Solving the Poisson equation

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

Once the density has been mapped onto the mesh the Poisson equation can be solved with two main classes of methods:

(a) Fourier techniques (convolution methods).
Here map to k-space, find solution in k-space and then map back to real space. Formally natural for periodic boundaries (natural situation in cosmological structure formation) but can be re-cast also for non-periodic boundaries.
Methods requires N^2 operations (N number of grid points) can be reduced to ~ NlogN using Fast Fourier Transform (FFT)

(b)Iterative methods (=find potential using a sequence of approximations with increasing accuracy convergence satisfactory).
Various methods of different accuracy, cost and convergence properties (Jacobi, Gauss-Seidel, multigrid..).

Provided the potential is found the acceleration/force can be determined by finite-differencing on the grid

$$\mathbf{a} = -\nabla\Phi.$$

In one direction the simplest estimate of the force on mesh would be:

$$a_x^{(i,j,k)} = -\frac{\Phi^{(i+1,j,k)} - \Phi^{(i-1,j,k)}}{2h},$$

This goes as $O(h^2)$. If one wants a more accurate determination then can account for more distant cells in the finite-difference estimate of the potential. For example:

$$a_x^{(i,j,k)} = -\frac{1}{2h}\left\{\frac{4}{3}\left[\Phi^{(i+1,j,k)} - \Phi^{(i-1,j,k)}\right] - \frac{1}{6}\left[\Phi^{(i+2,j,k)} - \Phi^{(i-2,j,k)}\right]\right\}$$

which now goes at $O(h^4)$. Same procedure can be followed for force computation in other directions y and z on cartesian mesh. Choice dictated by compromise between accuracy and speed of computation. In final step force on mesh interpolated back to particles (will cover briefly later)

Fourier techniques

These are powerful to solve PDEs whose solutions can be expressed as convolutions of more than one function (normally two) in real space. Then one maps to k-space and uses mathematical properties of Fourier transform. Formally periodic (wave-like solutions) but there are methods to adapt to non-periodic problems.
Numerically can be expensive but there are methods (eg FFT) that speed up calculation a lot by reducing the number of operations exploiting dependencies between terms in a series

Consider again Poisson equation

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

Simple general "ansatz" can be written by noting that potential at a point **x** must come from superpositions of contributions from individual mass elements (infinitesimally spaced if distribution is continuous), with the contribution of each mass element having the form of a potential from a point mass:

$$\Phi(\mathbf{x}) = -\int G \frac{\rho(\mathbf{x}')\,d\mathbf{x}'}{|\mathbf{x}-\mathbf{x}'|} .$$

This is recognized as a convolution integral of the type:

$$\boldsymbol{\Phi}(\mathbf{x}) = \int g(\mathbf{x}-\mathbf{x}')\,\boldsymbol{\rho}(\mathbf{x}')\,\mathrm{d}\mathbf{x}',$$

where $g(\mathbf{x}) = -\dfrac{G}{|\mathbf{x}|}$ is the Green's function of newtonian gravity

The convolution may be formally written as $\Phi = g * \rho$
Then using the *convolution theorem*, which states that the Fourier transform of the convolution of two functions is the product of their Fourier transforms, one can write:

$$\mathscr{F}(f \star g) = \mathscr{F}(f) \cdot \mathscr{F}(g),$$

where the curl *F* above denotes the Fourier transform. We have reduced the convolution integral into the product of two functions in Fourier space. The potential in k-space can then be determined by:

$$\hat{\boldsymbol{\Phi}}(\mathbf{k}) = \hat{g}(\mathbf{k}) \cdot \hat{\boldsymbol{\rho}}(\mathbf{k})$$

and one maps back into real space potential with $\boldsymbol{\Phi} = \mathscr{F}^{-1}\left[\mathscr{F}(g) \cdot \mathscr{F}(\boldsymbol{\rho})\right]$

