An introduction to numerical cosmology

R. Valdarnini (SISSA)

Introduction

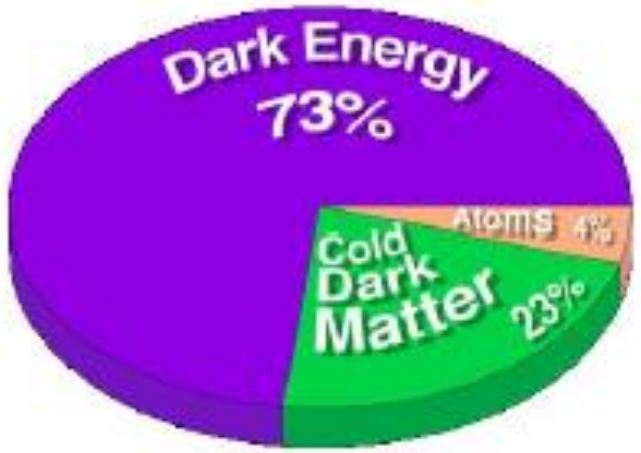
Over the last decades a growing number of observations have put in evidence that the galaxy distribution is not random but on large scales $(\geq 10 Mpc)$ is arranged in a web of sheet-like structures, with voids between them and filaments at the intersections.The size of these structures can be as large as

$\approx 50 - 100 Mpc$

12h
16h
8h
20h
4h
0h
0.02
0.04
0.06
0.08
0.10
0.12
0.14
Redshift z

There is now wide consensus that the present matter density budget can be written as follows :

baryon ~ 5% dark matter ~ 23 % dark energy ~ 72 %



where dark matter is an unknown form of matter (not baryonic) which interacts only gravitationally and dark energy is a vacuum energy density

The fundamental paradigm to describe the formation and evolution of structures in the Universe is that structure formation is driven by gravity within the following physical cosmology framework :

- The Universe is described by the General Theory of Relativity (GRT)

$$G_{ab} + \Lambda_{ab} = 8\pi G T_{ab}$$

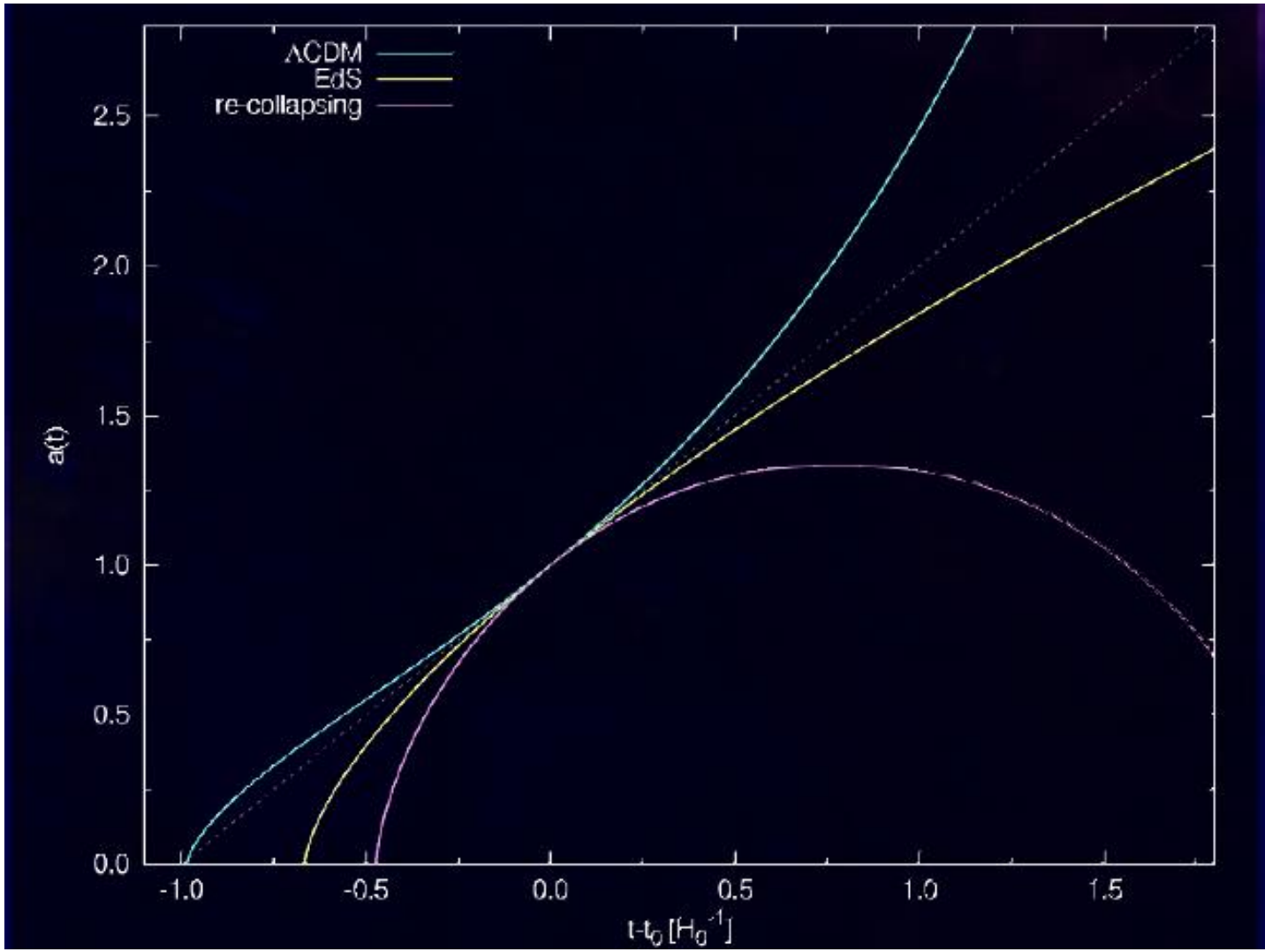
- The Universe is homogeneous and isotropic

$$ds^2 = -dt^2 + a^2(t)\left[\frac{dr^2}{1-kr^2} + r^2(d\theta^2 + sin^2\theta d\phi^2)\right]$$

- Friedmann's equations describe the Universe evolution

$$\left(\frac{\dot{a}}{a}\right)^2 \equiv H^2 = H_0^2\left[\frac{\Omega_r}{a^4} + \frac{\Omega_m}{a^3} + \frac{\Omega_k}{a^2} + \Omega_\Lambda\right]$$

The expansion radius a(t) vs time for different models in a FRW cosmology



NUMERICAL COSMOLOGY

- Structure formation is a complex non-linear phenomenon for which analytical methods are not sufficient. It is then necessary the use of numerical simulations.

