notice it. The Quijote suite uses boxes $2\times$ larger in each dimension ($L = 1\,h^{-1}\mathrm{Gpc}$) and thousands of realizations, enabling percent-level precision on clustering statistics.

Ensemble-Averaged 2D Correlation Function

Averaging the 2D correlation function $\xi(r_\perp, r_\parallel)$ across realizations dramatically reduces noise, giving a much cleaner view of the RSD anisotropy than the single-realization plot from Section 4.

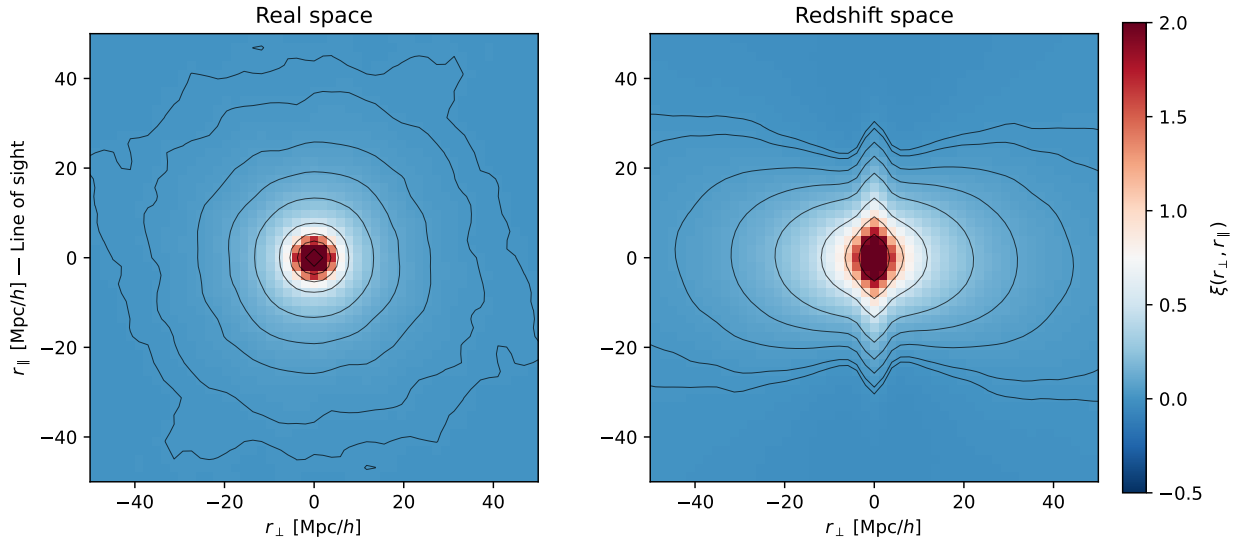


Figure 21.15.: Ensemble-averaged 2D correlation function $\xi(r_\perp, r_\parallel)$ in real space (left) and redshift space (right). Averaging over 10 realizations reveals clean, smooth contours. The Kaiser squashing (compression along $r_\parallel$) and Fingers of God (elongation at small separations) are now clearly visible.

Connection to Quijote

In Lectures 18–19, we analyzed power spectra from the Quijote simulation suite — thousands of N-body realizations in $1\,(h^{-1}\mathrm{Gpc})^3$ boxes with full ΛCDM gravity. What we have done here is the same idea in miniature: multiple realizations of the same cosmology, used to estimate the mean and variance of clustering statistics. The Quijote suite uses larger boxes (better sampling of large-scale modes), higher resolution (TreePM, not PM), and many more realizations — but the statistical framework is identical.

Part V.

PHYS6600: Neutrinos, DM etc.

22. Neutrinos in Cosmology - I

22.1. Preface

Just a note on these lectures - these are not part of our regular development, but are additional lectures written for PHYS6600. They are therefore intended to be more standalone lectures and will repeat some of the material in other parts of the course.

The lectures here largely follow Baumann's textbook closely with some commentary interspersed.

I plan for these lectures to roughly follow three broad themes:

- Relativistic Neutrinos (and other species) - Equilibrium thermodynamics in the Early Universe.
- The Boltzmann equation - freezing out of particles; WIMP freeze out and other examples.
- Massive Neutrinos - measuring the neutrino mass through large scale structure.

22.2. A Brief Review of Cosmology

22.2.1. Overview

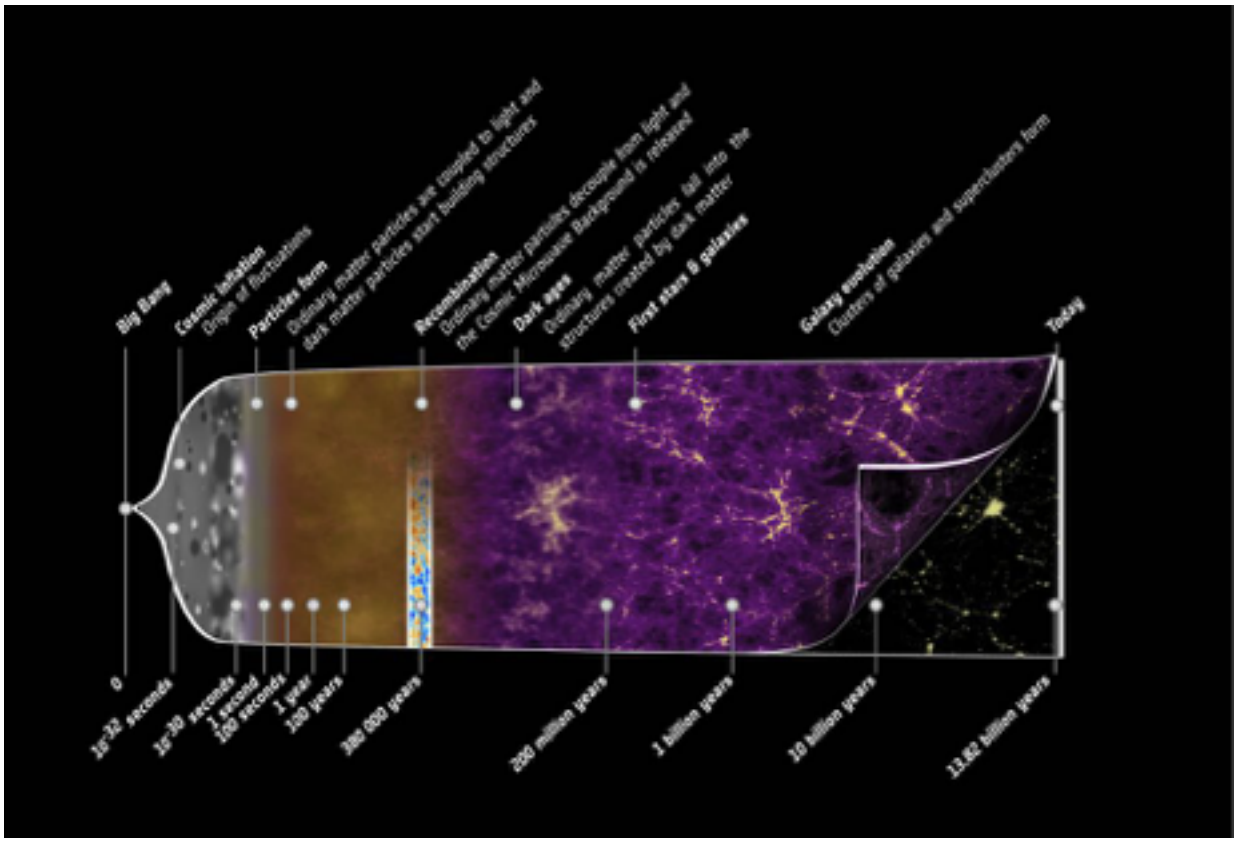


Figure 22.1.: A history of the Universe (ESA and Planck collaboration)

The figure above is an overview of the entire history of the Universe, from the Big Bang to today. Neutrinos play a role in this picture throughout, but especially in two epochs - the early radiation dominated epoch where they are effectively massless and help determine the expansion rate of the Universe. And their second act is at late times, when they transition to non-relativistic matter and act as warm dark matter. This first lecture focuses on the first of these.

The table below highlights some key times in the history of the Universe. This lecture will largely focus on the times < 10s after the Big Bang/inflation, when the temperature drops down to ~ 500 keV, just below the rest mass of the electron. The imprints of these will be seen both in BBN as well as the CMB.

Table 3.1 Key events in the history of the early universe

Event	Time	Redshift	Temperature
Inflation	?	–	–
Baryogenesis	?	?	?
Dark matter freeze-out	?	?	?
EW phase transition	20 ps	10^{15}	100 GeV
QCD phase transition	20 µs	10^{12}	150 MeV
Neutrino decoupling	1 s	6×10^9	1 MeV
Electron–positron annihilation	6 s	2×10^9	500 keV
Big Bang nucleosynthesis	3 min	4×10^8	100 keV
Matter–radiation equality	50 kyr	3400	0.80 eV
Recombination	290–370 kyr	1090–1270	0.25–0.29 eV
Photon decoupling	370 kyr	1090	0.25 eV

Figure 22.2.: Table 3.1 from Baumann

22.2.2. The Friedmann Universe

In all of what follows, we will use natural units $c = \hbar = 1$.

When we describe our Universe, we often break it into a homogeneous expansion, and the effects of small perturbations on this homogeneous expansion. We'll do the same here today, and for the first set of lectures, we'll focus in on the homogeneous Universe. We describe our expanding Universe by a scale factor $a(t)$ which measures the size of the Universe relative to its size today. The redshift is given by

$$1 + z = \frac{1}{a}$$

The Hubble parameter measures the expansion rate as is defined by

$$H = \left(\frac{\dot{a}}{a}\right)$$

with a value today of $H_0 \approx 70$ km/s/Mpc. Cosmologists often parametrize the remaining uncertainty in this value by writing $H_0 = 100h$ km/s/Mpc, which results in factors of h in expressions.

The expansion rate is determined by the energy density in the Universe. This is described by the Friedmann equation

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3}\rho(t) - \frac{k}{R_0^2 a^2}$$

and the continuity equation

$$\dot{\rho} + 3H(\rho + P) = 0$$

where the latter differs from the regular continuity equation due to the expansion history of the Universe.