The continuous Fourier transform

Just in the generic formula -> not yet in the mesh

In practice consider periodic box with size *L and r*epresent continuous density field $\boldsymbol{\rho}(\mathbf{x})$, i. e. before discretization with mesh is applied:

$$\rho(\mathbf{x}) = \sum_{\mathbf{k}} \rho_{\mathbf{k}} \, e^{i\mathbf{k}\mathbf{x}}$$

where sum over k-vectors extends over a discrete spectrum of wave vectors $\mathbf{k}$, with $\mathbf{k} \in 2\pi/L\ (n_1, n_2, n_3)$, n_1, n_2, n_3 being arbitrary integers. The allowed modes (=allowed values of $\mathbf{k}$) form thus an infinitely extended Cartesian grid with spacing $2\pi/L$. Reality constraints, for example for density, imply that $\rho_k = \rho^*_{-k}$ hence the modes are not all independent

The Fourier coefficients can be written as:

$$\rho_{\mathbf{k}} = \frac{1}{L^3} \int_V \rho(\mathbf{x}) \, e^{-i\mathbf{k}\mathbf{x}} d\mathbf{x}$$

for which orthogonality and closure relationships hold (see Lec. Notes)

Let us know turn back to the Poisson equation in Fourier space by expressing both the potential and the density as Fourier series;

$$\nabla^2 \left(\sum_{\mathbf{k}} \Phi_{\mathbf{k}} \, \mathrm{e}^{i\mathbf{kx}} \right) = 4\pi G \left(\sum_{\mathbf{k}} \rho_{\mathbf{k}} \, \mathrm{e}^{i\mathbf{kx}} \right) .$$

phi an rho in fourrier space

Using the expression of the laplacian in k-space this becomes:

we also have to replace the laplacian (gradient^2) then the equality above becomes this one:

$$\sum_{\mathbf{k}} \left(-\mathbf{k}^2 \Phi_{\mathbf{k}} \right) \mathrm{e}^{i\mathbf{kx}} = 4\pi G \sum_{\mathbf{k}} \rho_{\mathbf{k}} \, \mathrm{e}^{i\mathbf{kx}} .$$

The expression must hold for each of the Fourier modes separately which then implies (note "modes" are the individual terms in the sum labelled by **k)**

the following is the fundamental equation for any k

-> means that the potential and the density are given by a superposition of many waves

-> k ist just the frequency of the wave (or something)

$$\Phi_{\mathbf{k}} = -\frac{4\pi G}{\mathbf{k}^2} \rho_{\mathbf{k}} .$$

which then implies that in this case the *Green's function* is

$$g_{\mathbf{k}} = -\frac{4\pi G}{\mathbf{k}^2}$$

The discrete Fourier transform (DFT)

On the mesh the density function, or any other function, is not continuous, rather it is sampled by a discrete set of points (eg by CIC) In this case thus we have to consider discrete series, running on the set of *N* mesh points, rather than integrals, to represent the functions in k-space. Assuming *N* equally spaced mesh points the positions are:

$$\mathbf{x_p} = \frac{L}{N}\begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \quad \text{where } p_1, p_2, p_3 \in \{0, 1, \ldots, N-1\}$$

x_p is the position of any grid point

With the transformation $\mathrm{d}^3\mathbf{x} \to (L/N)^3$ the Fourier integral becomes:

$$\rho_{\mathbf{k}} = \frac{1}{N^3}\sum_{\mathbf{p}} \rho_{\mathbf{p}}\, \mathrm{e}^{-i\mathbf{k}\mathbf{x}_{\mathbf{p}}}.$$

Note the series is finite (N finite integer = number of cells in the mesh = *resolution of the mesh*). Because of periodicity it is invariant under the transformation **k** --> **k** + N x 2π/L

We can arbitrarily select as reference set of modes $\mathbf{k}_\mathbf{l}$ such that

$$\mathbf{k}_\mathbf{l} = \frac{2\pi}{L}\begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} \quad \text{where } l_1, l_2, l_3 \in \{0, 1, \ldots, N-1\}.$$

and the reconstruction of ρ via the Fourier series becomes thus a finite sum over the N^3 modes:

$$\hat{\rho}_\mathbf{l} = \frac{1}{N^3}\sum_\mathbf{p} \rho_\mathbf{p}\, \mathrm{e}^{-i\frac{2\pi}{N}\mathbf{lp}}$$

and its inverse

$$\rho_\mathbf{p} = \sum_\mathbf{l} \hat{\rho}_\mathbf{l}\, \mathrm{e}^{i\frac{2\pi}{N}\mathbf{lp}}.$$

