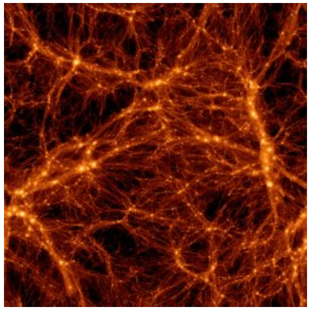
Numerical Cosmology

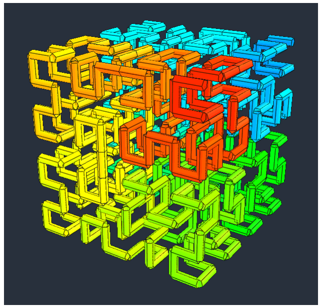
Lecture 1

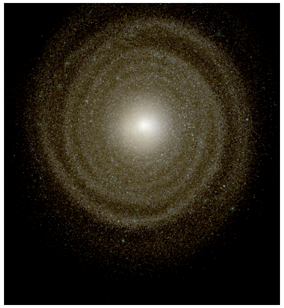
Collisionless N-body Dynamics

Romain Teyssier

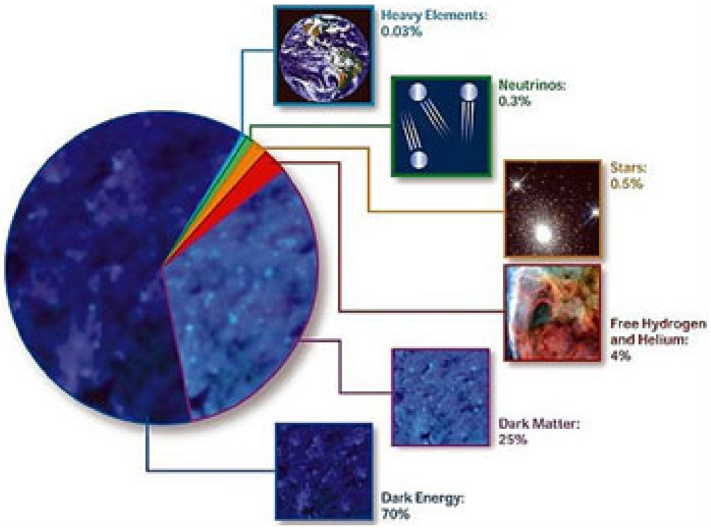
PRINCETON UNIVERSITY

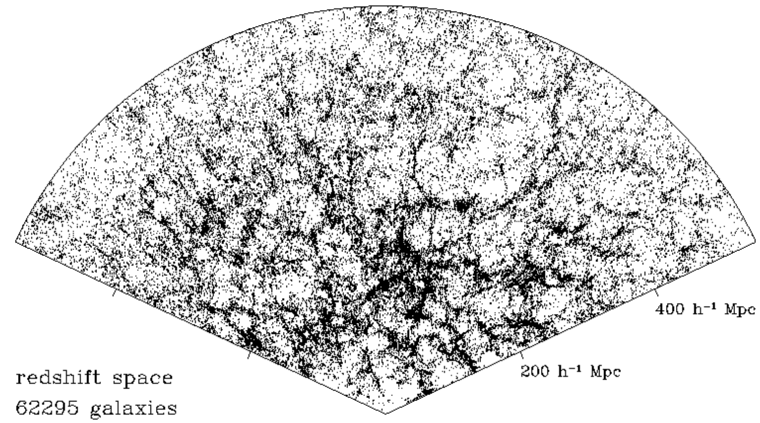


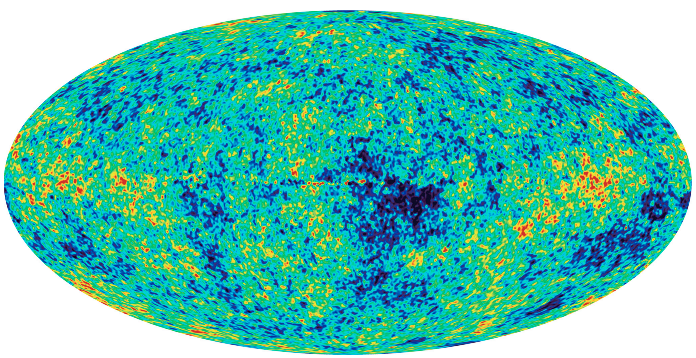


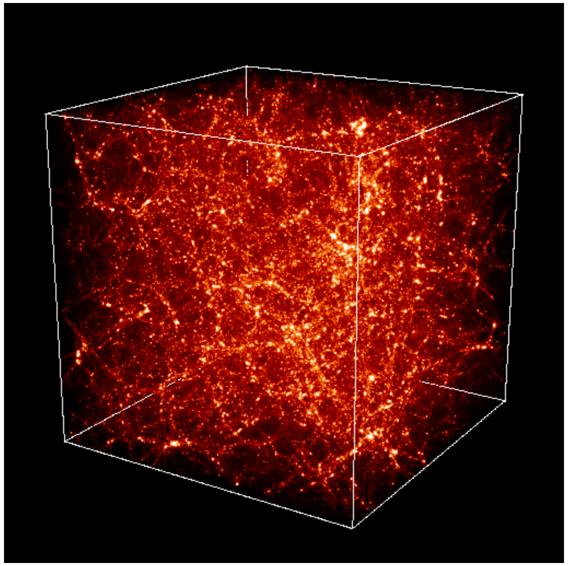


Cosmological simulations



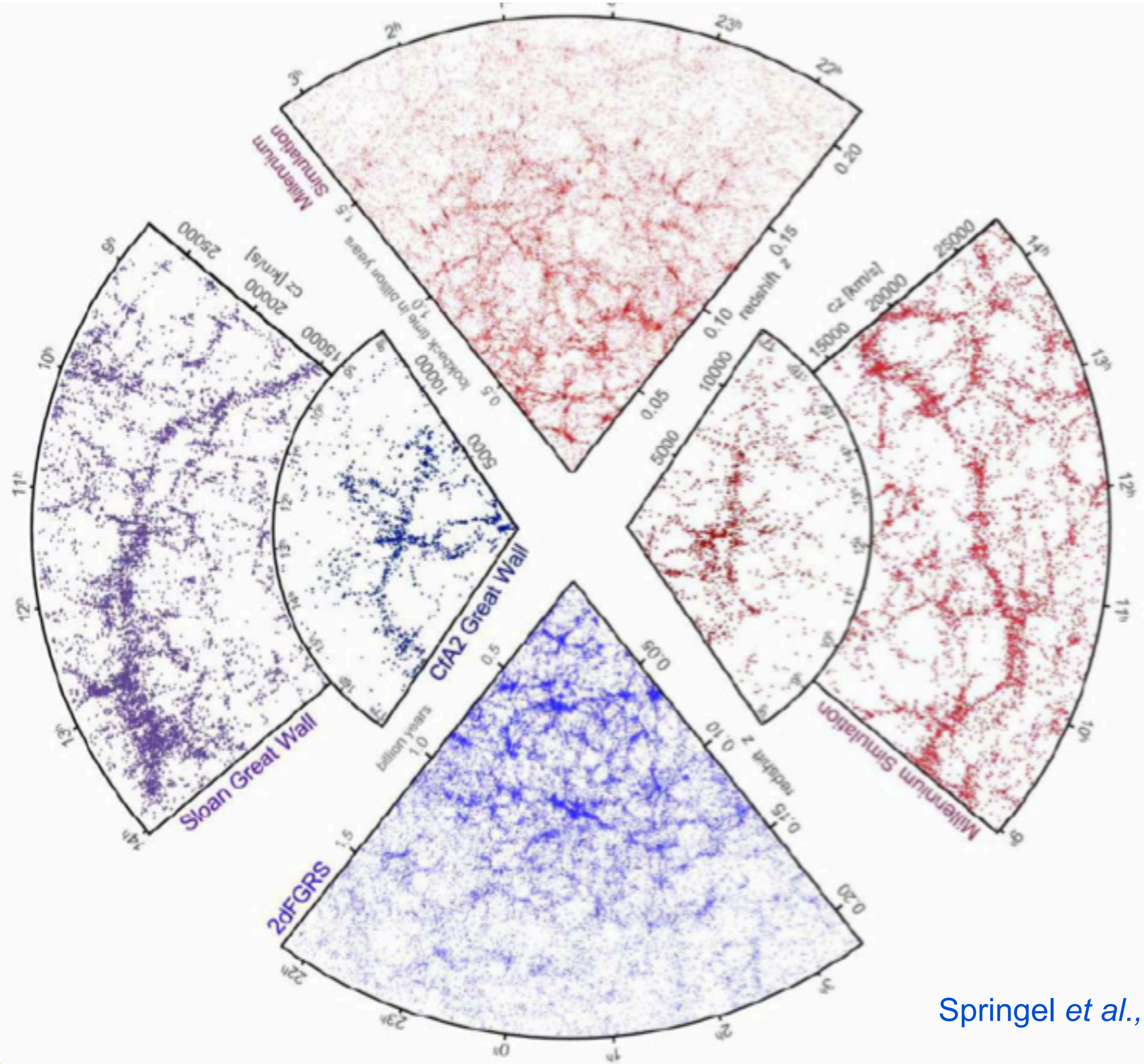






From Gaussian random fields to galaxies: non-linear dynamics of gravitational instability with N-body and hydrodynamics codes.

Cosmological simulations



Springel *et al.,* Nature, 2006

The Vlasov-Poisson equation

Collisionless limit of the Boltzmann equation:

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial}{\partial t} f(\mathbf{x}, \mathbf{p}, t) + \frac{\mathbf{p}}{ma^2} \frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}, \mathbf{p}, t) - m\nabla_{\mathbf{x}}.\Phi(\mathbf{x}) \frac{\partial}{\partial \mathbf{p}} f(\mathbf{x}, \mathbf{p}, t) = 0$$

Liouville theorem: number of particles in conserved in phase-space

Gravitational acceleration is given by the Poisson equation

$$\Delta\Phi(\mathbf{x}) = \frac{4\pi Gm}{a} \left(\int f(\mathbf{x}, \mathbf{p}, t) \mathrm{d}^3\mathbf{p} - \overline{n} \right),$$

3 solution strategies:

- pure fluid on a 6D grid (Yoshikawa et al. 2013) or on a cold 3D manifold (Abel et al. 2012)