Given an equation of state, the continuity equation allows us to describe the evolution of different components of the Universe. In cosmology, we assume a simple equation of state of the form $P = w\rho$. We have the following cases that are important…

- Pressure-less Matter/Dust

$$w = 0 \implies \rho = \rho_0 a^{-3}$$

- Radiation

$$w = \frac{1}{3} \implies \rho = \rho_0 a^{-4}$$

- Cosmological Constant

$$w = -1 \implies \rho = \rho_0$$

We can put this together a single figure that shows the evolution of each of these terms with scale factor:

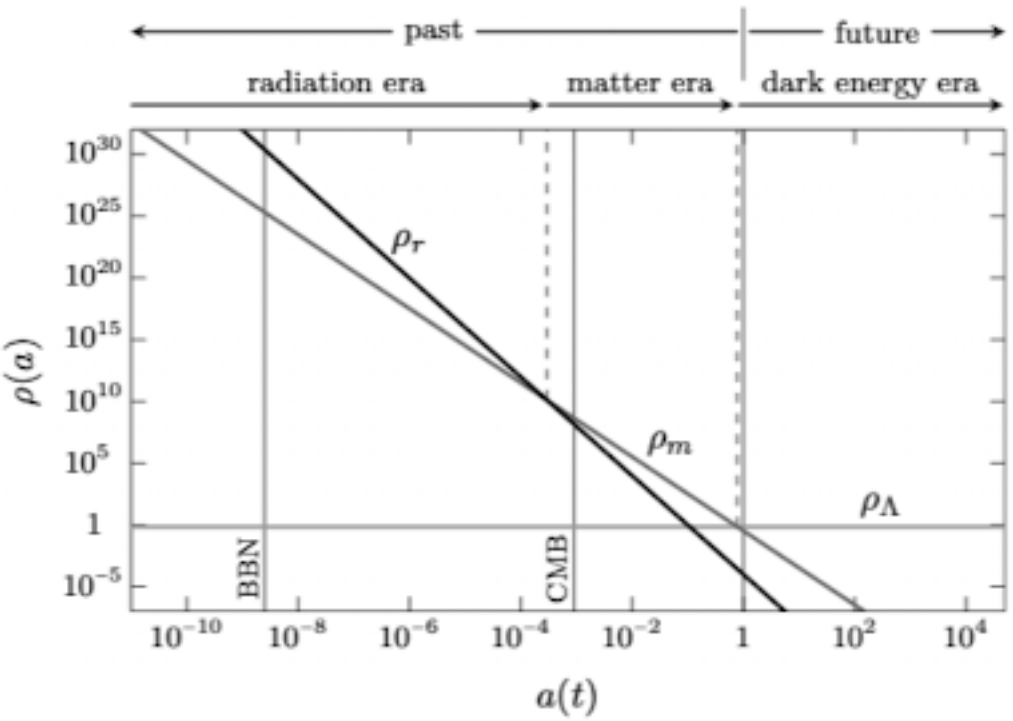


Figure 22.3.: Evolution of Energy Densities (Baumann Fig. 2.10)

Finally, we note that $T \propto a^{-1}$ (at least away from particle masses, we'll see a more accurate version later). What this means is that for the period of interest today, the Universe will be radiation

dominated, and the Friedmann equation becomes

$$H^2 = \frac{\rho}{3M_{pl}^2}$$

where

$$M_{pl}^2 \equiv \frac{\hbar c}{8\pi G} = 2.4 \times 10^{18}\,\mathrm{GeV}$$

We will also sometimes talk about densities relative to the critical density

$$\rho_{\mathrm{crit}} = \frac{3H^2}{8\pi G}$$

Note that the value of the critical density depends on H and therefore varies with time. The value today is

$$\begin{aligned}\rho_{\mathrm{crit}} &= 1.877 \times 10^{-26} h^2 \mathrm{kg\,m^{-3}} \\ &= 2.775 \times 10^{11} h^2 M_\odot \mathrm{Mpc}^{-3} \\ &= 1.053 \times 10^{10} \mathrm{eV}^4\end{aligned}$$

We can measure the density relative to the critical density

$$\Omega = \frac{\rho}{\rho_{\mathrm{crit}}}$$

22.3. A Brief Review of Thermodynamics

Since we're talking about the thermal history of the Universe, recall some key results from thermodynamics/statistical mechanics. We'll assume equilibrium throughout, so will assume a temperature T. All of the results here can be derived by starting from either the Bose-Einstein or Fermi-Dirac equations and integrating over the density of states.

A relativistic gas of a species with negligible chemical potential has a number density

$$n = \frac{\zeta(3)}{\pi^2} g T^3 \begin{cases} 1 & \text{bosons} \\ \frac{3}{4} & \text{fermions} \end{cases}$$

where g is the number of internal degrees of freedom it has. We get a analogous relation for the density

$$\rho = \frac{\pi^2}{30} g T^4 \begin{cases} 1 & \text{bosons} \\ \frac{7}{8} & \text{fermions} \end{cases}$$

and the pressure in the relativistic regime is given by

$$P = \frac{\rho}{3}$$

Note that the density evolves as $\rho \propto T^4$ which is consistent with our previous scaling with radiation. We should note here that radiation includes all relativistic particles - photons, neutrinos and as we go back in time, all standard model particles.

As an exercise, we can use the above to calculate the number density and energy density of the CMB photons today. Using $T_{\rm CMB} \approx 2.73\,K$, we can plug in to the above and find

$$\begin{aligned} n_{\gamma,0} &\approx 410\,\text{photons}\,\text{cm}^{-3} \\ \rho_{\gamma,0} &\approx 4.6 \times 10^{-34}\,\text{g}\,\text{cm}^{-3} \\ \Omega_\gamma h^2 &\approx 2.5 \times 10^{-5} \end{aligned}$$

In the non-relativistic limit, we find

$$n = g\left(\frac{mT}{2\pi}\right)^{3/2} e^{-m/T}$$

and

$$\rho \approx mn + \frac{3}{2}nT$$

$$P = nT \approx 0$$

For our purposes today, this means that as a particle goes non-relativistic, its number density is exponentially suppressed and it quickly becomes irrelevant. This statement is only true in equilibrium - we will return to this later.

22.4. Counting States - I

Given the above, understanding the expansion history of the Universe during radiation domination is therefore just a matter of accounting for all the relativistic species

$$\rho = \sum_i \rho_i$$

It is convenient to write this as

$$\rho = \frac{\pi^2}{30} g_*(T) T^4$$

where we have defined an effective number of states

$$g_*(T) = \sum_{i=\text{bosons}} g_i \left(\frac{T_i}{T}\right)^4 + \frac{7}{8} \sum_{i=\text{fermions}} g_i \left(\frac{T_i}{T}\right)^4$$

In the above, we allow for the possibility that different species have different temperatures, which is possible when they decouple from the overall thermal bath. The temperature T is the common temperature of the rest, often just assumed to be the photon temperature. In the standard model, this effect is only relevant for neutrinos, which decouple the first, since they have the weakest cross section.

We can now go ahead and count internal degrees of states. The two tables below (taken from Baumann) describe the particle content of the standard model, highlighting the masses of the particles as well as their spin. When $T_i \gg m$, the particles will be relativistic and will contribute to the overall radiation density of the Universe. In the opposite limit (assuming the particles are still in equilibrium), the particles rapidly annihilate and their number density is exponentially suppressed.

Note that this assumes equilibrium, which is true for our story today. However, non-equilibrium effects play a critical role in the number density calculations and we'll turn to these in the next lecture.

Some rules here:

- Massless particles with spin have two polarization states.
- Massive particles of spin s has $2s + 1$ polarization states.
- Count both particles and anti-particles.
- Quarks have 3 colors, while there are 8 gluons.

Type	Particle	Mass	Spin	g
gauge bosons	γ	0	1	2
	$W^{\pm}$	80 GeV	1	3
	Z	91 GeV		
gluons	g_i	0	1	$8 \times 2 = 16$
Higgs boson	H	125 GeV	0	1
quarks	$t, \bar{t}$	173 GeV	$\frac{1}{2}$	$2 \times 3 \times 2 = 12$
	$b, \bar{b}$	4 GeV		
	$c, \bar{c}$	1 GeV		
	$s, \bar{s}$	100 MeV		
	$d, \bar{d}$	5 MeV		
	$u, \bar{u}$	2 MeV		

Particle	Mass	Spin	g
$\tau^{\pm}$	1777 MeV	$\frac{1}{2}$	$2 \times 2 = 4$
$\mu^{\pm}$	106 MeV		
$e^{\pm}$	511 keV		
$\nu_\tau, \bar{\nu}_\tau$	< 0.6 eV	$\frac{1}{2}$	$2 \times 1 = 2$
$\nu_\mu, \bar{\nu}_\mu$	< 0.6 eV		
$\nu_e, \bar{\nu}_e$	< 0.6 eV		

Of course, the story for neutrinos is more complicated. If they were massless, then this would be easy. However, if they have mass, then we need to decide on what type of mass - a *Majorana* or *Dirac* mass. For our purposes, the important piece here is that is neutrinos had a Majorana mass, they would be their own anti-particle, and so would not get an additional factor of 2 from that. If they are Dirac particles, then you would get an additional factor of 2, and neutrino measurements would be inconsistent with BBN. The way out would be to have half of these degrees decouple in the very early Universe and then dilute away. Determining the mass type is still an open question.

Adding up internal degrees of freedom, we find

- $g_b = 28$: photons(2), $W^{\pm}$ and Z (3x3), gluons (8x2) and Higgs (1)
- $g_f = 90$: quarks (6x12), charged leptons (3x4), neutrinos (3x2)