BASIC APPROXIMATIONS

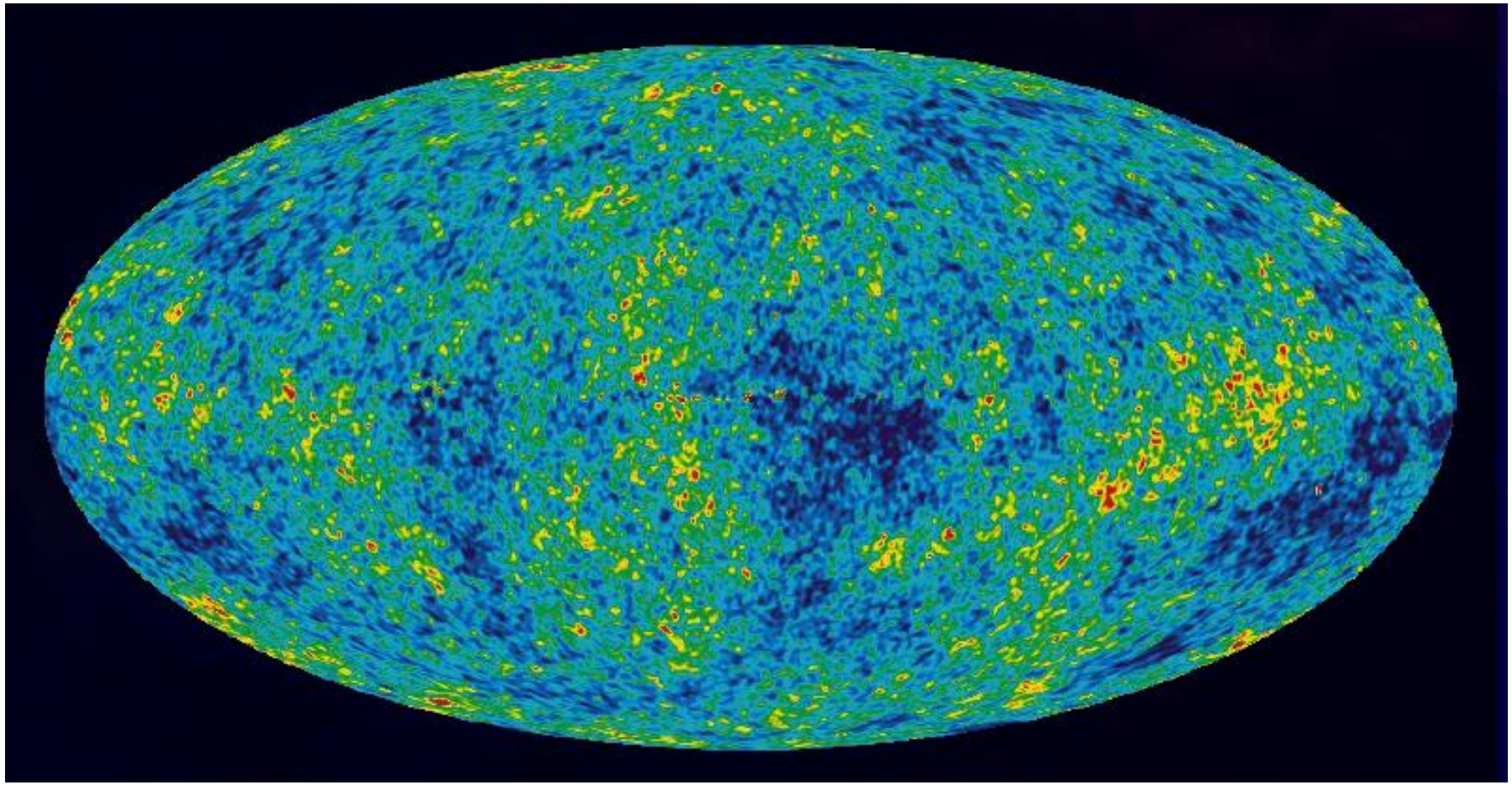
- On large scale the universe is homogeneous and isotropic
- On small scales the Newtonian limit of GRT is a good approximation

Therefore we can assume

- Classical physics in an expanding universe to describe the evolution of the components:
  - Dark energy — expansion a(t)
  - Dark matter — collisionless , only gravity
  - Baryons — Hydrodynamic, gravity

plus cooling, star formation, SN feedback, magnetic fields etc.

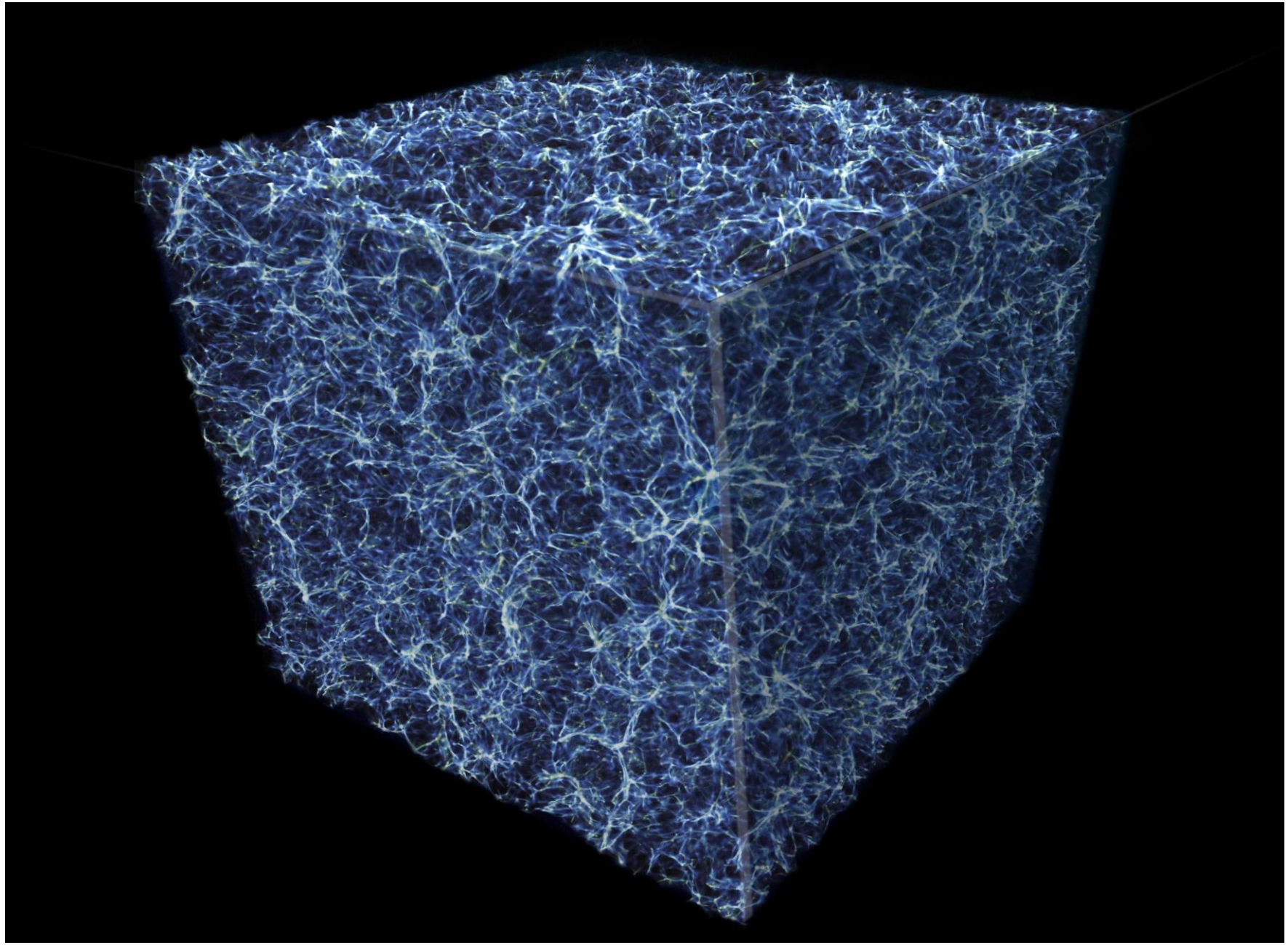
The homegeneity and isotropy assumption : the fluctuations in the CMB map are in one part in 10^5



An overview of a numerical simulation :

- General framework (cosmological model)
- Set-up initial conditions (box size , number of particles….)
- Evolve gravity (solve the N-body problem)
- Evolve gas (numerical method to solve the Euler equations)
- Analyse results

and the final results should look like this …..