- pure N body using direct force computations or fast multipole methods (Barnes & Hut 1986; Bouchet & Hernquist 1988)

- mixture of the 2: the Particle-Mesh method (Hockney & Eastwood 1988)

The Particle-In-Cell method

N body integrator coupled to a grid-based Poisson solver

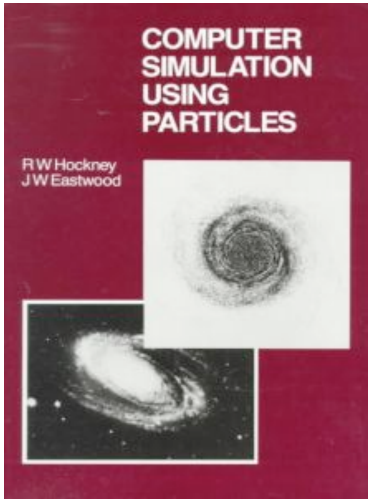
$$\frac{d\mathbf{x}_p}{dt} = \mathbf{v}_p \quad \text{and} \quad \frac{d\mathbf{v}_p}{dt} = -\nabla_x \phi$$

1- Compute the mass density field on the grid from the particle distribution

2- Solve for the Poisson equation on the grid

3- Interpolate the force back to the particle position

Hockney, R.W., Eastwood, J.W., “Computer Simulation Using Particles”, CRC Press (1985)



The PIC or PM (Particle-Mesh) scheme has been applied to:

- Hydrodynamics (compressible, incompressible, MHD)
- Plasma physics
- Self-gravitating systems

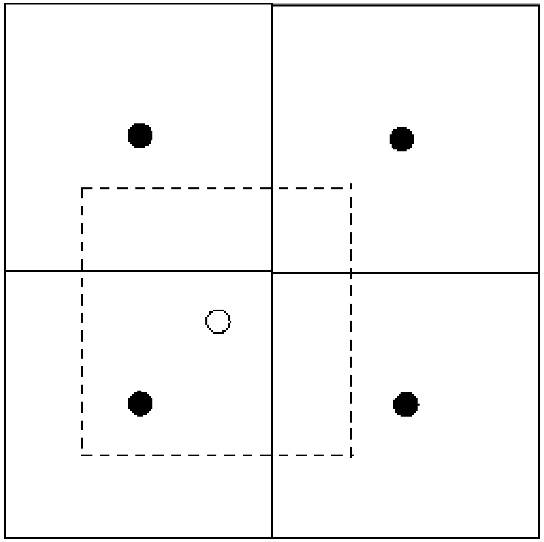
Mass assignment schemes

Assign to each particle a "shape"

Nearest Grid Point (NGP): $$S(x) = \frac{1}{\Delta x}\delta\left(\frac{x}{\Delta x}\right)$$

Cloud-In-Cell (CIC): $$S(x) = \frac{1}{\Delta x}\Pi\left(\frac{x}{\Delta x}\right)$$

Triangular Shape Cloud (TSC): $$S(x) = \frac{1}{\Delta x}\Delta\left(\frac{x}{\Delta x}\right)$$



"Cloud-In-Cell" interpolation

The contribution of each particle to the mass in the cell is:

$$W^{\rho}(x_p - x_i) = \int_{x_i-\Delta x/2}^{x_i+\Delta x/2} S(x_p - x)dx$$

The total mass in the cell is:

$$\rho_i = \frac{1}{\Delta x}\sum_{p=1}^{N_p} m_p W^{\rho}(x_p - x_i)$$

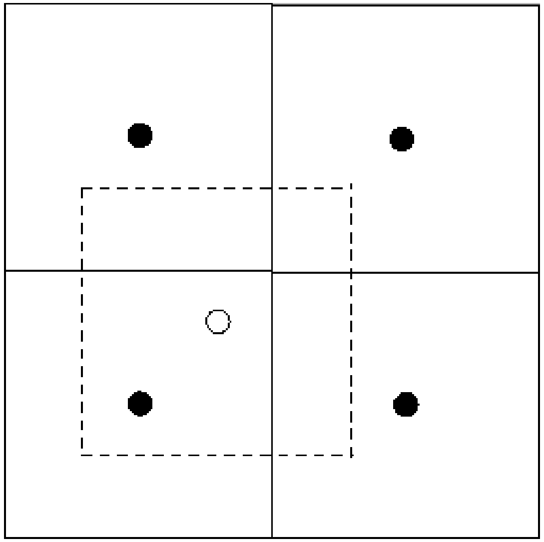
Force interpolation schemes

Use another interpolation scheme to get the mesh force at particle positions.

$$F(x_p) = m_p \sum W^F(x_p - x_i) F_i$$

Momentum conservation is enforced if:

- 2 interacting particles see equal but opposite forces
- no self-forces



"Cloud-In-Cell" interpolation

Poisson equation $\Delta_{ij}\Phi_j = \rho_i$

Gradient of the potential $F_i = -\nabla_{ij}\Phi_j$

Self-force for particle p: $\partial F(x_p) =$

$$-m_p^2 \sum_i \sum_j W^F(x_p - x_i) \left(\nabla.\Delta^{-1}\right)_{ij} W^\rho(x_p - x_j)$$

Self-force is zero if operator is antisymmetric and force and mass assignment schemes are equal.