Putting this together, we get

$$g_* = g_b + \frac{7}{8} g_f = 106.75$$

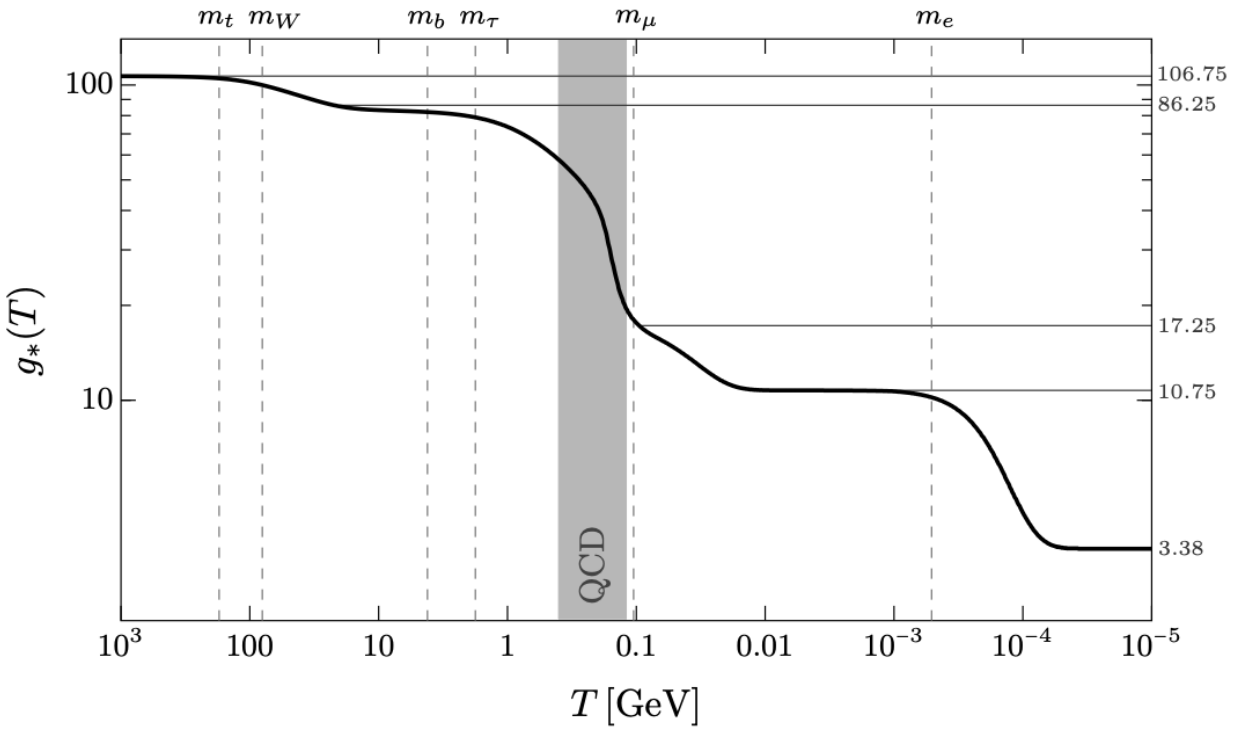


Figure 22.4.: Evolution of the effective number of relativistic species (Baumann)

As the temperature of the Universe decreases, various species go non-relativistic and drop out from g_*. This continues until the charm quark. At this point, the QCD phase transition takes place and the quarks get bound up into baryons and mesons, of which only the pions are light enough to contribute here. These are spin 0, and so carry a total of 3 internal degrees of freedom, resulting in a total of $g_* = 17.25$. Soon after this, muons and pions annihilate with g_* dropping to 10.75.

At this point, electrons and protons will annihilate, and the neutrinos will decouple. However, to fully understand that, we need to discuss the evolution of entropy.

22.5. Entropy

Let's start with the first law of thermodynamics (in the absence of a chemical potential)

$$TdS = dU + PdV$$

Rewriting this in terms of the entropy s and energy density ρ, we have

$$\begin{aligned} Td(sV) &= d(\rho V) + PdV \\ TVds + TsdV &= \rho dV + Vd\rho + PdV \\ (Ts - \rho - P)dV + V\left(T\frac{ds}{dT} - \frac{d\rho}{dT}\right)dT &= 0 \end{aligned}$$

Since this must be true for all variations dV and dT, we have

$$s = \frac{\rho + P}{T}$$

and

$$\frac{ds}{dT} = \frac{1}{T}\frac{d\rho}{dT}$$

Using the continuity equation

$$\frac{d\rho}{dt} = -3H(\rho + P) = -3HTs$$

we get

$$\frac{d(sa^3)}{dt} = 0$$

or the statement that the total entropy is conserved in the Universe, and $s \propto a^{-3}$.

We can also write out an expression for s

$$s = \sum_i \frac{\rho_i + P_i}{T_i} = \frac{2\pi^2}{45} g_{*,S}(T) T^3$$

where

$$g_{*,S}(T) \approx \sum_{i=\text{bosons}} g_i \left(\frac{T_i}{T}\right)^3 + \frac{7}{8} \sum_{i=\text{fermions}} g_i \left(\frac{T_i}{T}\right)^3$$

away from mass thresholds. Note the similarities and differences with g_*; if all species have a common temperature, then $g_* = g_{*,S}$.

A consequence of entropy conservation is

$$g_{*,S}(T) T^3 a^3 = \text{constant}$$

We can also plug this in to the Friedmann equation - here we just note that at about 1s, the Universe was $T = 1$ MeV, and evolves as $t^{-1/2}$ before that.

22.6. Counting States - II : $e^{\pm}$ annihilation and Neutrino Temperature

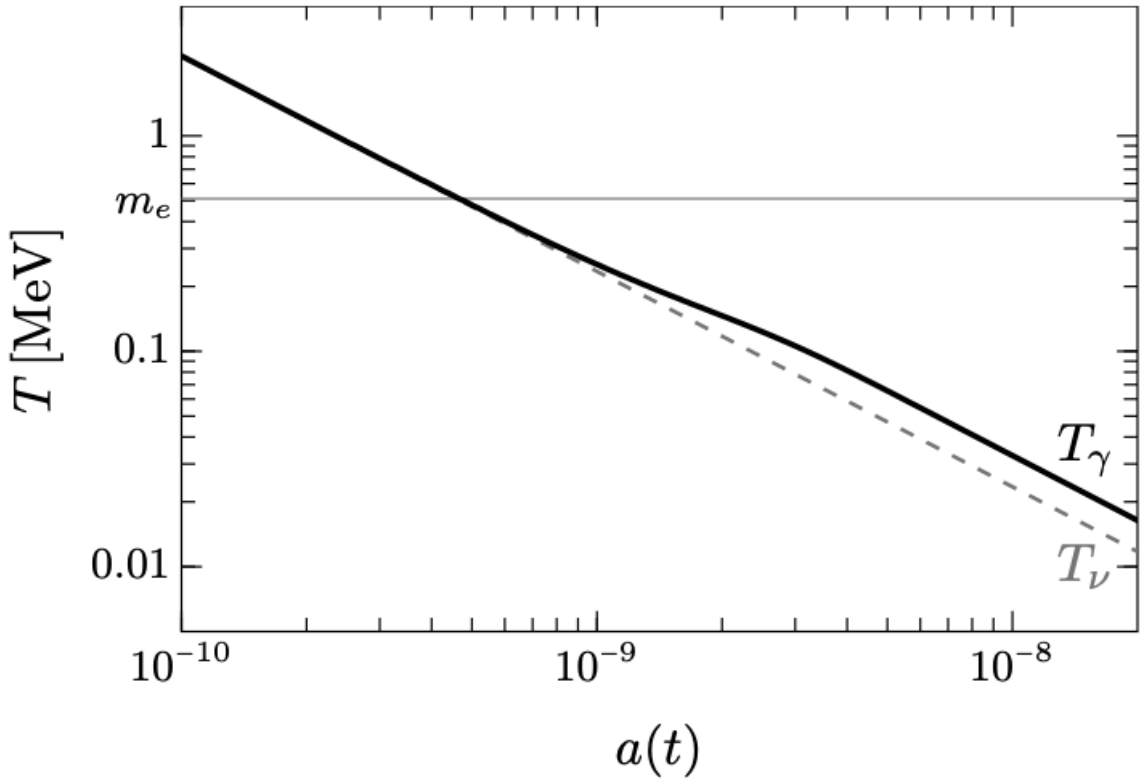
At 1 MeV, neutrinos decouple, and soon after, electron-positron annihilation takes place. The energy from $e^{\pm}$ annihilation gets deposited into the photons but not the neutrinos, raising the photon temperature relative to the neutrino temperature.

We can calculate this effect by using the conservation of entropy. We have

$$g_{*,S} = \begin{cases} 2 + \frac{7}{8} \times 4 = \frac{11}{2} & T > m_e \\ 2 & T < m_e \end{cases}$$

which raises the photon temperature by a factor of a factor of $(11/4)^{1/3}$. We therefore have