Generation of the Initial Conditions

- We assume that at early epochs the density field is a random Gaussian process with power spectrum P(k) – This completely specifies the initial conditions

- The shape of P(k) depends on the model under consideration

- A random realization of the spectrum P(k) is done by assigning random amplitudes and phases to the modes $\vec{k}$ of the simulation cube

- Particles are initially arranged in a uniform lattice and their displacements and velocities are computed according to the Zel'dovich approximation

The shape of the standard ΛCDM power spectrum:

$$P(k) = A\frac{k^n}{\{1+\left[ak/\Gamma + (bk/\Gamma)^{3/2} + (ck/\Gamma)^2\right]^\nu\}^{1/\nu}}$$

$$n = 0$$

$$\Gamma = 0.21$$

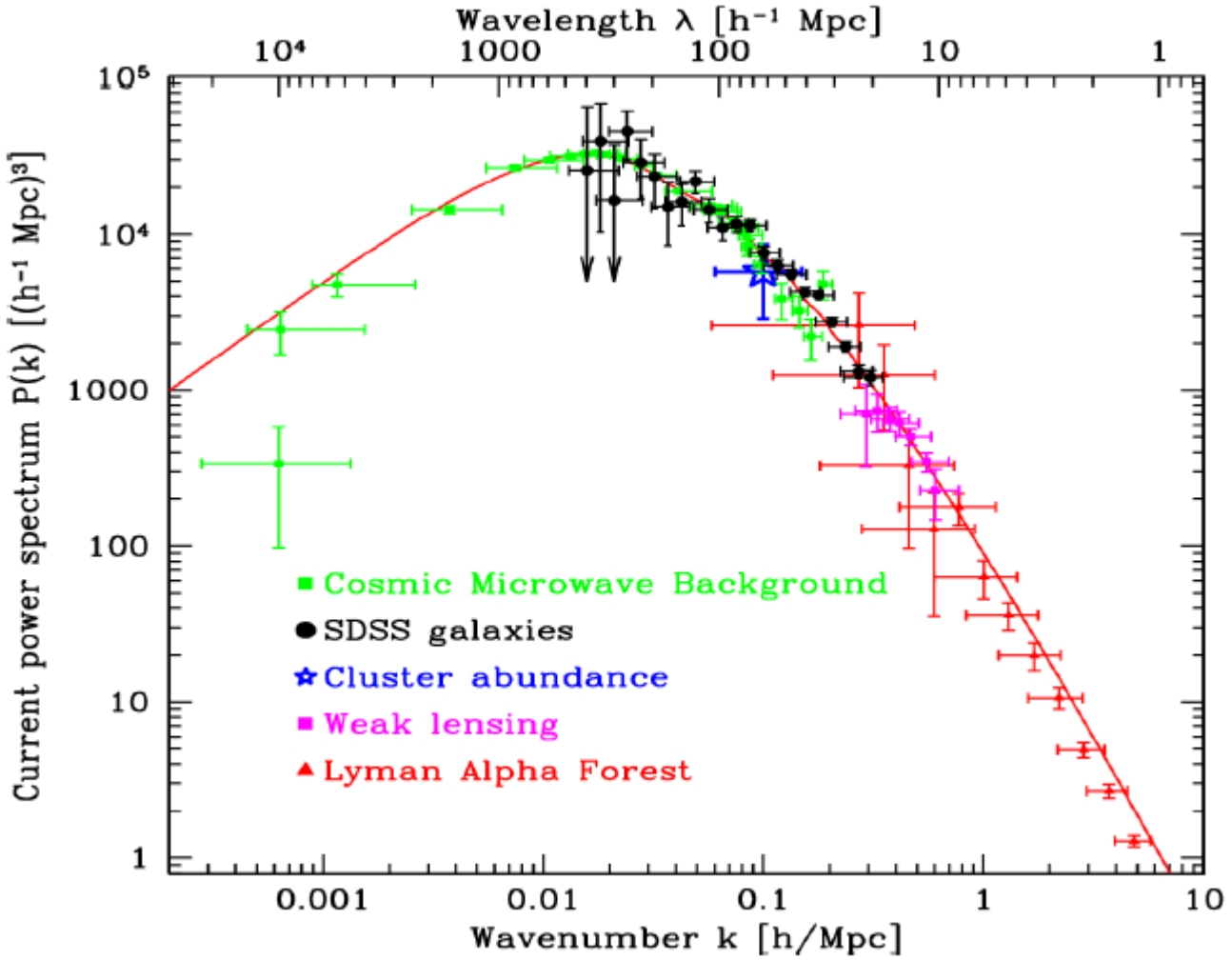
$$a = 6.4h^{-1}Mpc$$

$$b = 3.0h^{-1}Mpc$$

$$c = 1.7h^{-1}Mpc$$

$$\nu = 1.13$$

credit M. Tegmark



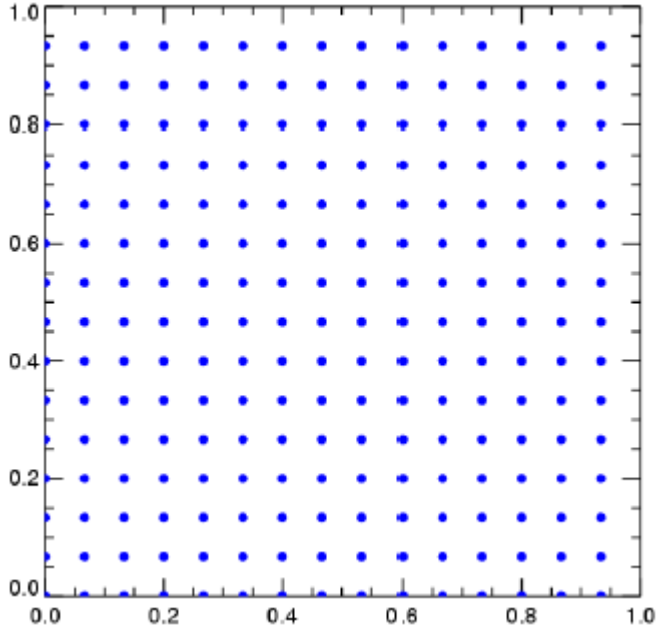
The normalization constant A is determined by observations of present clustering:

$$\sigma^2(M,t) = D_+^2(t)\frac{V}{(2\pi)^3}\int_0^\infty 4\pi k^2 P(k) W(kR) dk$$

$$R = 8h^{-1}Mpc : \sigma_8 \simeq 0.8 - 0.9 \rightarrow A$$

These conditions fully specify the initial conditions

We then assume a grid of particles as a representation of our unperturbed density field