Force solver using Fourier analysis

Use of Fast Fourier Transform to solve for the Poisson equation

Poor's man Poisson solver:

$$\frac{\partial^2\Phi}{\partial x^2} = \rho \qquad -k^2\tilde{\Phi}(k) = \tilde{\rho}(k) \qquad \tilde{G}(k) = -\frac{1}{k^2}$$

$$\frac{\partial\Phi}{\partial x} = -F \qquad -ik\tilde{\Phi}(k) = \tilde{F}(k) \qquad \tilde{D}(k) = -ik$$

Using finite difference approximations:

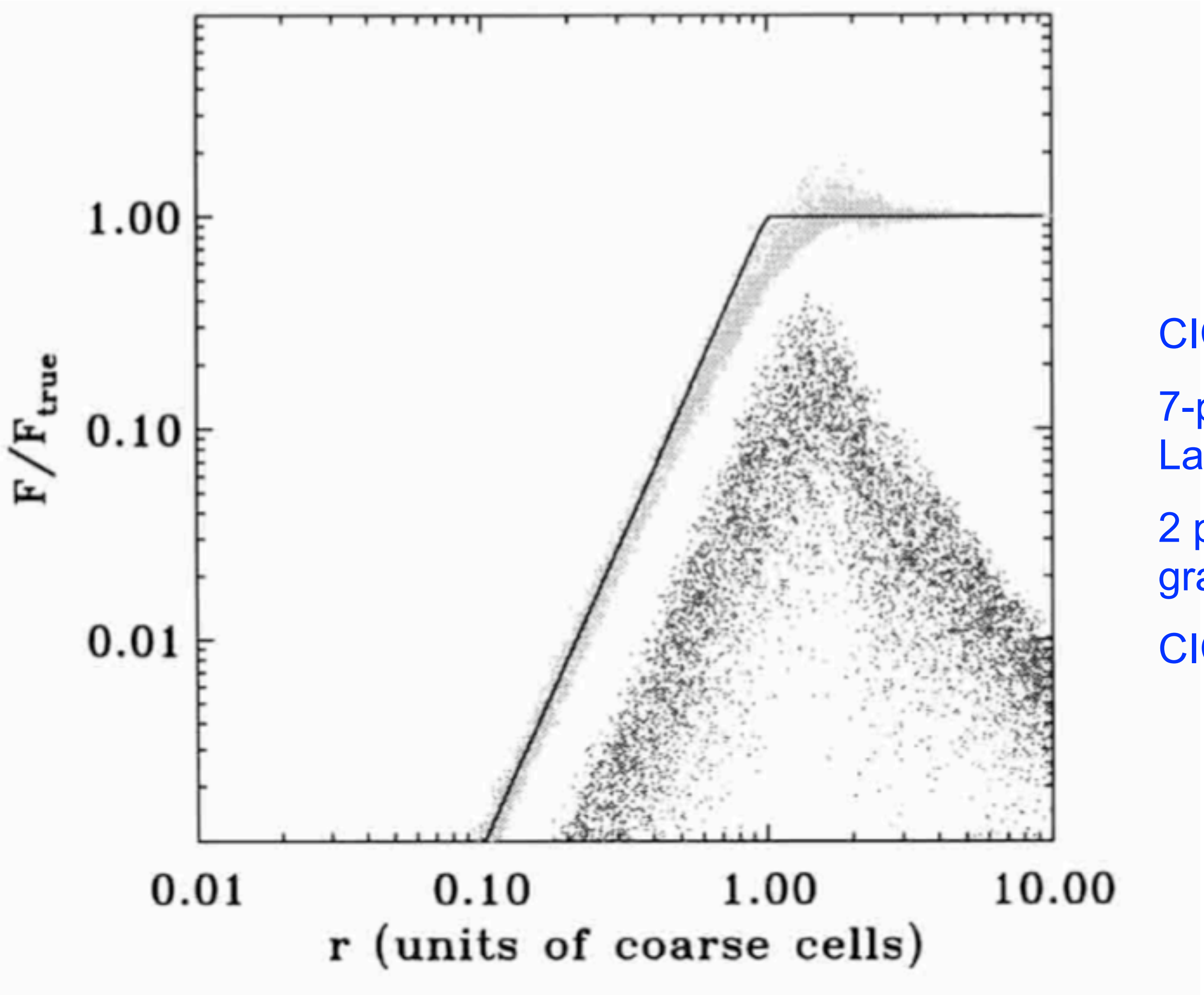
$$\Phi_{i+1} - 2\Phi_i + \Phi_{i-1} = \rho_i\Delta_x^2 \qquad \tilde{G}(k) = -\frac{\Delta x^2/4}{\sin(\frac{k\Delta x}{2})^2}$$

$$-(\Phi_{i+1} - \Phi_{i-1}) = F_i\Delta_x \qquad \tilde{D}(k) = -i\frac{\sin(k\Delta x)}{\Delta x}$$

Final force is given by:

$$\tilde{F}(k) = -\frac{m_p^2}{\Delta x^2}\tilde{W}^F(k)\tilde{D}(k)\tilde{G}(k)\tilde{W}^\rho(k)\tilde{n}(k)$$