$$T_\nu = \left(\frac{4}{11}\right)^{1/3} T_\gamma \approx 1.9\text{K}$$

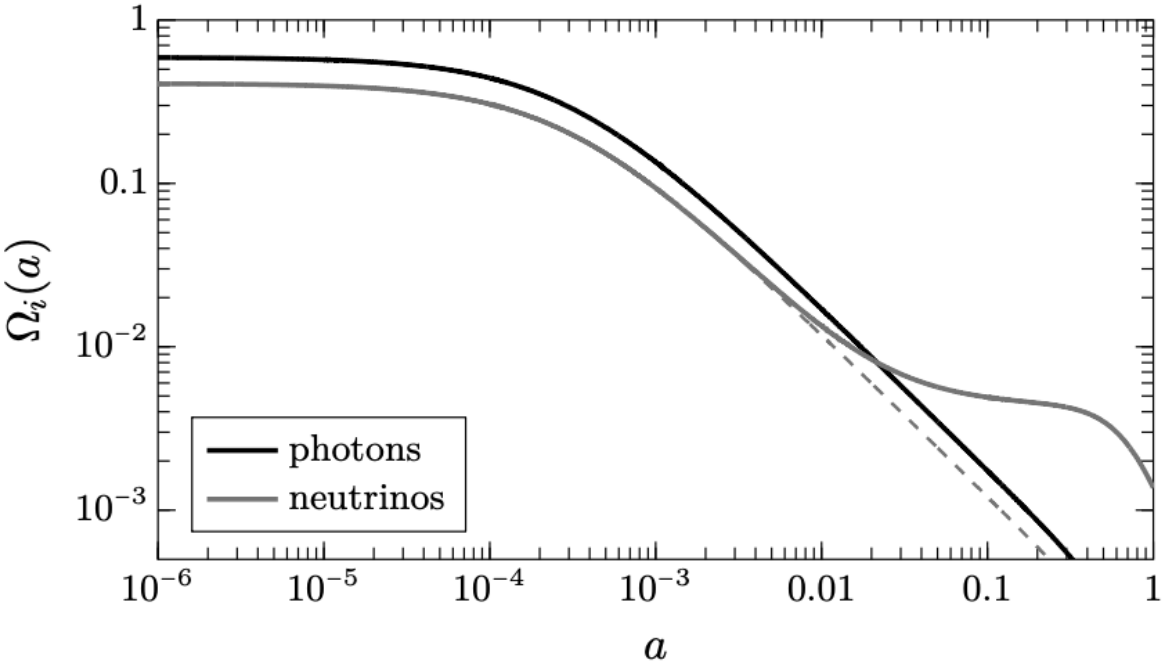


Furthermore, we can calculate the number of degrees of freedom, and we find

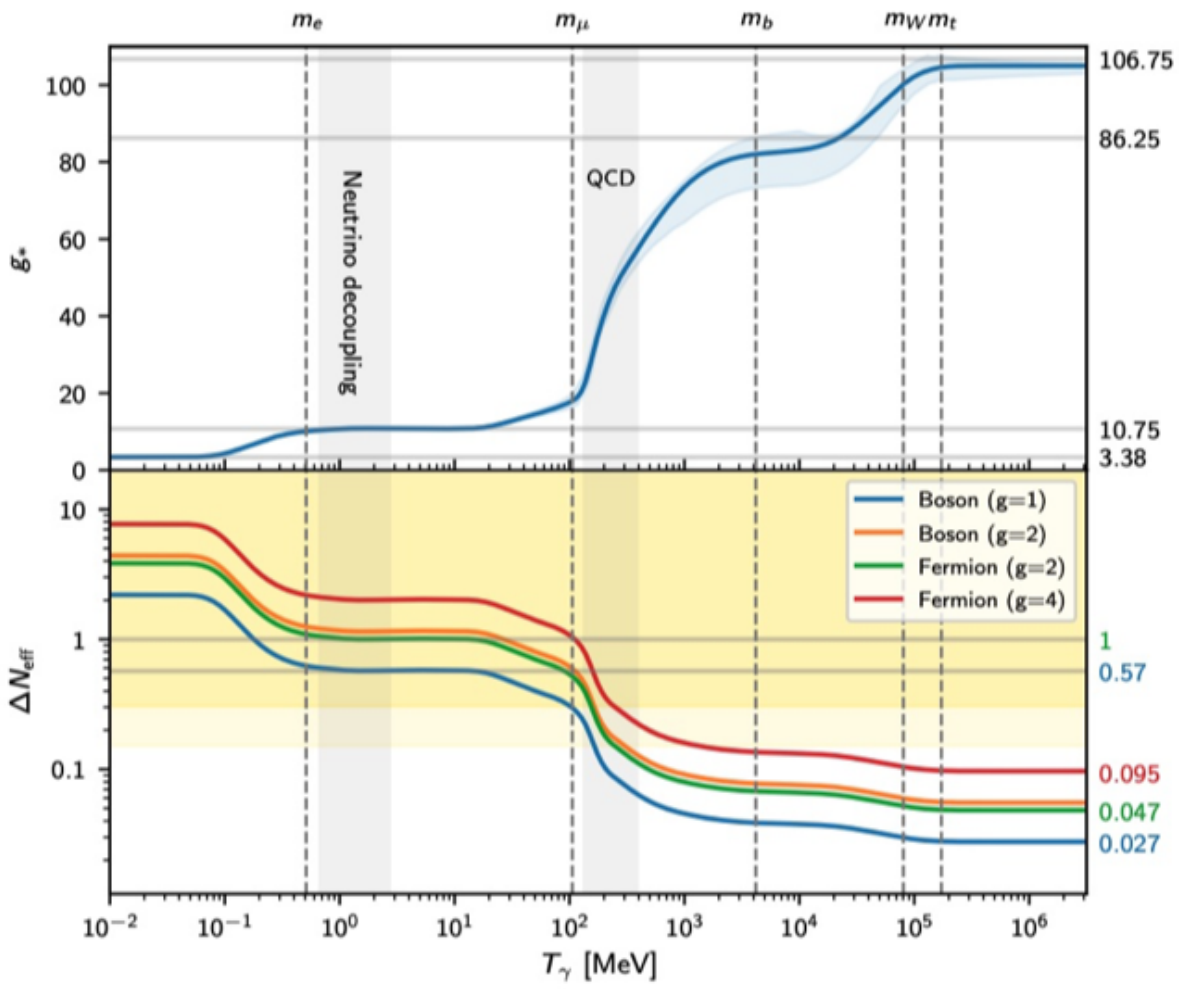
$$g_* = 2 + \frac{7}{8} \times 2N_{eff} \left(\frac{4}{11}\right)^{1/3} = 3.36$$

Note that the actual value of $N_{eff} = 3.046$, accounting for the fact that neutrino decoupling is not instantaneous and some energy from $e^{\pm}$ annihilation gets deposited into the neutrinos.

Finally, we can end with plotting the energy density of photons and neutrinos as a function of time. As we see, they largely track each other until low redshift, where the neutrinos go non-relativistic and behave like ordinary matter. But that's a different discussion.


1
0.1
10^{-2}
10^{-3}
$\Omega_i(a)$
photons
neutrinos
10^{-6}
10^{-5}
10^{-4}
10^{-3}
0.01
0.1
1
a

22.7. Additional Relativistic Species



Similar to the neutrino decoupling case, we have

$$T_X = \left(\frac{g_*(T_{dec,\nu})}{g_*(T_{dec,X})}\right)^{1/3} T_\nu = 0.465 \left(\frac{106.75}{g_*(T_{dec,X})}\right)^{1/3} T_\nu$$

and

$$\Delta N_{eff} = \frac{\rho_X}{\rho_\nu} = 0.027 g_{*,X} \left(\frac{106.75}{g_*(T_{dec,X})}\right)^{4/3}$$

The figure above, taken from the Planck 2018 cosmological parameters paper shows the $\Delta N_{\rm eff}$ contribution from different particle species as a function of when they decouple.

22.8. Probes

We'll consider three probes here : Big-Bang Nucleosynthesis, the CMB and baryon acoustic oscillations. A full treatment of these is beyond our scope here, but there are references below for

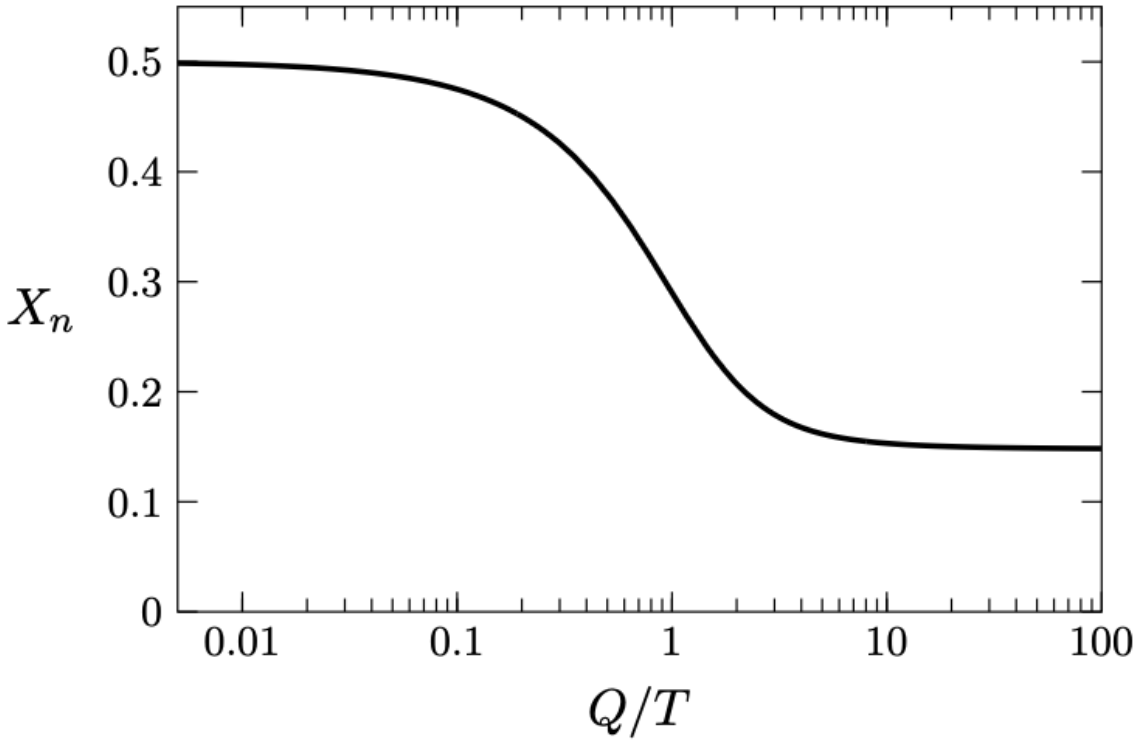
more details.

22.8.1. BBN

The abundance of elements - in particular, helium and deuterium are set by the neutron-proton ratio. We'll return to a more formal way to calculate this later, but protons and neutrons are coupled to each other by β and inverse-β decay:

$$n + \nu_e \leftrightarrow p^+ + e^-$$
$$n + e^+ \leftrightarrow p^+ + \bar{\nu}_e$$

While the weak interactions are relevant, the neutron to proton ratio is exponentially suppressed by the small mass difference between them and the neutron fraction rapidly falls. However, when the expansion rate is of the same order as the interaction cross-section, the neutron density freezes out, as the figure below shows.



These neutrons will either decay, or get bound into deuterium and then to helium. Since this residual abundance is set by the expansion rate, which is turn is set by the number of relativistic species, the observed helium and deuterium abundances can constrain N_{eff}.

The figures here are from Yeh, Shelton et al, 2022. BBN depends both on the expansion rate, but also the baryon to photon ratio η - the figure below shows the joint constraints on $N_{\rm eff}$ and η. The CMB is very effective at constraining η and so breaks much of the degeneracies.

A note on the cases presented. The CMB damping tail (as we'll discuss below) depends on the helium abundance Y_p. The standard CMB analyses use the BBN relationship to determine this

value, but one can marginalize out this value as well. We'll refer to this as the CMB+BBN case, on top of which one can add the measurements of helium and deuterium.