Individual modes in Fourier space are assigned assuming random phases and amplitudes

$$\delta_k = B_k e^{i\varphi_k} \qquad < \delta_k^{\,2} >= \quad P(k)$$

The random realization can represent modes up to the Nyquist frequency

$$k_{Ny} = \frac{2\pi}{L}\frac{N_p}{2}$$

From the density fluctuations we must now obtain the particle displacements and velocities. This is obtained in the Lagrangian representation (Zel'dovich approximation)

$$\vec{x} = \vec{x}_0 + D_+(t)\vec{\psi} \qquad \vec{\Psi} = -\vec{\nabla}\phi$$

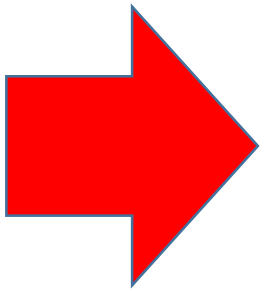
$$\vec{v} = a(t)\dot{D}_+(t)\vec{\psi} \qquad \delta = -\vec{\nabla}\cdot\vec{\psi}$$

The choice of N

- In cosmology the formation of structures proceeds hierarchically : small structures form first and the larger ones follow because of merging and accretion
- This poses a problem in cosmological simulations :

The mass $m = \frac{M}{N} = \frac{\bar{\rho} L^3}{N}$ of the particles must be small to resolve the internal halo structure

The box size L must be large in order to contain a representative sample of the Universe



need large N

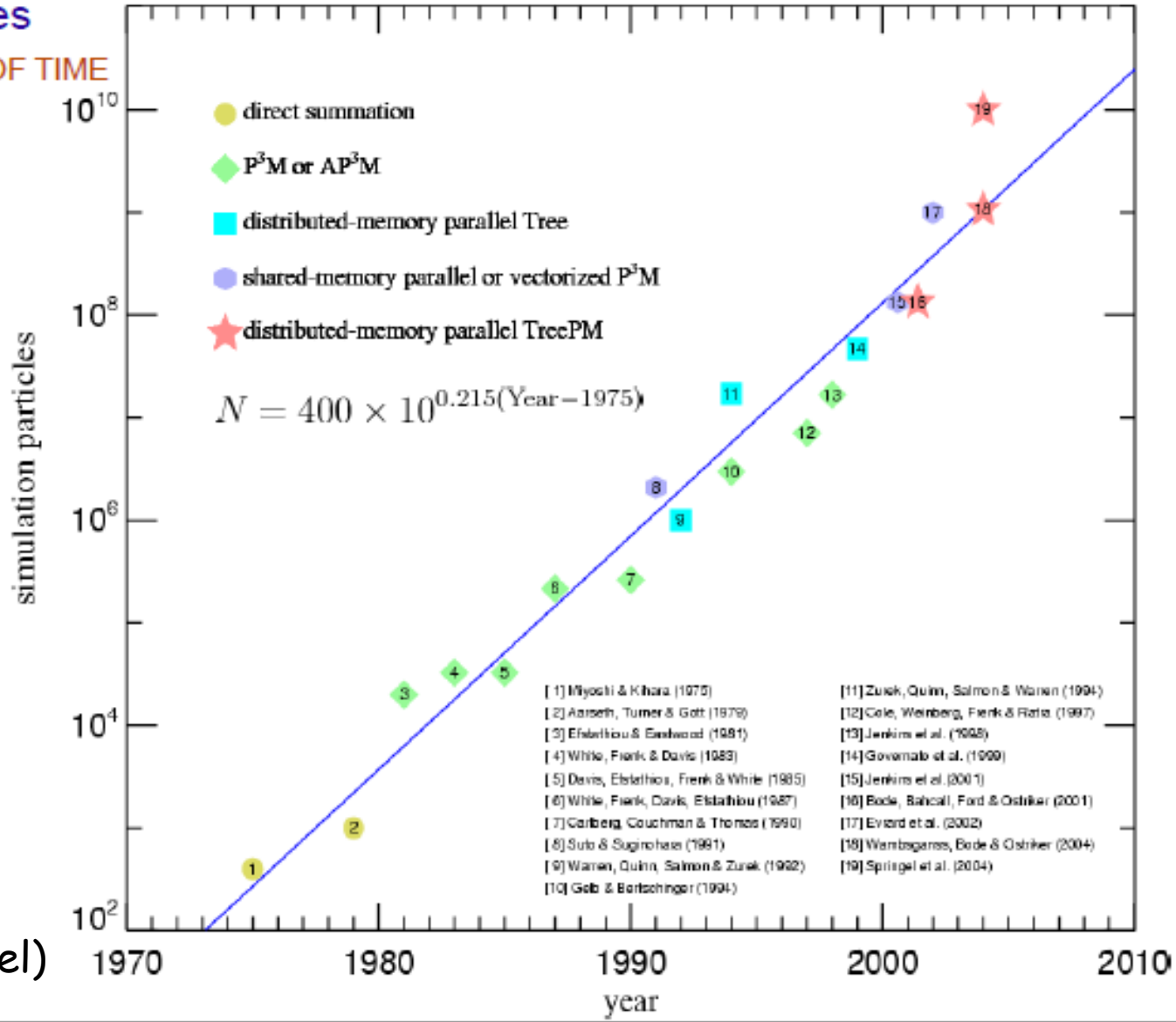
In numerical cosmology, the number of particles N represents the 'quality' of the simulations

A lot of efforts have been put over the years in the building of efficient parallel codes

Cosmological N-body simulations have grown rapidly in size over the last three decades

"**N**" AS A FUNCTION OF TIME

- Computers double their speed every 18 months (Moore's law)
- N-body simulations have doubled their size every 16-17 months
- Recently, growth has accelerated further.
  The Millennium Run should have become possible in 2010 – we have done it in 2004 !



(credit V.Springel)

The gravity calculation

We assume that dark matter interacts only gravitationally —> collisionless dynamics

The system is described by the single particle distribution function $f(\vec{x},\vec{v},t)$

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial \vec{x}} \cdot \vec{v} + \frac{\partial f}{\partial \vec{v}} \cdot \left(-\frac{\partial \Phi}{\partial \vec{x}} \right) = 0 \qquad \nabla^2 \Phi(\vec{x},t) = 4\pi G \int f(\vec{x},\vec{v},t) d^3v$$

- This system of partial differential equation cannot be solved directly
- The wise approach is to use a finite set of N particles to represent $f(\vec{x},\vec{v},t)$

We introduce N particles which move along the characteristics of the system:

$$\frac{d^2\vec{x}_i}{dt^2} = -\vec{\nabla}_i \Phi$$

$$\Phi(\vec{x}) = -G \sum_{j=1}^{N} \frac{m_j}{(\vec{x}-\vec{x}_j)^2 + \varepsilon^2}$$

Cost/step $\propto N^2$

ε is the gravitational softening and is introduced to avoid large-angle scattering and keep the two-body relaxation time large

A brute solution of the N-boby problem implies a computational cost $\propto N^2$, for large N

$(\geq 10,000)$ this approach is clearly impratical

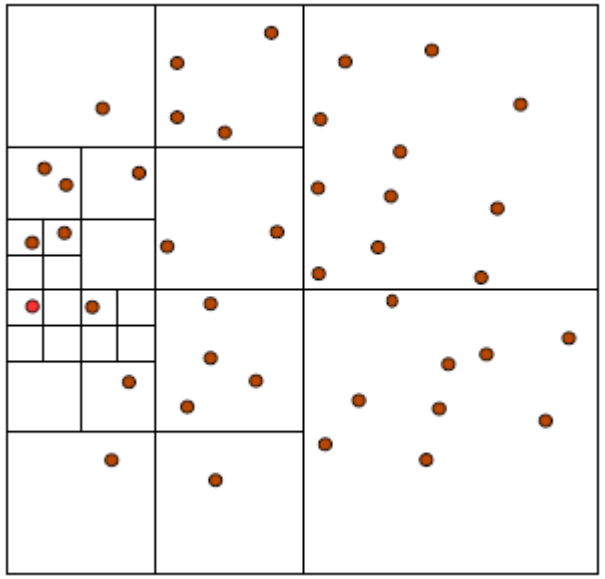
It is then necessary the use of approximate methods . The most popular are :