Overall PM force accuracy



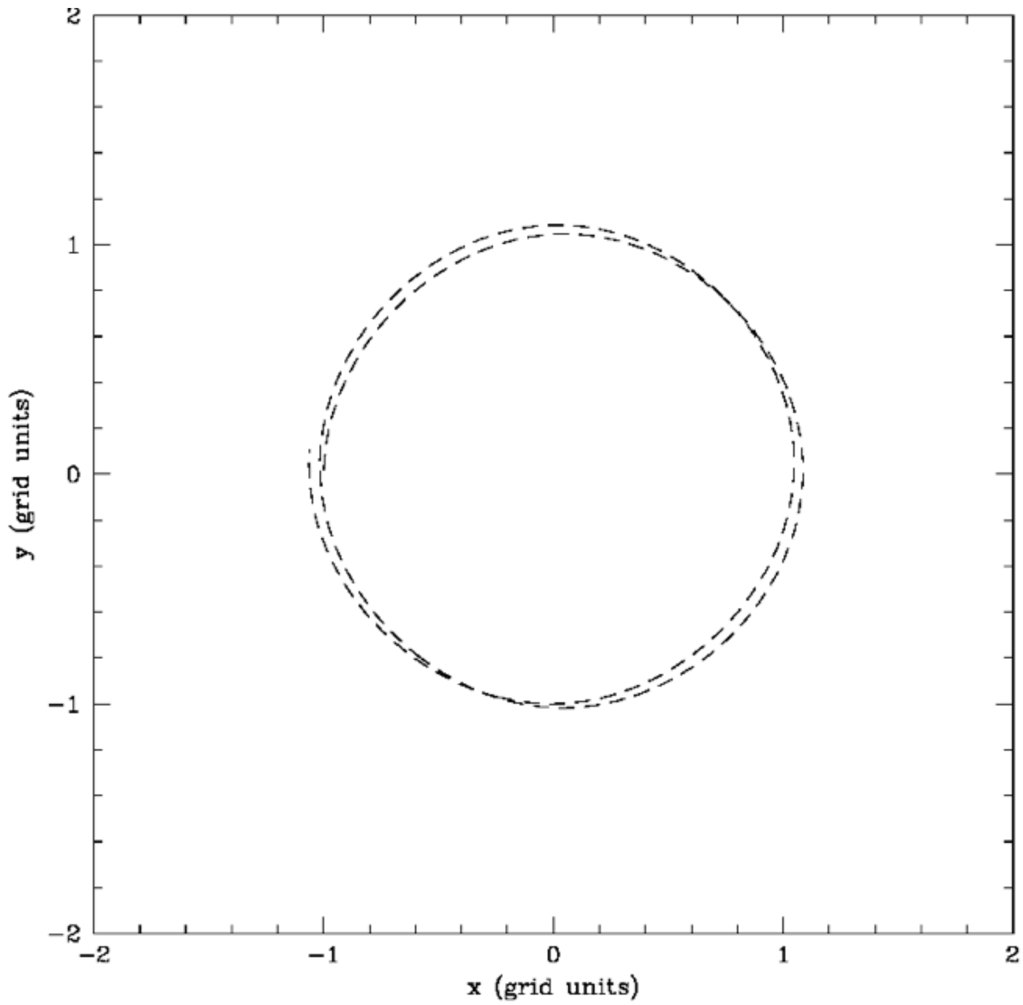
CIC

7-point Laplacian

2 points gradient

CIC^{-1}

Overall PM force accuracy

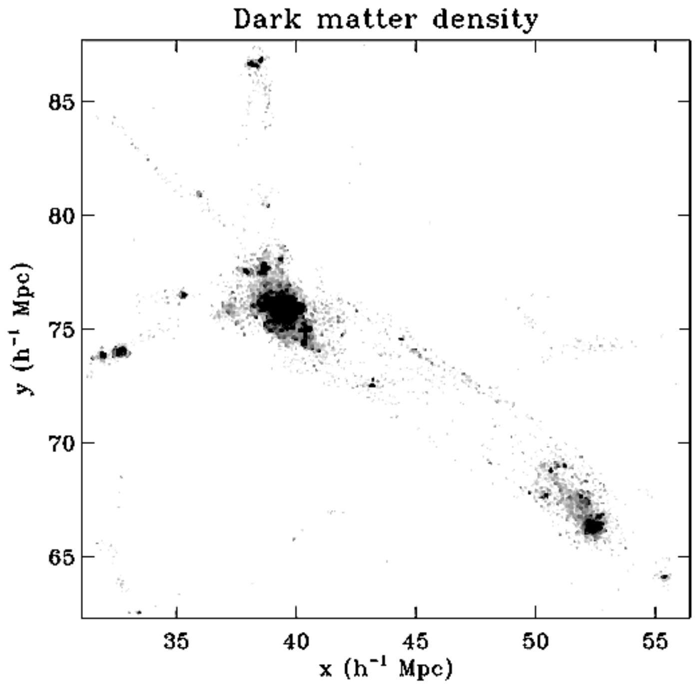


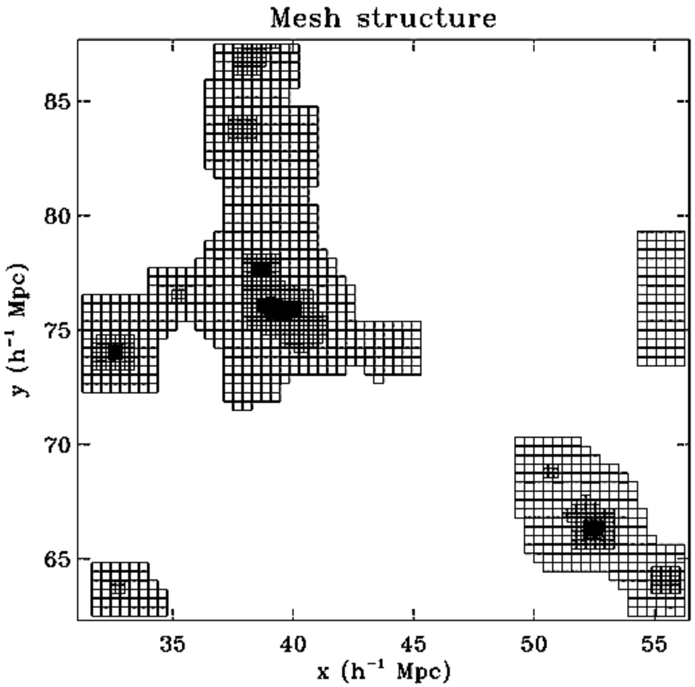
example of particle trajectory

PM with Adaptive Mesh Refinement

At each grid level, the force softening is equal to the local grid size.

For pure dark matter simulations, using a quasi-Lagrangian strategy, the particle shot noise is kept roughly constant.





Relaxation solvers for the Poisson equation

Solve the linear system $\Delta_{ij}\Phi_j = \rho_i$ with arbitrary mesh geometry.

Simplest scheme: the Jacobi method (in 2D).

$$\phi_{i,j}^{n+1} = \frac{1}{4}\left(\phi_{i+1,j}^n + \phi_{i-1,j}^n + \phi_{i,j+1}^n + \phi_{i,j-1}^n\right) - \frac{1}{4}\rho_{i,j}$$

Converge very slowly for long wavelength and large grids.

Very sensitive to the initial guess.