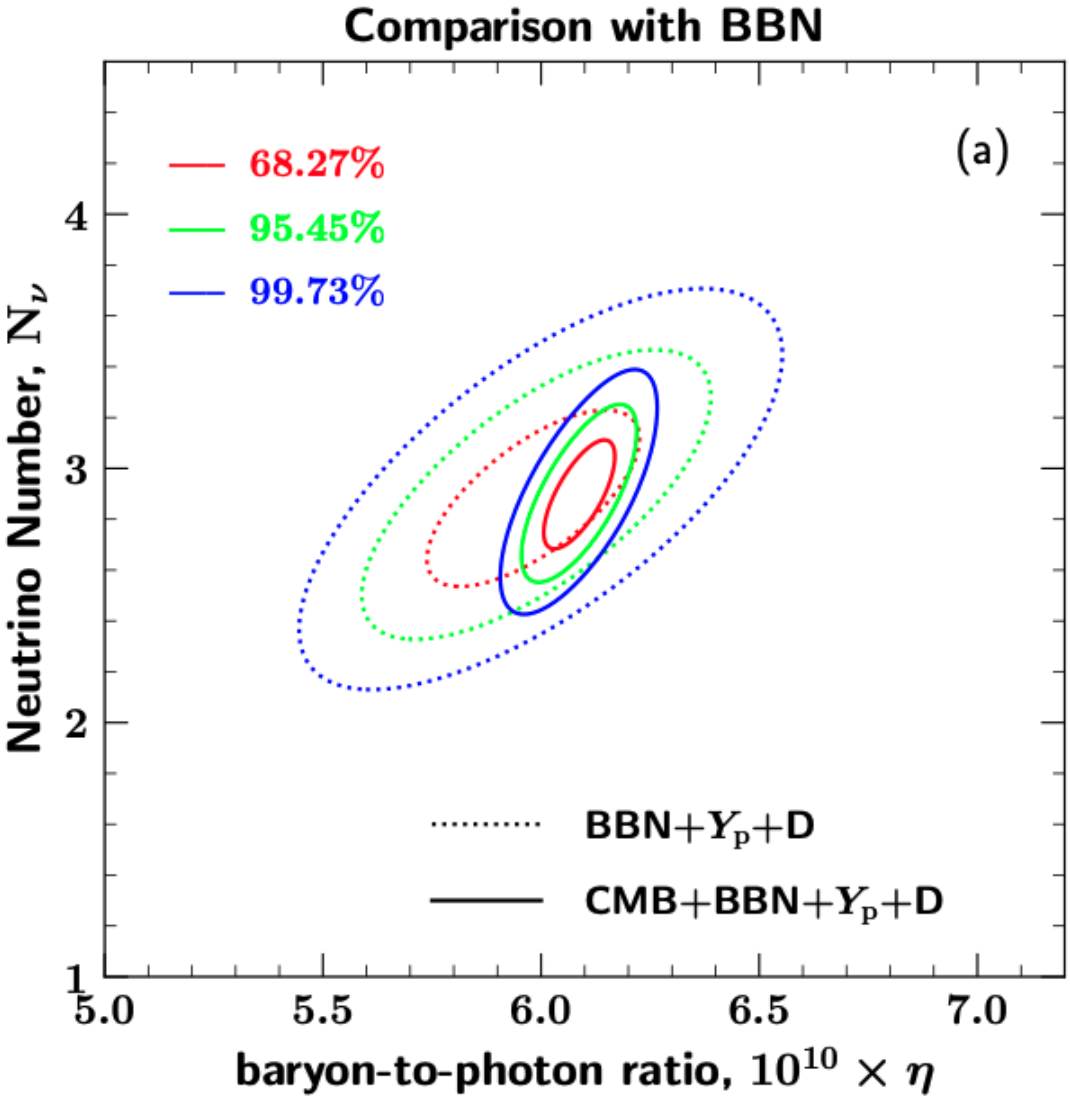
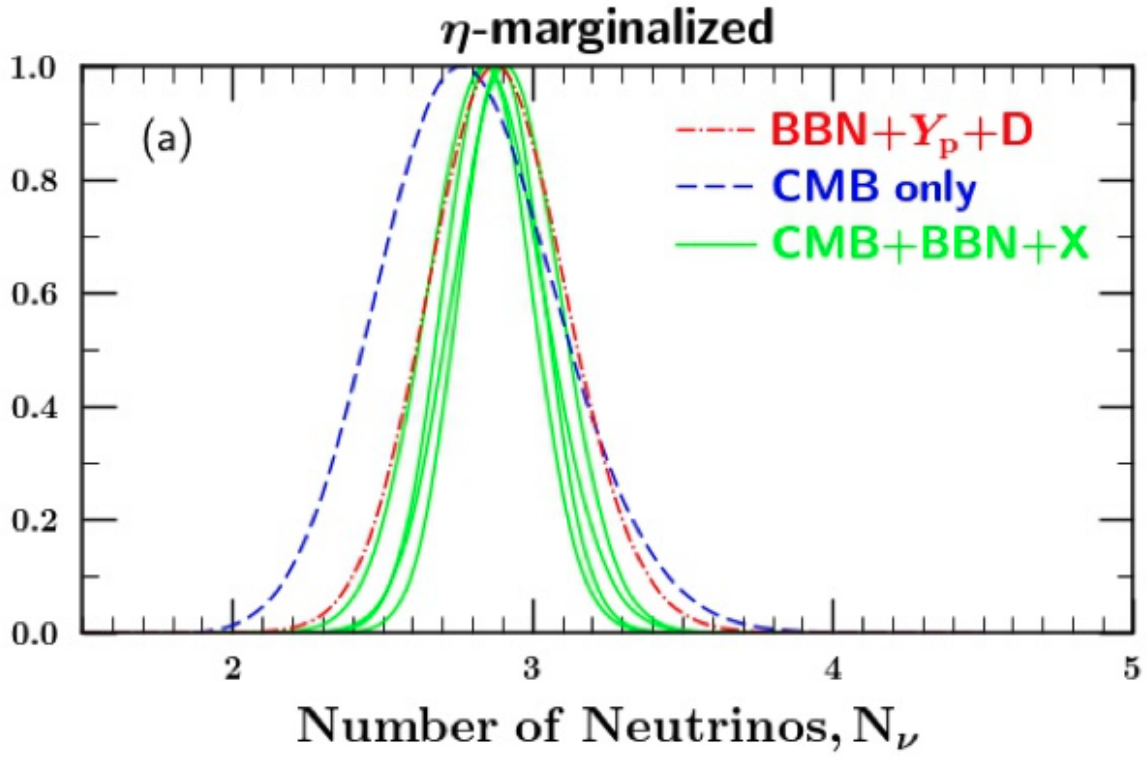
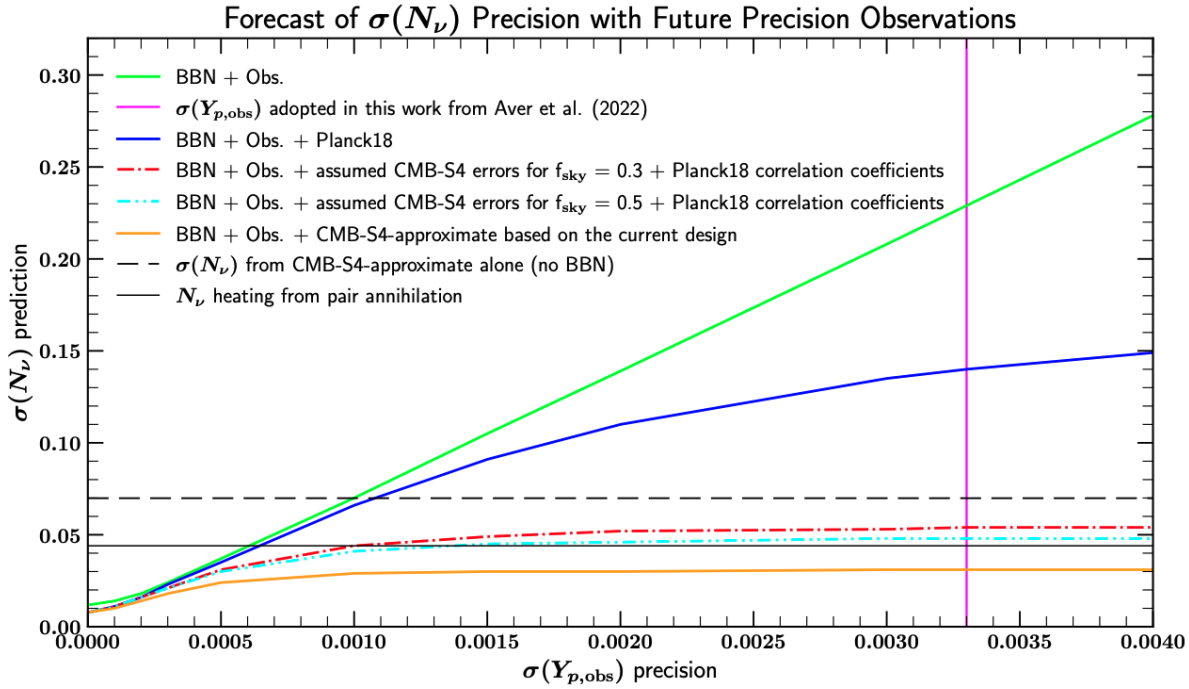


Figure 22.5.: Shelton et al, 2022

The figure below marginalizes out the value of η, X here is none, Y_p, D, and $Y_p + D$.



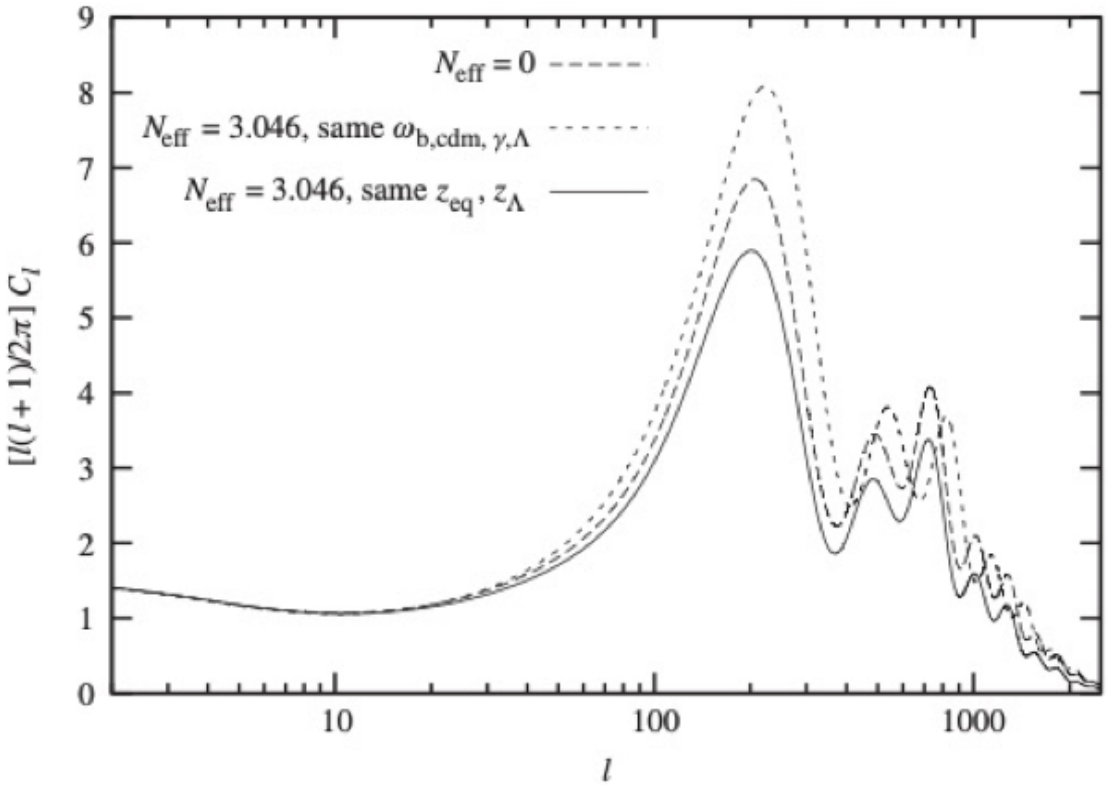
Finally, the figure below shows forecasts as a function of possible future measurements including improved measurements of the helium abundance. The target from $e^{\pm}$ is shown as a gray line.