1. Grid methods
2. Hierarchical tree methods

We shall describe the latter

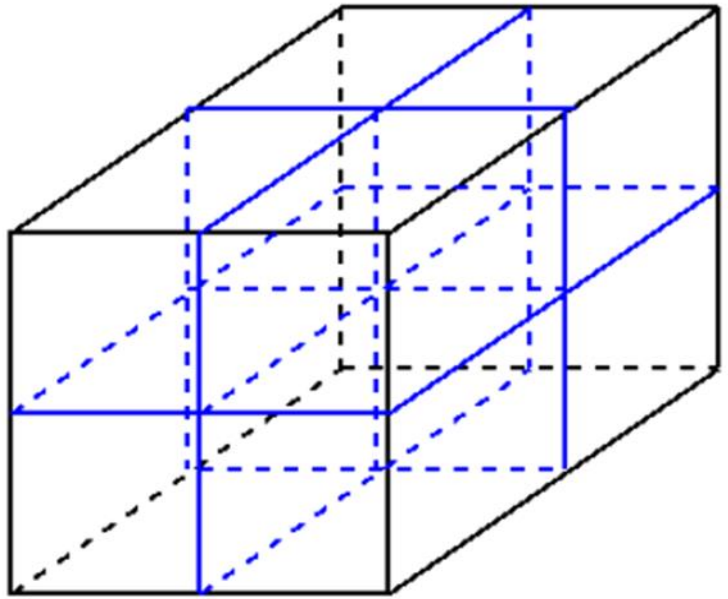
The tree algorithm

The idea is to start from a root cell of size L and perform a recursive partition of space by dividing the domain into subcells of size $l_k = L / 2^k$ (k=0, 1, 2, …) until there are no particles left to examine



In two (three) dimensions the structure is called a quad(oct) -tree

An example of a Barnes-Hut oct-tree:



The whole procedure is equivalent to a hierarchical partition of space

The potential of the system is : $\Phi(\vec{x}) = -G \sum_i \frac{m_i}{|\vec{r} - \vec{x}_i|}$

Once that you have constructed the tree you compute in the CM frame the moments (monopole, quadrupole etc..) of the cells

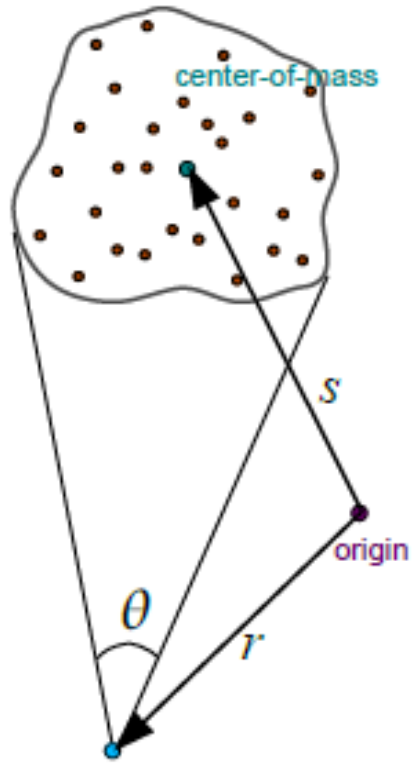
$$\frac{1}{|\vec{r} - \vec{x}_i|} = \frac{1}{|(\vec{r} - \vec{s}) - (\vec{x}_i - \vec{s})|}$$

$$|\vec{x}_i - \vec{s}| << |\vec{r} - \vec{s}|$$

$$\vec{y} \equiv \vec{r} - \vec{s}, \vec{s} = c.o.m.$$

$$M = \sum_i m_i$$

$$Q_{ij} = \sum_k m_k \left[3(\vec{x}_k - \vec{s})_i (\vec{x}_k - \vec{s})_j - \delta_{ij} (\vec{x}_k - \vec{s})^2\right]$$

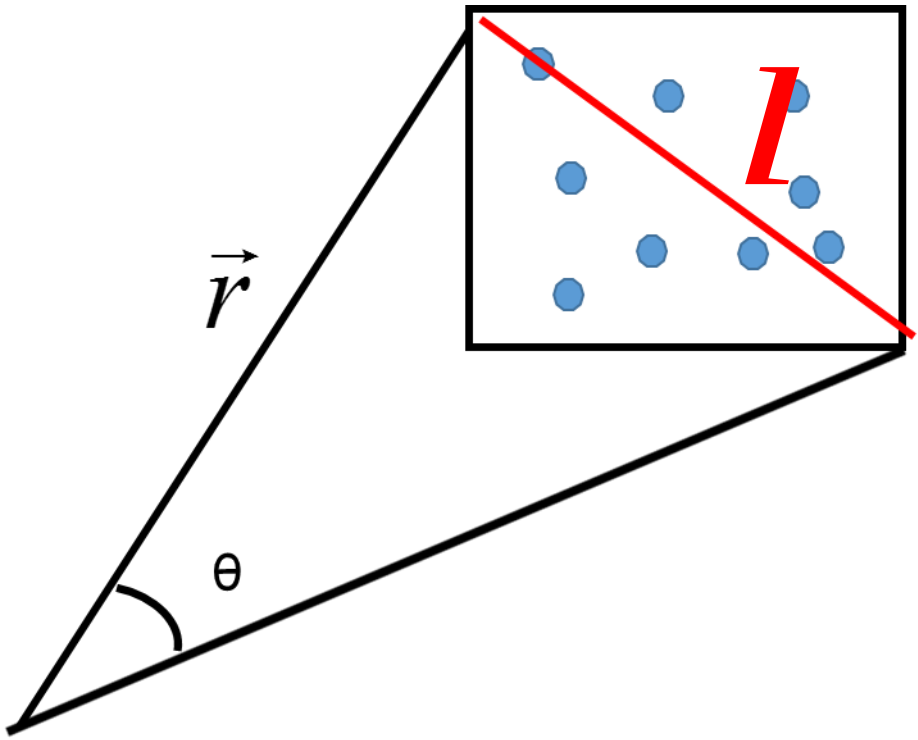


Here is the breakthrough of the algorithm: a tolerance parameter θ is introduced to compute the potential. For particle i at distance r from a cell, the contribution of the cell to the potential is

$$\Phi(\vec{r}) = -G\left[\frac{M}{|\vec{y}|} + \frac{1}{2}\frac{\vec{y}^T \cdot \vec{Q} \cdot \vec{y}}{|\vec{y}^5|}\right]$$

If the condition $r > l/\theta$ is satisfied (θ~0.3-0.6)

For a single force evaluation the number of terms is now $\propto \log(N)$



Main advantages of a tree:

- No restrictions on the dynamic range
- Force accuracy adjusted to the specified level
- Computational cost weakly dependent on clustering
- Allows for arbitrary geometries