Faster convergence is obtained for Gauss-Seidel “over-relaxation“ method with red-black ordering.

$$\phi_{i,j}^{n+1} = \omega\phi_{i,j}^n + (1-\omega)\phi_{i,j}^{n+1} \quad \text{with} \quad 1 < \omega < 2$$

Fastest convergence for $\omega \simeq \dfrac{2}{1+\alpha\frac{\pi}{N}}$ (a depends on dim. and BC)

Similar performance with the Conjugate Gradient method. For a PxP grid: exact convergence in P^2 iterations, so formally it is an N^2 algorithm.

In practice, order P iterations are necessary to reach the level of truncation errors.

Multigrid solver for the Poisson equation

Use coarse-grid sampling to speed-up convergence at large scale.

Proposed by Brandt (1973) to solve elliptic problems.

- Use smoothing properties of Jacobi and Gauss-Seidel scheme to reduce high-frequency modes in the error.

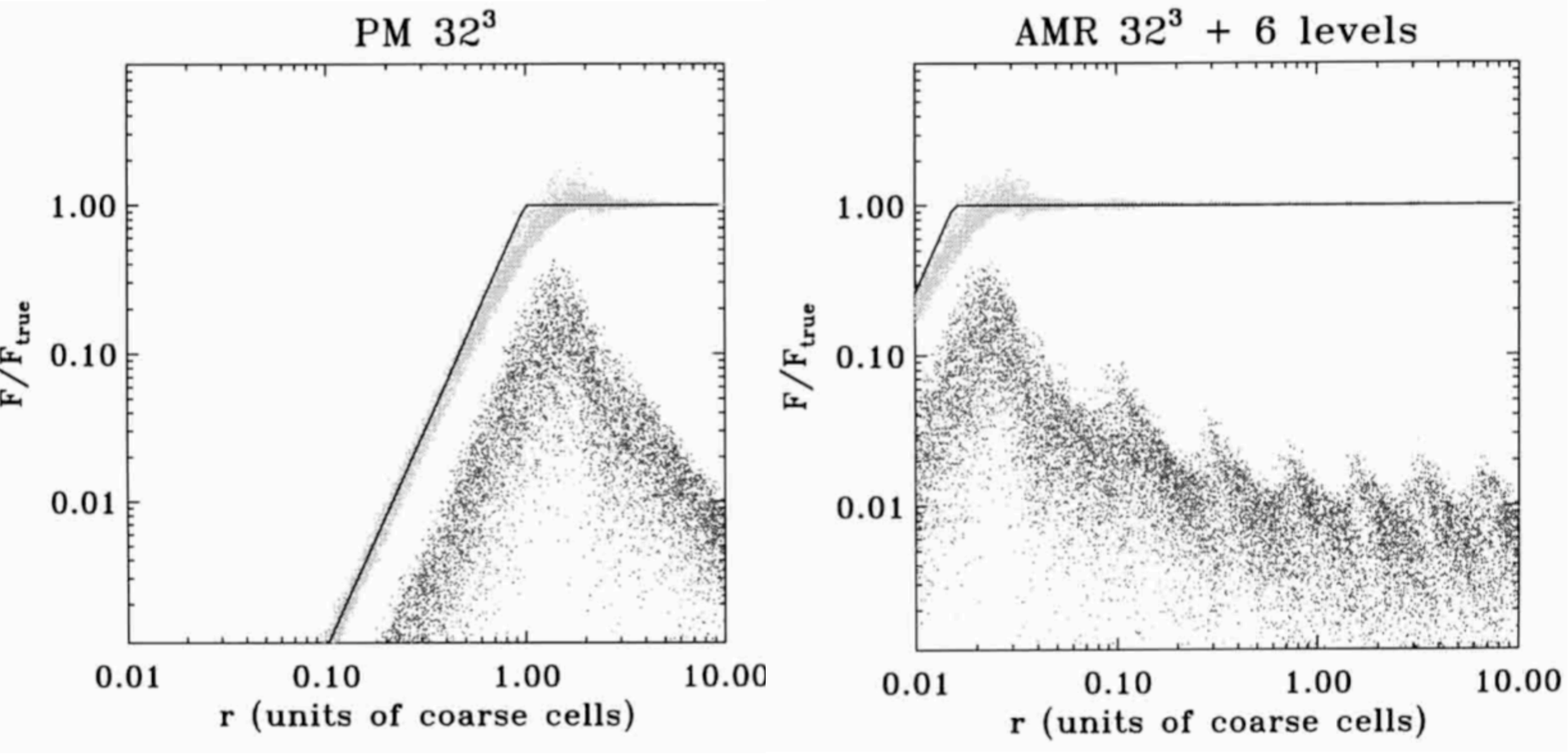
- Use coarsening to reduce low-frequency modes at a faster rate.