To label the frequency values, $\mathbf{k} = (2\pi/L) \cdot \mathbf{l}$, one often conventionally uses the set $l \in \{-N/2, \ldots, -1, 0, 1, \ldots, \frac{N}{2} - 1\}$ instead of $l \in \{0, 1, \ldots, N-1\}$, which is always possible because shifting l by multiples of N does not change anything as this yields only a 2π phase factor. With this convention, the occurrence of both negative and positive frequencies is made more explicit, and they are arranged quasi-symmetrically in a box in $\mathbf{k}$-space centered on $\mathbf{k} = (0,0,0)$. The box extends out to

$$k_{\max} = \frac{N}{2}\frac{2\pi}{L}, \tag{124}$$

When we use a library to compute this
-> it may ommit the N factor in such calculations
-> needs to be added again later

There are various useful mathematical properties of the transforms (eg Percival's theorem on the norms, see Lecture notes), which are linear invertible maps. There are conventions also for DFT computer libraries, eg normally they omit N factors

can represent the forward transform of any complex number as:

$$y_k = \sum_{j=0}^{N-1} x_j e^{-i\frac{2\pi}{N} j \cdot k}.$$

and the backward transform is:

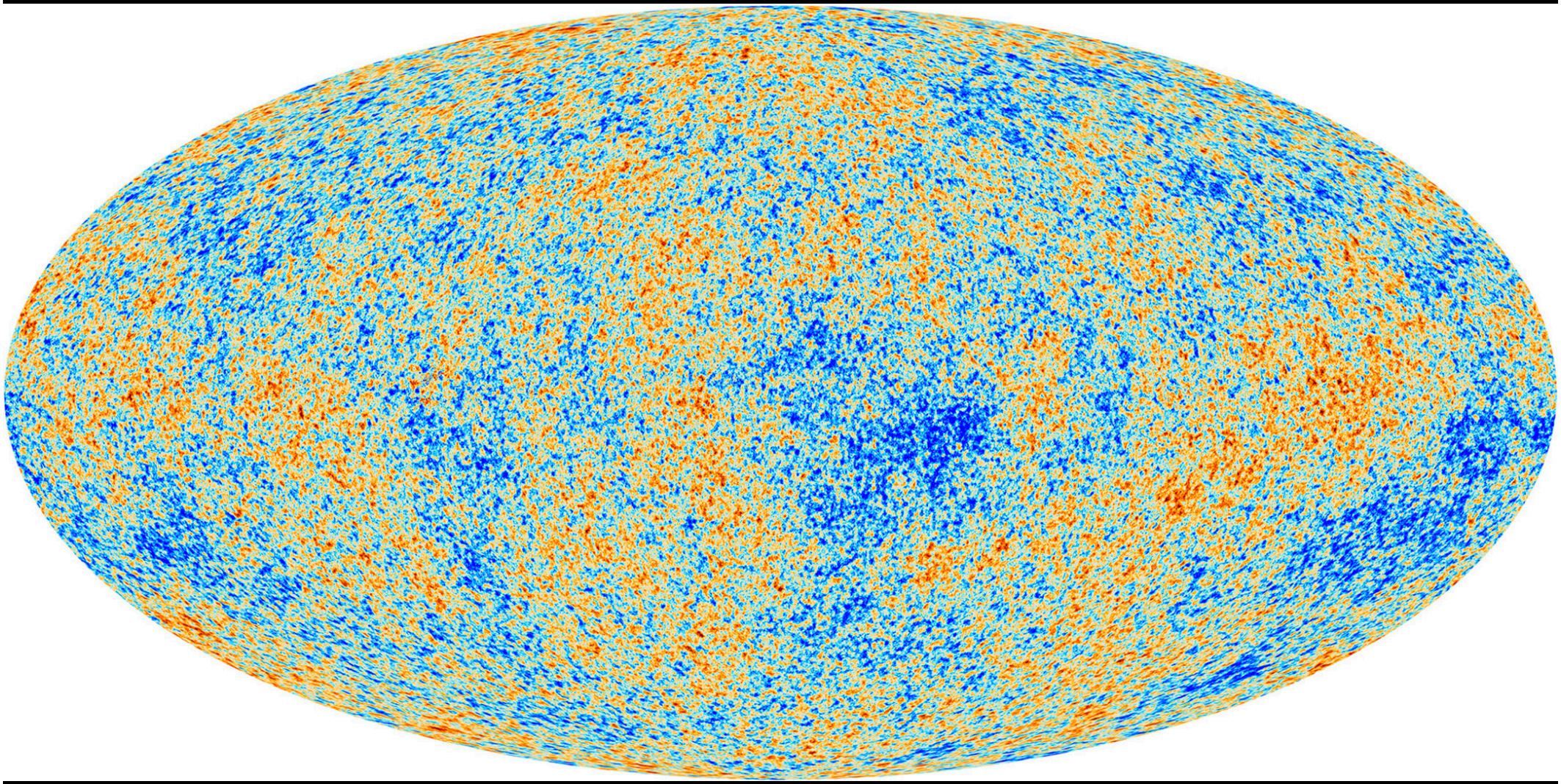
$$y_k = \sum_{j=0}^{N-1} x_j e^{i\frac{2\pi}{N} j \cdot k}.$$