CODE PARALLELIZATION

The main challenge to parallelize a code is the domain decomposition and the load balancing

One has to distinguish between

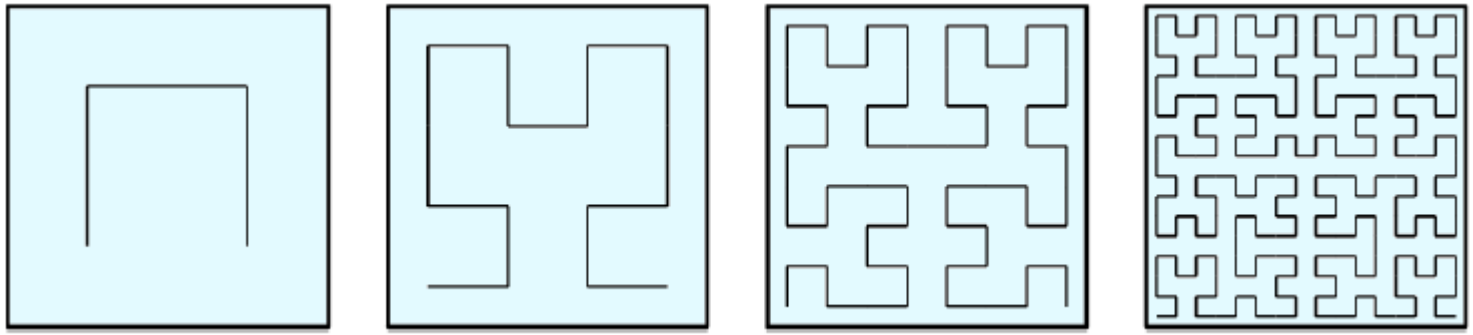
- Homogeneous particle distribution (initial state, early epochs…)
  - Cartesian decomposition
  - Easy FFT decomposition

- Strongly clustered distribution (late times)
  - Use a space filling curve (SFC : Hilbert)
  - FFT pattern complicated

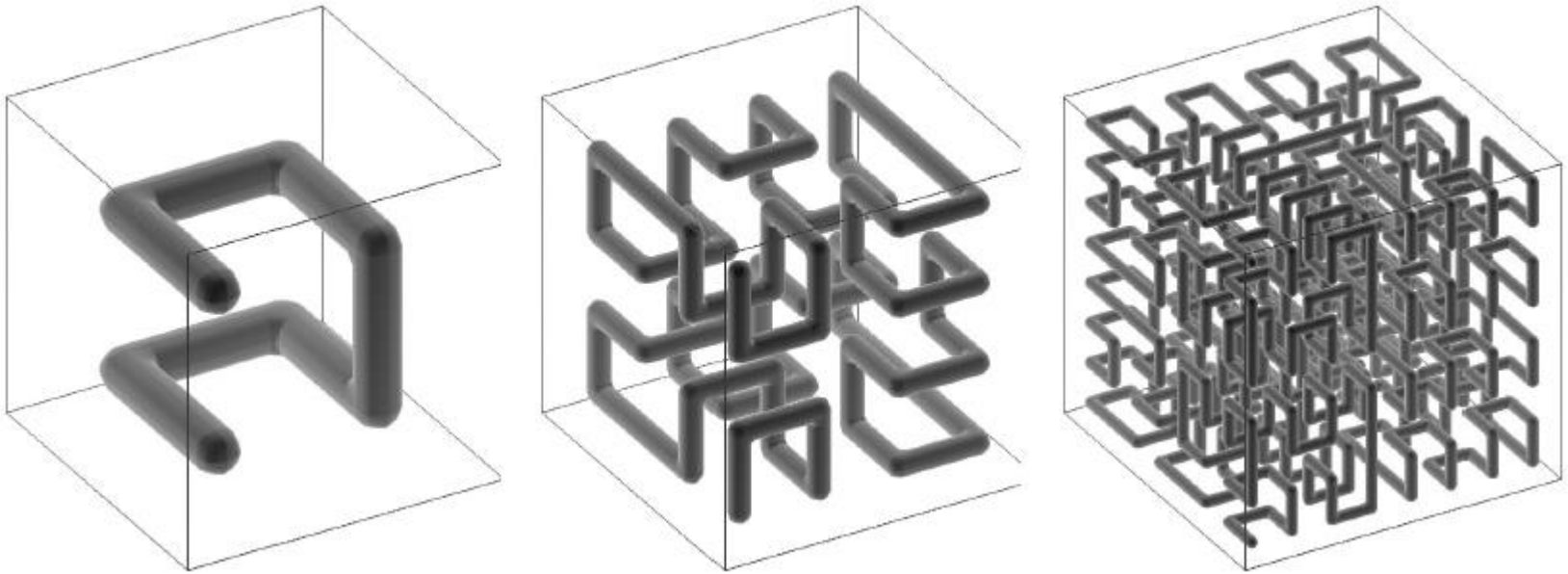
The Hilbert curve decomposition:

- All particles are ordered according to a 1D index
- The curve is a fractal that fills the space with several properties
- The most important is that proximitity of points in 3D is kept in 1D