Reduce the cost of relaxation solvers from N^2 to N.

Theoretically better than the FFT approach (N log N) !

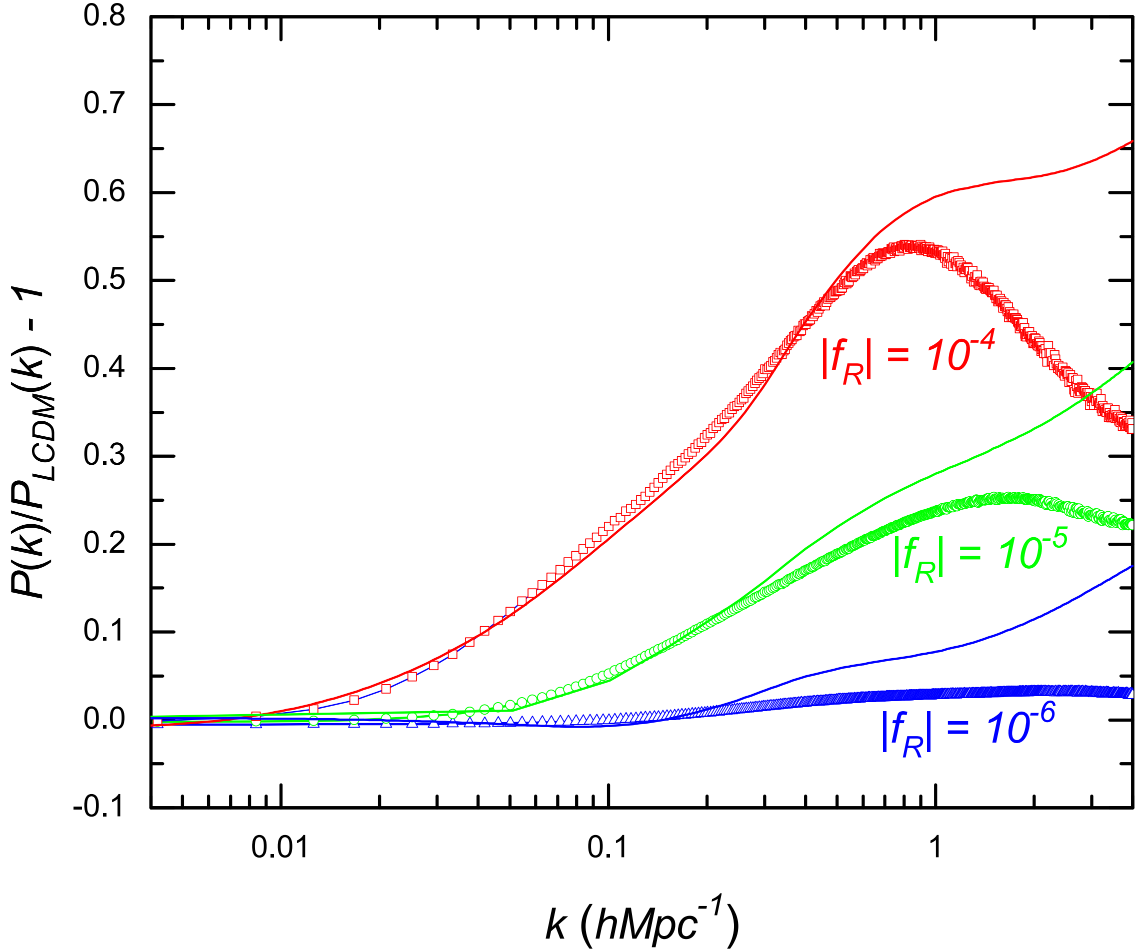
Briggs, W.L., "A Multigrid Tutorial", SIAM Monograph, (2000)

Overall Adaptive PM force accuracy



Beyond standard gravity models ?