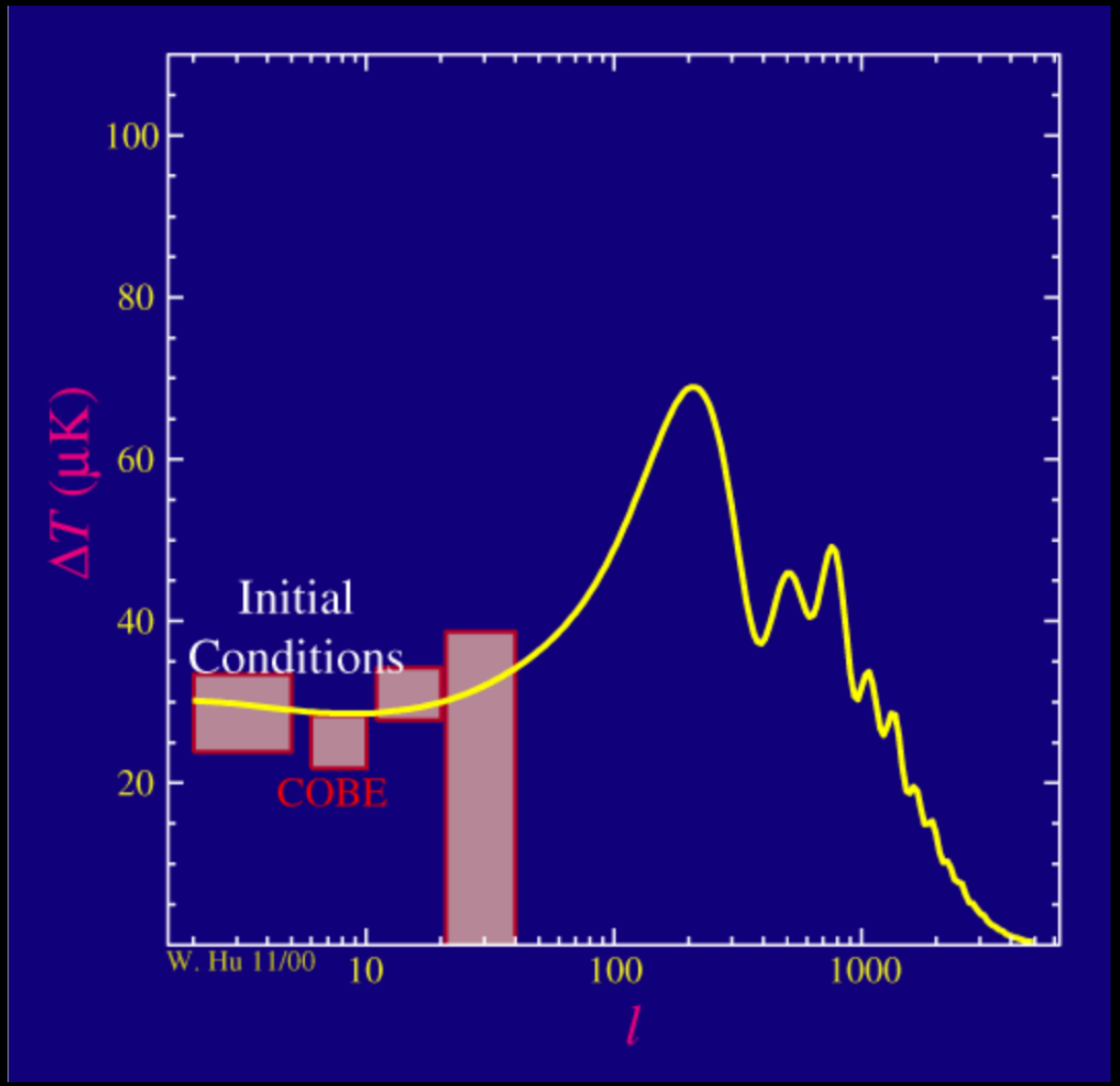
Written this way the transform is nicely symmetric (only difference in forward and backward transform is sign of exponent) but note that $\mathscr{F}^{-1}(\mathscr{F}(\mathbf{x})) = N\mathbf{x}$ hence one should now divide by N.