HILBERT CURVES OF ORDERS 1,2,3,4 IN 2D

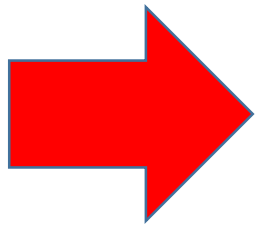


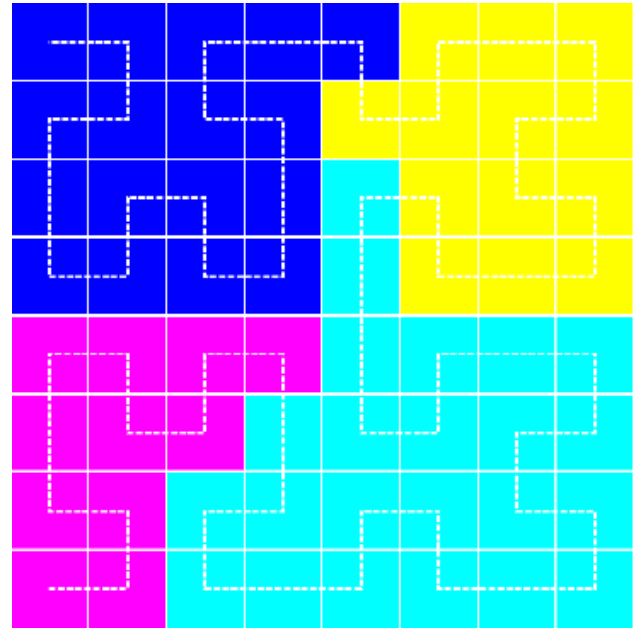
HILBERT CURVES IN 3D



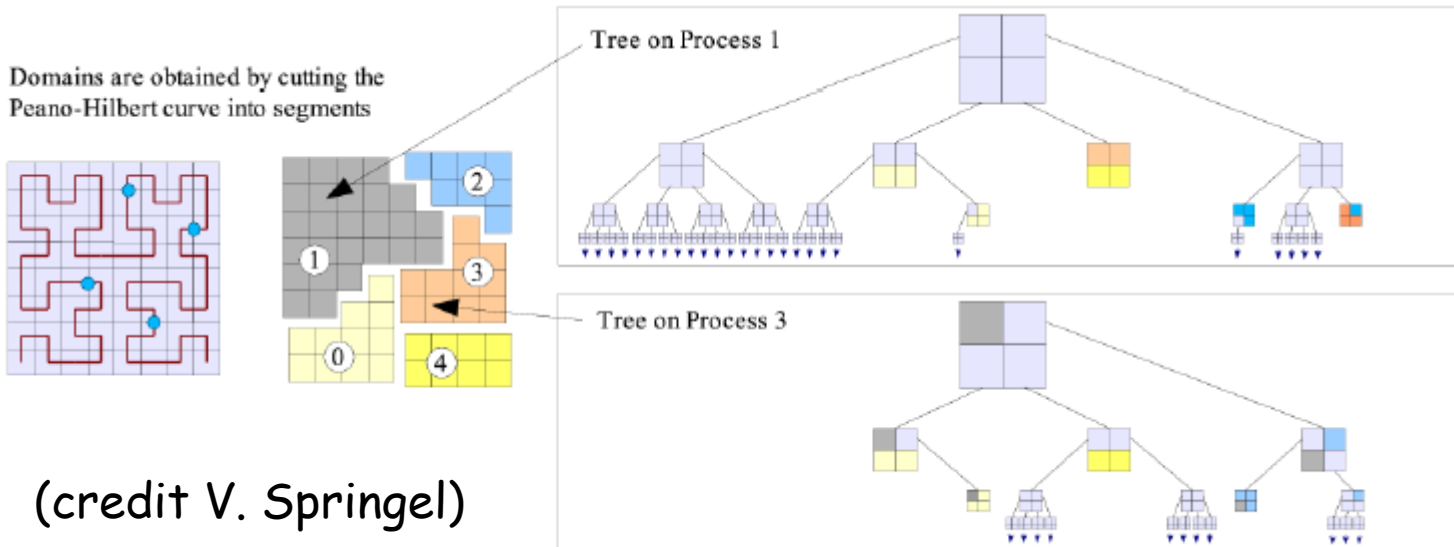
To parallelize the code the global Hilbert curve is first calculated

Each thread is then defined by a segment of the H curve so to have the same number of particles or of work load



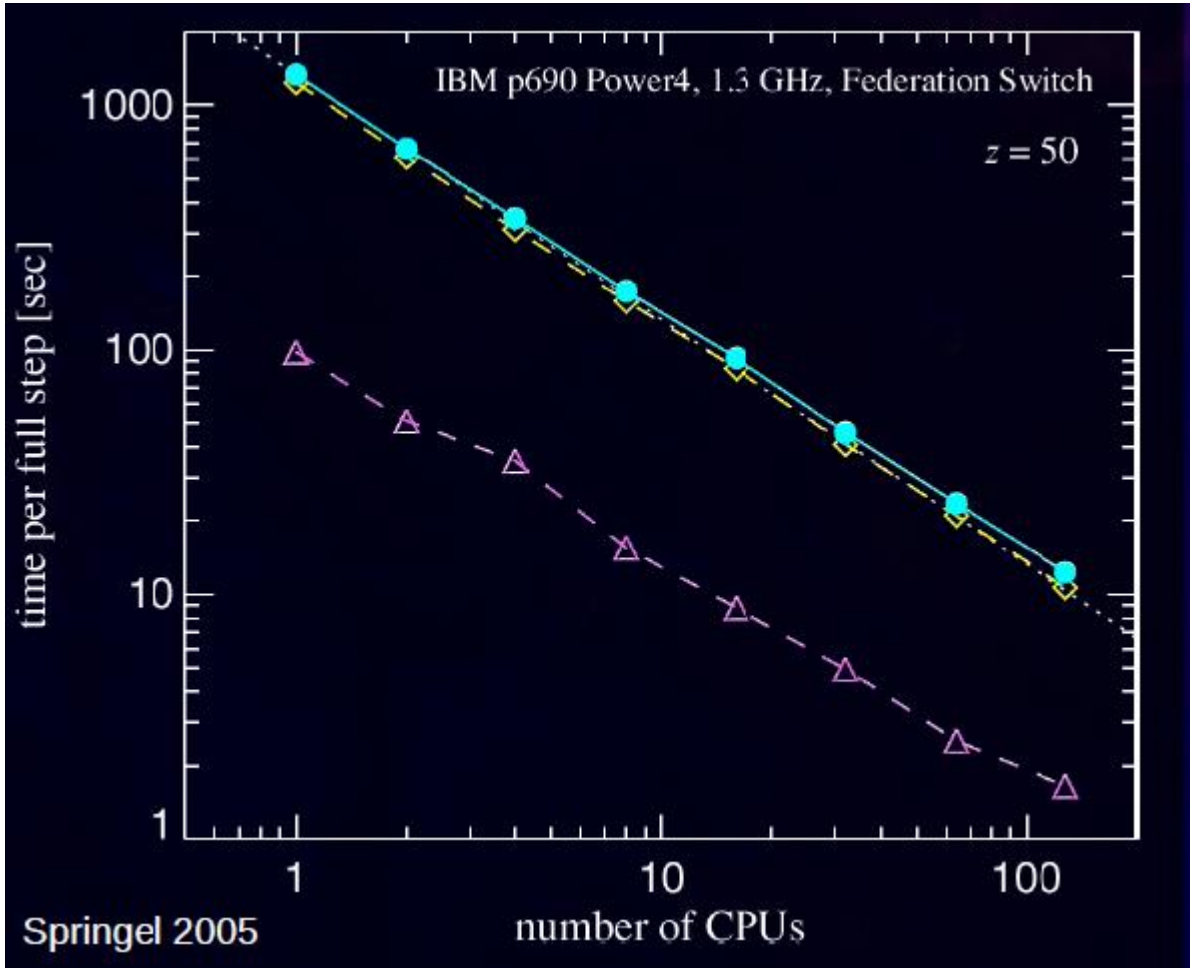


For a tree code the global tree is splitted among the processors



(credit V. Springel)

This is a plot of the scalability of the most popular (Gadget) Lagrangian code used in numerical cosmology:



Hydrodynamic

In addition to gravity, in numerical cosmology a self-consistent modelling requires to incorporate the equations of hydrodynamic to follow the evolution of baryons in the Universe

In cosmology, hydrodynamic codes that are used to perform simulation of structure formation fall into two categories:

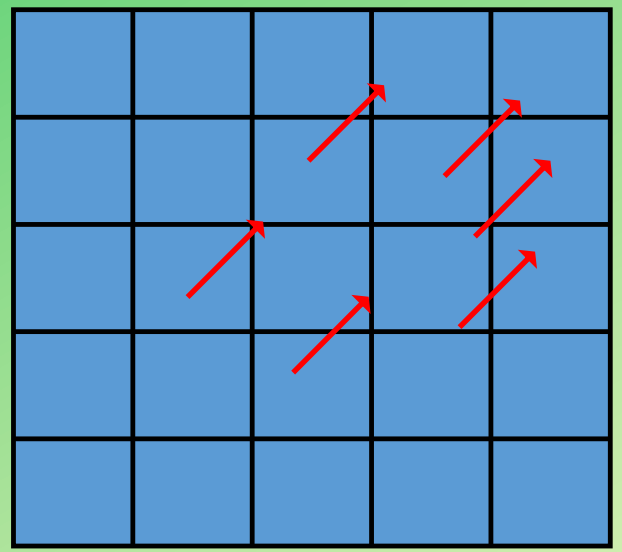
- Eulerian: fluid equations solved onto a mesh
- Lagrangian : fluid equations described trough a set of evolving particles of costant mass