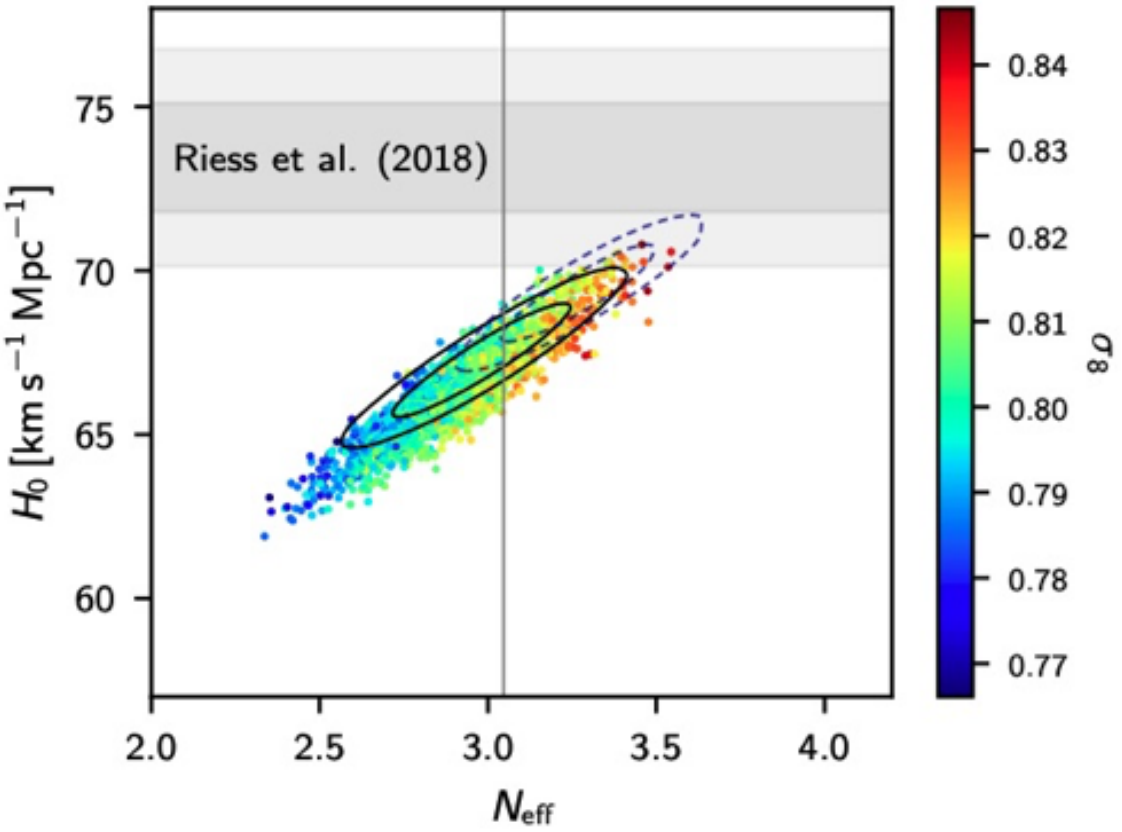
22.8.2. CMB

- Primary effect is through the epoch of matter radiation equality
- Secondary effects from damping
- Complicated degeneracies with other parameters.
- Phase shift gives a unique view onto neutrinos.

Since the neutrinos contribute non-negligibly to the radiation density of the Universe, they play a significant role in many of the features of the CMB, as this figure from Lesgourges et al, 2013 shows.



One can however introduce $N_{\rm eff}$ as a parameter into the model and constrain it simultaneously with other parameters, the result of which is shown below (from the Planck 2018 results).



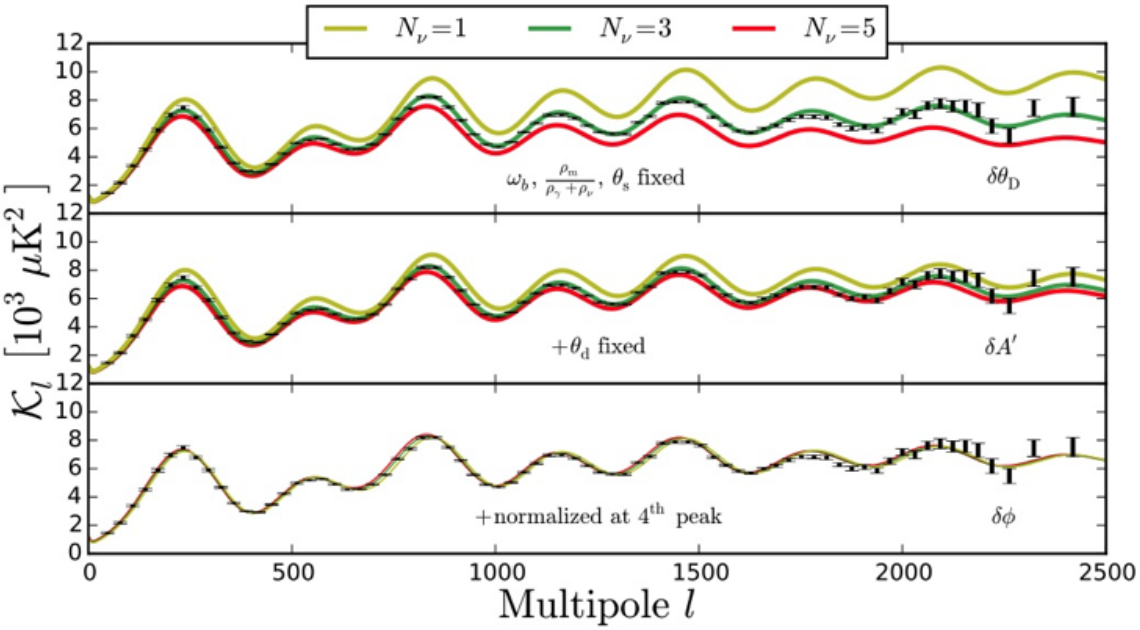
The overall results in that paper are

- $N_{\rm eff} = 3.00^{+0.57}_{-0.53}$ (95%, *Planck* TT+lowE),
- $N_{\rm eff} = 2.92^{+0.36}_{-0.37}$ (95%, *Planck* TT,TE,EE+lowE),
- $N_{\rm eff} = 3.11^{+0.44}_{-0.43}$ (95%, TT+lowE+lensing+BAO)
- $N_{\rm eff} = 2.99^{+0.34}_{-0.33}$ (95%, TT,TE,EE+lowE+lensing+BAO).

for different cuts through the data and combinations with external data sets.

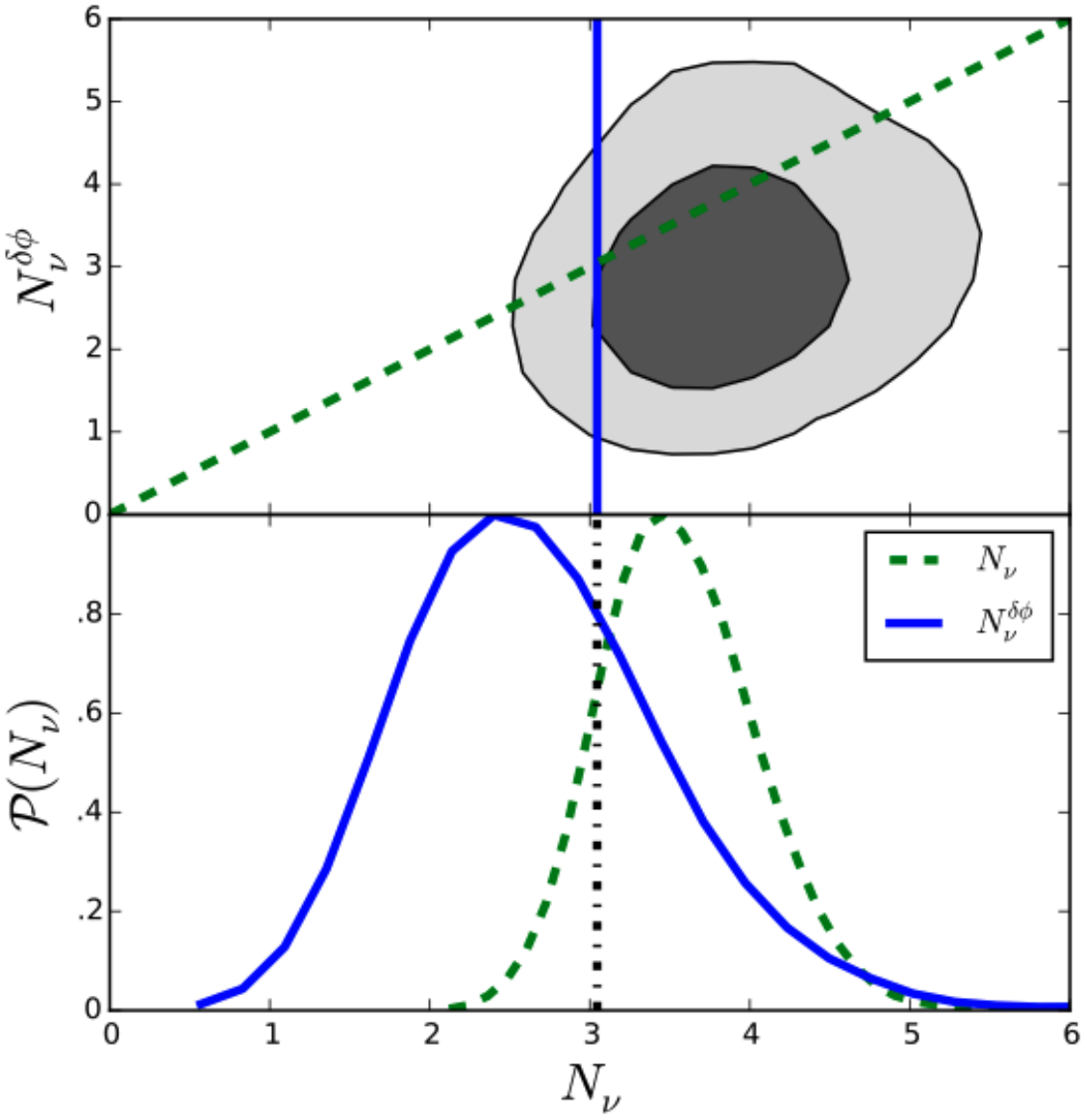
The two figures below are from Follin et al, 2015, where they detect a phase shift in the CMB acoustic peaks and argue that this is orthogonal from the other effects of neutrinos on the CMB. We reproduce their argument below.