Nyquist frequency = maximum frequency that can be sampled correctly

$$k_{\max} = \frac{N}{2}\frac{2\pi}{L}.$$

Anisotropies in cosmic microwave background: the signature of early matter/energy perturbations (on microKelvin scale, while T ~ 2.73 K everywhere)



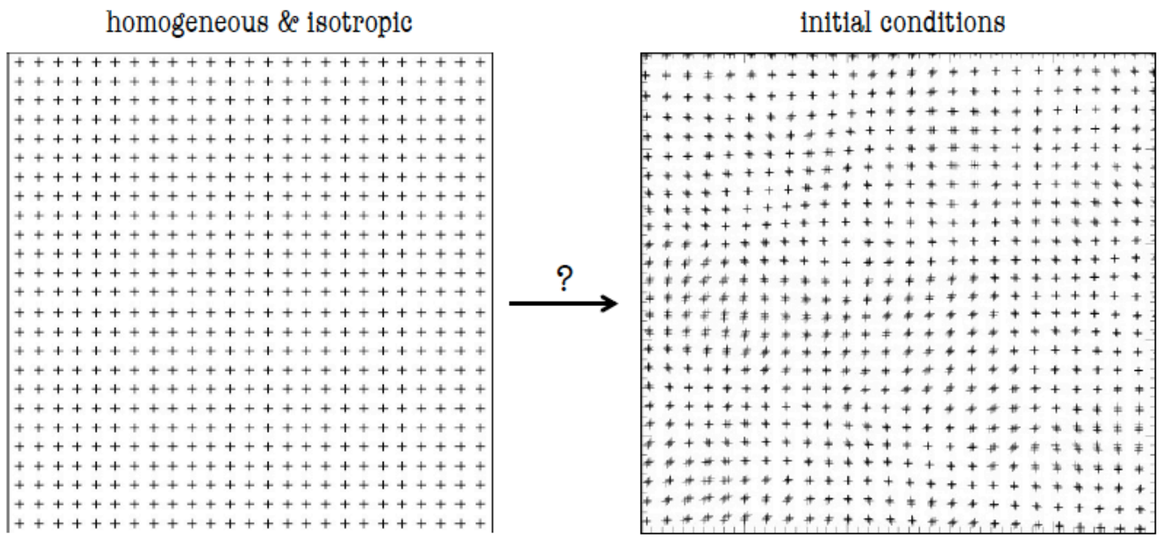

100
80
60
40
20
ΔT (μK)
Initial
Conditions
COBE
W. Hu 11/00
10
100
1000
l

Example: Initial Conditions for Cosmological Simulations

Assumption is matter and energy density start early as the superposition of wave-like perturbations on