Here, we will shortly describe the second method

NUMERICAL HYDRODYNAMICS: DIFFERENT METHODS TO DISCRETIZE A FLUID

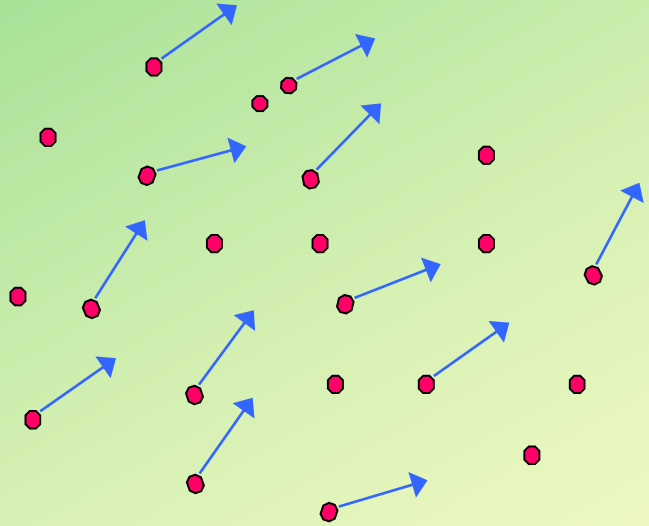
Eulerian

discretize space:
representation on a mesh



Lagrangian

discretize mass:
representation by fluid elements (particles)



MAIN ADVANTAGES:

high accuracy (shock capturing),
low numerical viscosity

resolution adjust
automatically to the flow

SPH IN A NUTSHELL

- The fluid is represented by an ensemble of particles carrying mass, momentum and hydro properties: $m_i, \quad \rho_i, \quad v_i, \quad \ldots$

- Hydro observables are defined by local averages

$$< f(\vec{r}) >= \int f(\vec{r}')W(\vec{r}-\vec{r}',h)d\vec{r}'$$

- W is the smoothing kernel and must satisfy

$$\int W d\vec{r} = 1 \quad \lim_{h\to 0} < f(\vec{r},h) > \to f(\vec{r})$$

- The basic concept is that of a particle representation of the fluid: integrals are replaced by summation over discrete set of points :

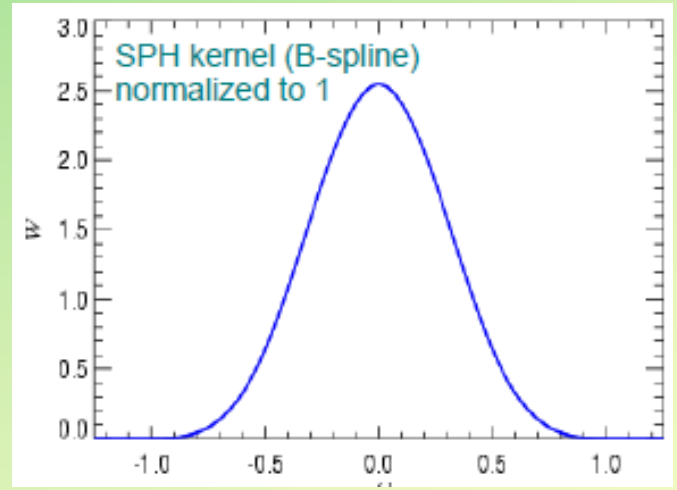
$$\rho_i = \sum_j m_j W(r_{ij},h_i), \; h_i = \eta(m_i/\rho_i)^{(1/D)} \to \frac{4\pi(2h_i)^3\rho_i}{3} = N_{sph}m_i$$

- The mass of a particle is smeared over the kernel domain: smoothed particle hydrodynamics

Kernel interpolation is used in SPH to construct fluid quantities from discrete tracers

For a given function A(r): $$< A(\vec{r}) >= \int W(\vec{r} - \vec{r}', h) A(\vec{r}') d^3 r'$$

The integral is replaced by a sum according to the rule : $$d^3 r' \rightarrow \frac{m_j}{\rho_j}$$

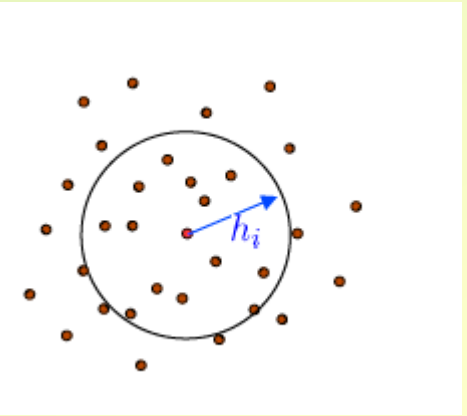


$$< A_i >= \sum_{j=1}^{N} \frac{m_j}{\rho_j} A_j W(\vec{r}_{ij}, h_i)$$

setting $A_i = \rho_i$ one recovers the density estimate

In SPH kernels always have compact support

$$\rho_i = \sum_{j=1}^{N} m_j W(| r_{ij} |, h_i)$$



FUNDAMENTAL HYDRO EQUATIONS

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \vec{v}\cdot\vec{\nabla}\rho = -\rho\vec{\nabla}\cdot\vec{v}$$

continuity

$$\frac{d\vec{v}}{dt} = \frac{\partial \vec{v}}{\partial t} + \left(\vec{v}\cdot\vec{\nabla}\right)\vec{v} = -\frac{1}{\rho}\vec{\nabla}P + \eta\vec{\nabla}^2\vec{v} + \left(\zeta + \frac{\eta}{3}\right)\vec{\nabla}(\vec{\nabla}\cdot\vec{v})$$

momentum

$$\frac{d\varepsilon}{dt} = \frac{\partial \varepsilon}{\partial t} + (\vec{v}\cdot\vec{\nabla})\varepsilon = T\frac{ds}{dt} - \frac{p}{\rho}\vec{\nabla}\cdot\vec{v}$$

energy

viscous terms

The hydro equations are complemented by an EOS: $P = A(s)\rho^{\gamma}$ and self-gravity

In the entropy-conserving scheme :

artificial viscosity tensor

$$\frac{d\vec{v}_i}{dt} = -\sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij}\right)\vec{\nabla}_i W_{ij}(h_i)$$

$$\frac{dA_i}{dt} = \frac{1}{2}\frac{\gamma-1}{\rho_i^{\gamma-1}}\sum_j \underbrace{m_j \Pi_{ij}\vec{v}_{ij}\cdot\nabla_i \bar{W}_{ij}}_{\text{entropy due to numerical viscosity}} - \frac{\gamma-1}{\rho_i^{\gamma}}\Lambda(\rho_i, T_i),$$

cooling

NEIGHBOUR SEARCH

Because the kernel W has compact support, the summation $\rho_i = \sum_{j=1}^{N} m_j W(|r_{ij}|, h_i)$

reduces to the problem of finding all of the points for which $r_{ij} \leq h_i$

In SPH the search for close particle pairs is a crucial problem:

- This is part of a more general problem: the k-nearest neighbour problem (kNN)
- The solution is to find a subset Ω of the input data set S such that any member of Ω is closer to the query point q that any member of Ω\S, the set difference between Ω and S

There are different approaches:

- Hashing (subdivision into fixed cells)
- Tree (recursive spatial subdivision)
- Sorting of SFC