They start by noting that the neutrino background plays a critical role in setting the sound horizon, which in turn sets the scale of the acoustic peaks in the CMB. However, this physical scale isn't directly measurable, since the distance to the CMB is uncertain due to dark energy in the early Universe (although low redshift measures of the expansion history can help break this degeneracy). Therefore the key observable in the CMB is the angular scale of the acoustic peaks, and this is kept constant in all panels and across all variations (as is the baryon density and baryon to photon ratio, both of which are equally well measured and insensitive to the number of neutrino species). The dominant effect is then the damping of the higher acoustic peaks, seen in the top panel of the figure below.



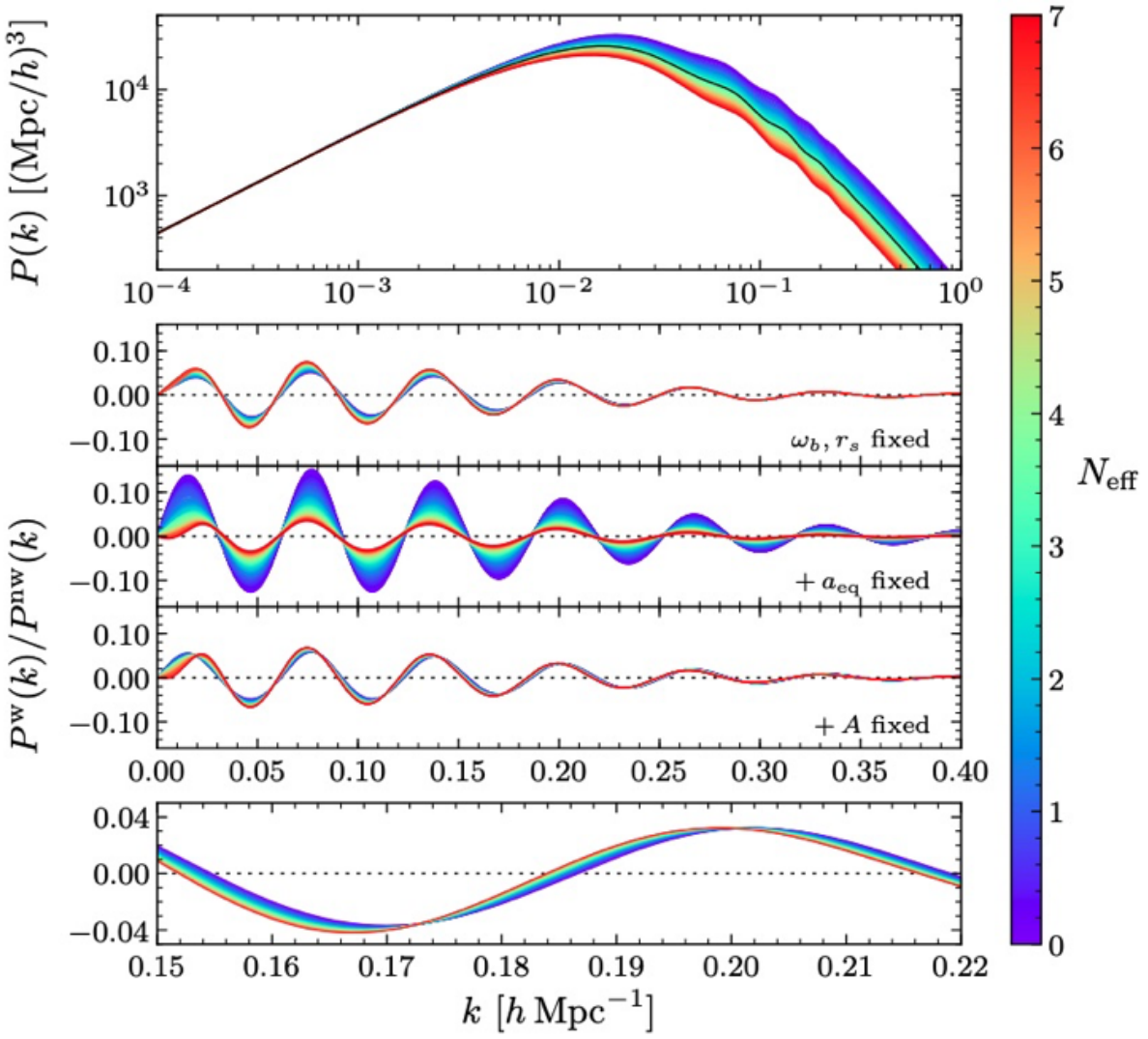
However, this damping scale can be adjusting by changing the helium fraction, giving rise to the second panel. The most noticeable effect is now the an amplitude shift, which is normalized away in the final panel. What we are left with is a phase shift in the acoustic peaks, which can be understood from how neutrino perturbations affect the gravitational potential and therefore the driving of the harmonic oscillator.

The key point is that neutrinos can change both the amplitude and phase of the acoustic oscillations. The amplitude can be absorbed by many other parameters, but the phase is much more robust. The figure below shows the first detection of this phase, where they allow the N_ν that drives the gross CMB effects to be separate from the N_ν that drives the phase. Note that these are both consistent with each other and the expected value that we previously calculated.

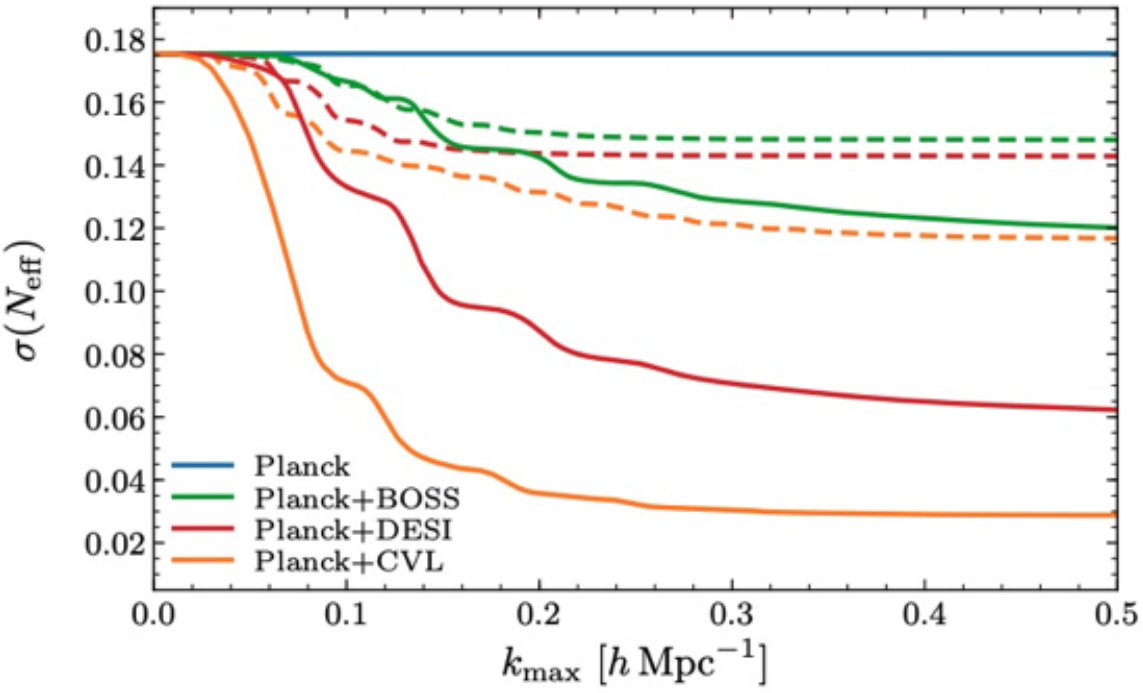


22.8.3. BAO

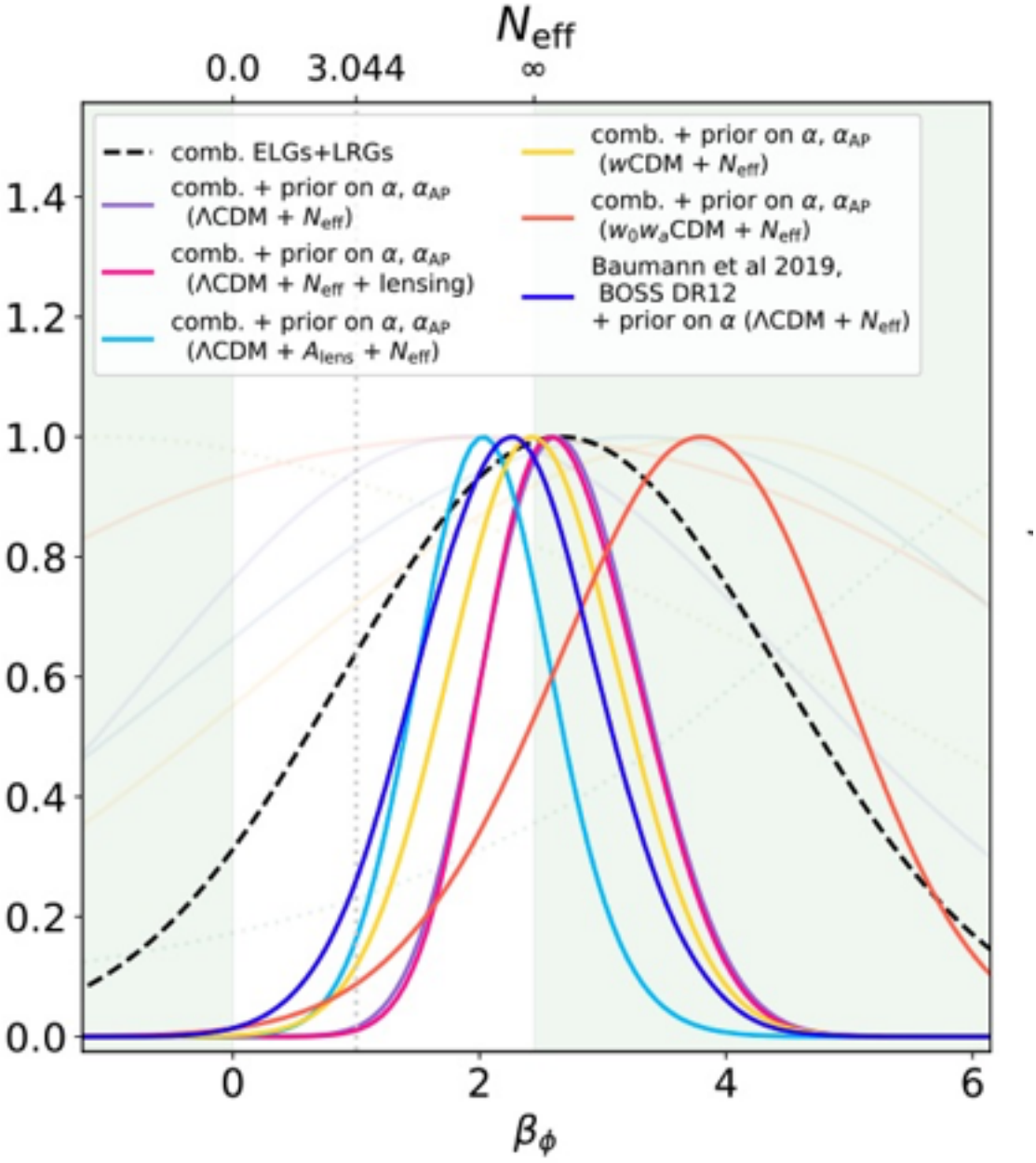
A similar effect can be seen in the galaxy power spectrum. The next two figures (from Baumann et al 2018) highlight the various effects. As with the CMB, there is a residual phase shift that survives even as other physical effects are kept fixed (or marginalized over). The figure below (Fig 3 in the paper) steps through these effects, similar to the CMB discussion above.