top of a uniform homegeneous density background. If there is no "special location" in the Universe then can just sample a small volume and assume periodic boundaries ---> ideal case for DFT!

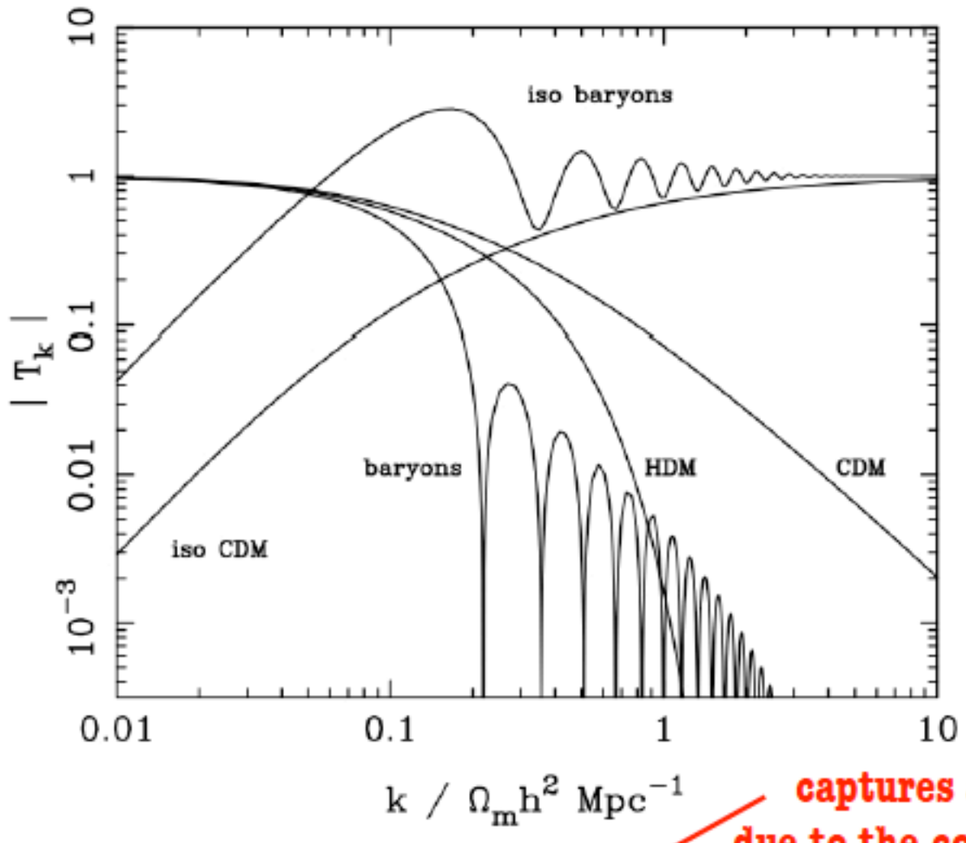
DFT of density perturbation field $\delta(\vec{x}) = \sum \hat{\delta}(\vec{k}) e^{-i\vec{k}\cdot\vec{x}}$

$P(k) = \left\langle \left| \hat{\delta}(\vec{k}) \right|^2 \right\rangle_{|\vec{k}|=k}$ Power spectrum in k-space



$$P_i(k) = Ak^n \qquad \text{with } n = 1 \text{ (Harrision - Zeldovich spectrum)}$$

Initial power spectrum from inflation

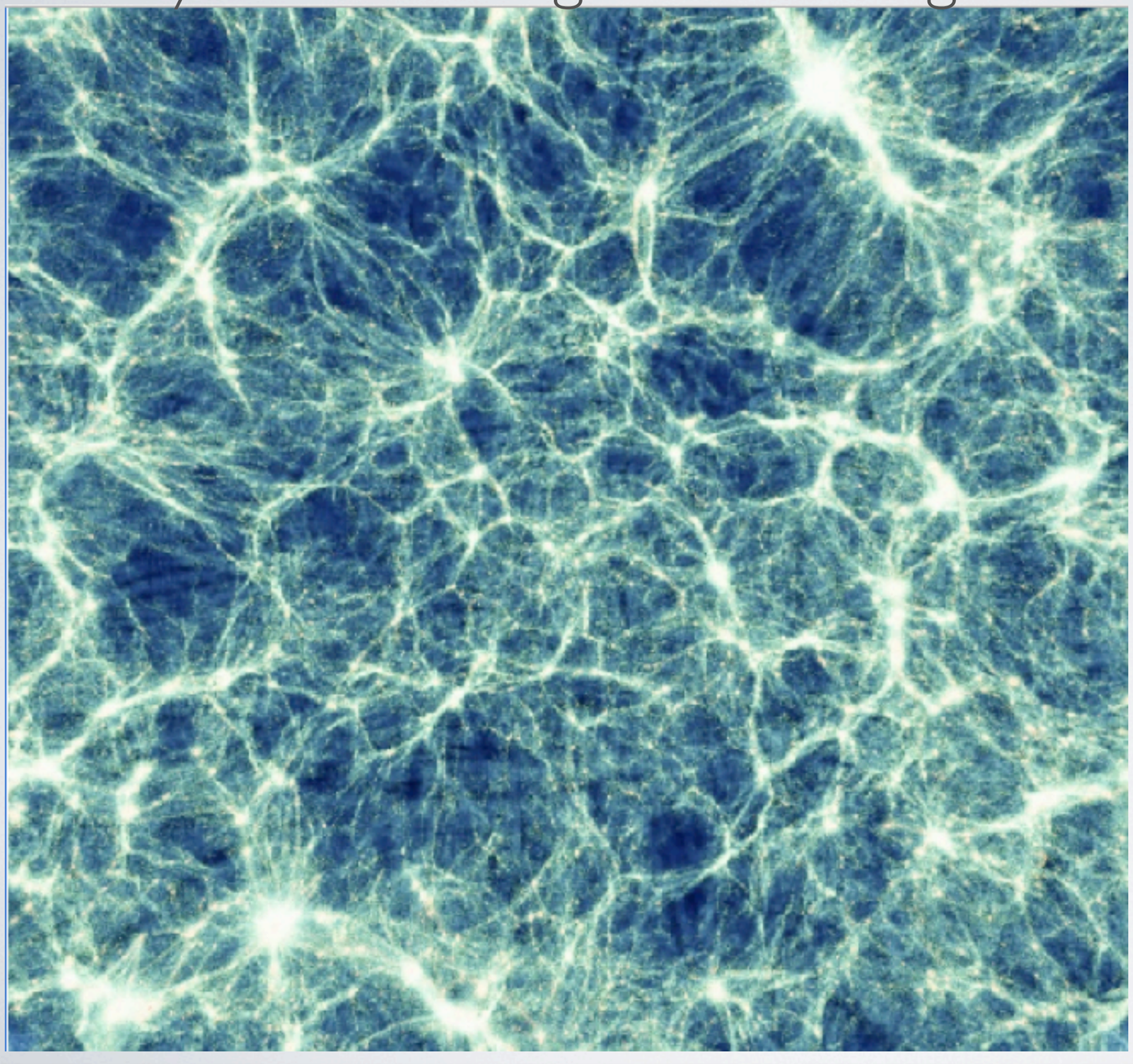


captures all the complicated physics due to the coupling of radiation and matter

$$P(k) = T^2(k)P_i(k)$$

Power spectrum after matter and radiation have decoupled (age of Univ. ~ 300.000 yr) is what is sampled on the mesh at t=0 in a simulation (the transfer function T(k) given by theory)