The figure below (Fig 13 in the paper) shows the forecasted sensitivity for both the BAO and galaxy power spectrum for different surveys as a function of the maximum k scale (the smallest scales that can be reliably used) considered. The dashed lines use just the BAO signal, while the solid line shows the information from the full power spectrum.



Finally, the figure below from Whitford et al, 2024 shows an initial measurement of this phase shift in the DESI BAO measurements.



22.9. References

- Baumann, **Cosmology**
- Lesgourges et al, **Neutrino Cosmology**
- Tsung-Han Yeh et al JCAP10(2022)046
- Follin et al, 1503.07863
- Baumann et al, 1712.08067
- Whitford et al, 2412.05990

23. The Boltzmann Equation and Dark Matter

The discussion follows Sec 3.2 in Baumann's textbook.

As we discussed earlier, in equilibrium, the number density of non-relativistic particles scales as

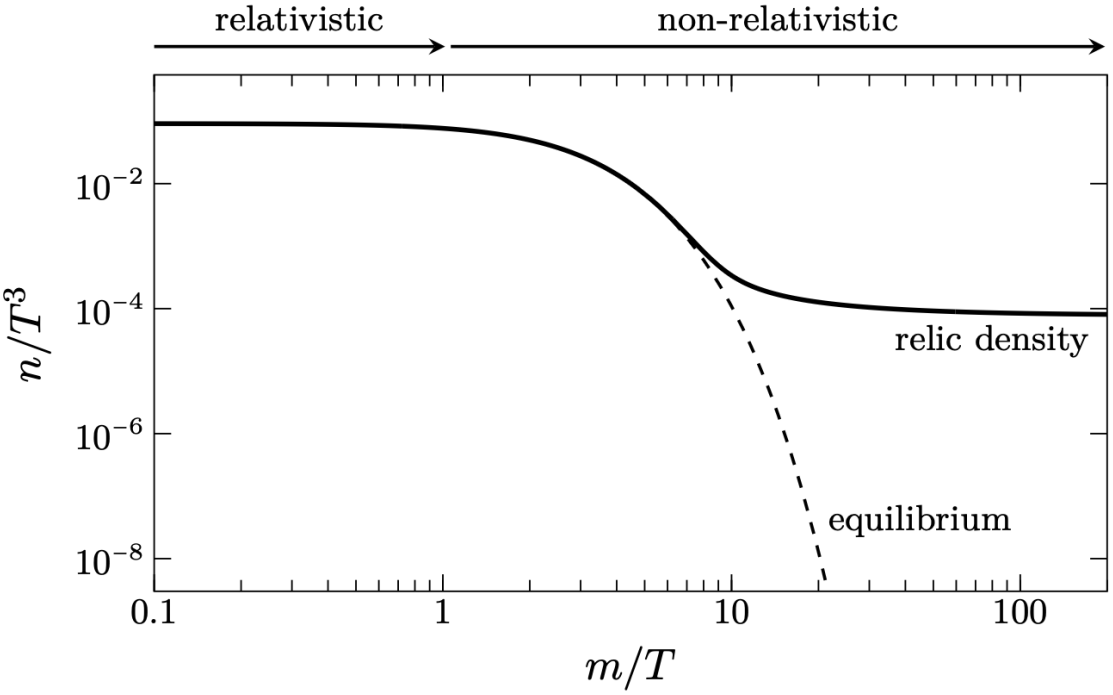
$$n \propto T^{3/2} e^{-m/T}$$

or is exponentially suppressed. This suppression can be simply thought of as annihilation removing particles and not enough energy in the thermal bath to produce new particles. However, this behavior is complicated by the expansion rate of the Universe. If the rate of the interactions keeping the particle in thermal equilibrium Γ becomes comparable to the Hubble expansion, then the annihilations get suppressed and the number density can freeze out (factoring out the cosmic expansion).

We can therefore summarize two regimes here:

- $\Gamma \gg H$: equilibrium
- $\Gamma \ll H$: Frozen out

The figure below captures this schematically.



This freezing out mechanism is responsible for a number of effects in cosmology

- Estimating the residual dark matter density for WIMPs and other thermally produced particles.

- Recombination
- The neutron density in BBN
- Baryogenesis

23.1. The Boltzmann Equation

These effects are captured by the Boltzmann equation

$$\frac{1}{a^3}\frac{d(na^3)}{dt} = C$$

The LHS just captures the dilution of the density of particles as the Universe expands, while the right hand side captures the effect of interactions.

To be more specific, let us consider two-body interactions

$$1 + 2 \leftrightarrow 3 + 4$$

and suppose we are interested in tracking particle 1. We then have

$$\frac{1}{a^3}\frac{d(na^3)}{dt} = -\alpha n_1 n_2 + \beta n_3 n_4$$

Using detailed balance, we estimate

$$\beta = \left(\frac{n_1 n_2}{n_3 n_4}\right)_{\rm eq} \alpha$$

which gives

$$\frac{1}{a^3}\frac{d(na^3)}{dt} = -\langle\sigma v\rangle \left[n_1 n_2 - \left(\frac{n_1 n_2}{n_3 n_4}\right)_{\rm eq} n_3 n_4\right]$$

where we have introduced the thermally averaged cross-section for α. While this is in terms of a physical density, it is convenient to write this in terms of the comoving density $N_i \equiv n_i/s$, where we use the entropy density to track the expansion. This yields (using the definition of the Hubble parameter)

$$\frac{d\ln N_1}{d\ln a} = -\frac{\Gamma_1}{H}\left[1 - \left(\frac{N_1 N_2}{N_3 N_4}\right)_{\rm eq}\frac{N_3 N_4}{N_1 N_2}\right]$$

where

$$\Gamma = \langle\sigma v\rangle n_2$$

This captures our intuition - when $\Gamma \gg H$, the system is in equilibrium, while in the opposite limit, $N_1 \to$ constant.

23.2. Thermal Relic Density of Dark Matter

Suppose we had a dark matter candidate X with the following reaction

$$X + \bar{X} \leftrightarrow l + \bar{l}$$

where l is just a light SM particle. We will assume that l is tightly coupled to the thermal bath, so the number density is in equilibrium. We also assume no asymmetry between X and $\bar{X}$, so that their number densities are equal. Finally, we assume that there are no additional annihilations taking place, so that $T \propto a^{-1}$ (see our previous discussion).

The Boltzmann equation is then

$$\frac{1}{a^3}\frac{d(na^3)}{dt} = -\langle\sigma v\rangle \left[n_X^2 - (n_X)_{\rm eq}^2\right]$$

Now for some convenient definitions

- $Y \equiv n_X/T^3$: this tracks the comoving number density
- A new time variable $x \equiv M_x/T$, where $dx/dt = Hx$.

Finally, the Friedmann equation can be written as

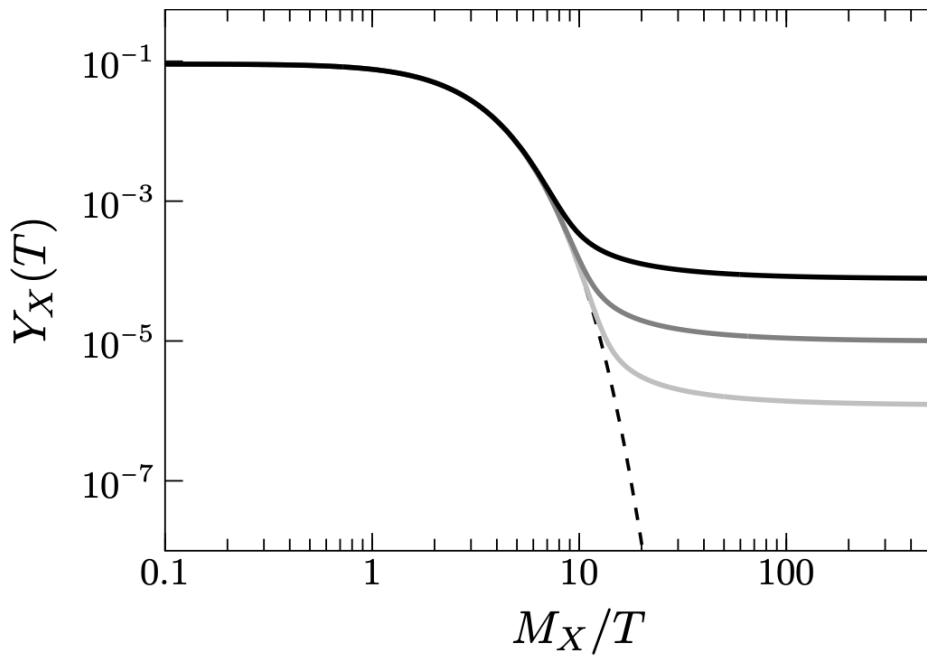
$$H = \frac{H(M_X)}{x^2}$$

Putting all this together and defining

$$\lambda \equiv \frac{\Gamma(M_X)}{H(M_X)} = \frac{M_X^3\langle\sigma v\rangle}{H(M_X)}$$

we get

$$\frac{dY}{dx} = -\frac{\lambda}{x^2}\left[Y^2 - Y_{\rm eq}^2\right]$$



The numerical solution for this gives

With a little more effort, we then get

$$\Omega_X \sim 0.1 \frac{x_f}{\sqrt{g_*(M_X)}} \frac{10^{-8}\text{GeV}^{-2}}{\langle \sigma v \rangle}$$

where $x_f \sim 10$ is the rough freeze out time. If we choose

$$\sqrt{\langle \sigma v \rangle} \sim 10^{-4}\text{GeV}^{-1} \sim 0.1\sqrt{G_F}$$

we get a dark matter density similar to what we observe.