In the case of collisionless cold dark matter one solves then the Poisson equation for the initial perturbed density field and begins time integration....after some time:

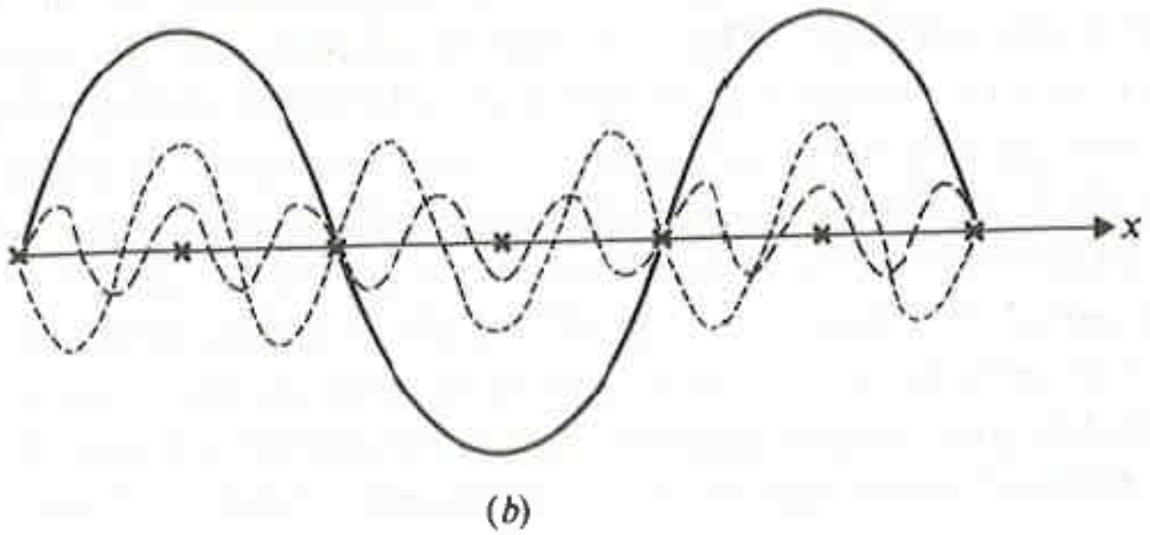


structure develops through gravitational collapse

Sampling theorem (DFT on mesh)

States that, in order to represent a function with characteristic frequency *fc* the sampling rate has to be at least double of such frequency. This in practice means the mesh has to have a resolution of at least half the wavelength of the function that one wants to represent, namely at least equal to the Nyquist frequency. (Resolution here is how we call the distance between two mesh points).

If the resolution of the mesh is below this requirement then one has the problems of *aliasing*, namely the effect of unsampled waves present in the DFT that represents the function, whose amplitudes are transferred to lower resolved frequencies and compromise the result of the DFT itself. The unsampled waves should be filtered out of the sum by truncating at the Nyquist frequency



If mesh points (crosses) correspond to maxima, minima or zeroes for both low-order harmonic and higher frequency aliases, then amplitudes are increased (solid line wave in (c))

If mesh points are such that eg one or more of the aliases are out-of -phase, then amplitudes are decreased (dashed line wave in (c))

Fast Fourier Transform (FFT)

Computing DFT requires to compute N terms in the Fourier series (N multiplications), and sum them up, *for each of the numbers* ρ_k. This means the computation goes as $O(N^2)$. This is a lot - today cosmological simulations for example consider up to 10^{11} grid points on large volumes in order to do "precision cosmology" (interpret latest data from eg Cosmic Microwave Background Experiments)--> with direct DFT generating initial conditions would be impossible!

The Fast Fourier Transform (FFT), invented in the 60s, is a method to reduce the order of the calculation by splitting the individual sums recursively so that one has to compute a number of terms much lower then *N* for each of the *N* numbers. The calculation, if N can be expressed as power in base 2, reduces to $O(N\log_2 N)$

Base example: replace a N-point DFT with the sum of many 2 point DFTs