RAMSES with f(R) modified gravity



Baojiu Li *et al.* 2011

Non-linear multigrid relaxation on each AMR levels.This gives much faster convergence even under much more stringent convergence criteria associated with such non-linear elliptic problems.

Cosmological initial conditions

Download Gaussian random fields generators from various sources:
- original code from Ed Bertschinger : http://web.mit.edu/edbert/grafic2.101.tar.gz
- MPI version from Simon Prunet : http://www2.iap.fr/users/pichon/mpgrafic.html
- C++ MPI version from Doug Potter: http://sourceforge.net/projects/grafic/
- MUSIC: an IC generator by Oliver Hahn: http://www.stanford.edu/~ohahn/

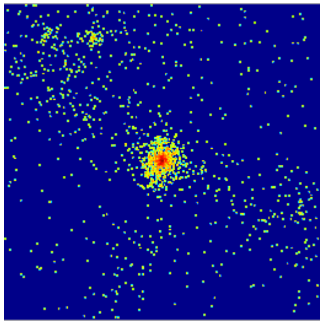
Note: grafic1 and mpgrafic generate only periodic unigrid IC.
grafic2, grafic++ and music generate nested-grid IC: zoom simulations.
mpgrafic described in Prunet et al., ApJS, 2008, 178, 179
music described in Hahn & Abel, MNRAS, 2011, 415, 210

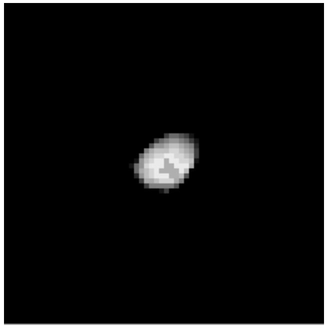
Cosmological inputs are:
- analytical power spectrum from Eisenstein & Hu, ApJ, 1998, 496, 605
- cosmo parameters: omega_m, omega_lambda, omega_b, n_s, sigma_8
- run parameters: box size, grid size, noise random seed or external white noise file
grafic format features 7 binary unformatted fortran files:
ic_velcx, ic_velcy, ic_velcz, ic_deltab, ic_velbx, ic_velby, ic_velbz

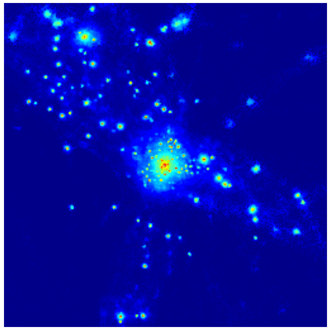
Cosmological zoom initial conditions

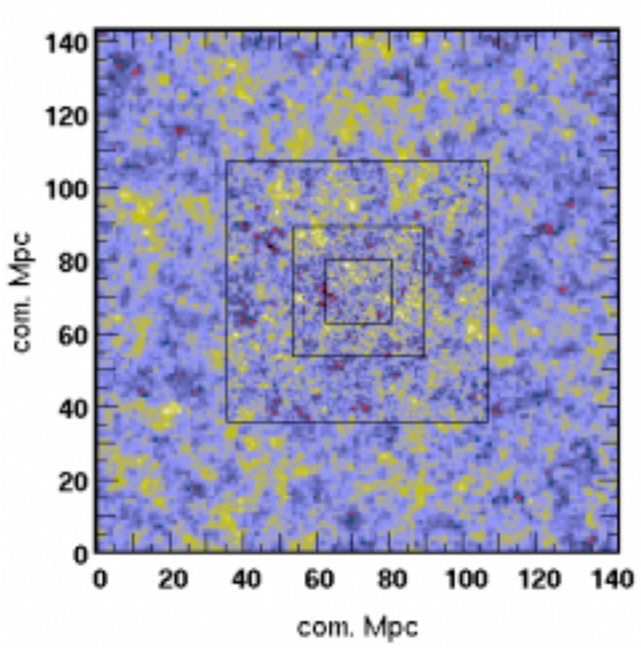
1: detect first one halo of interest in a cosmological simulation.

2: compute the Lagrangian volume in the low resolution IC







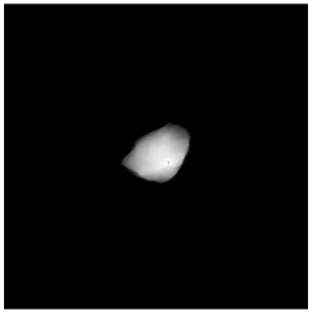


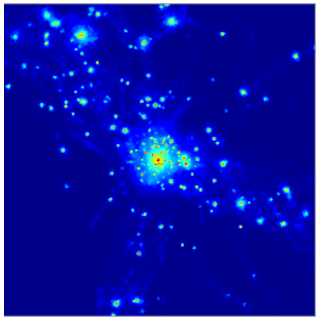
3: generate high-resolution IC by adding high frequency waves to the low resolution initial Gaussian random field

4: use the Lagrangian volume as a map to initialize high resolution particles.

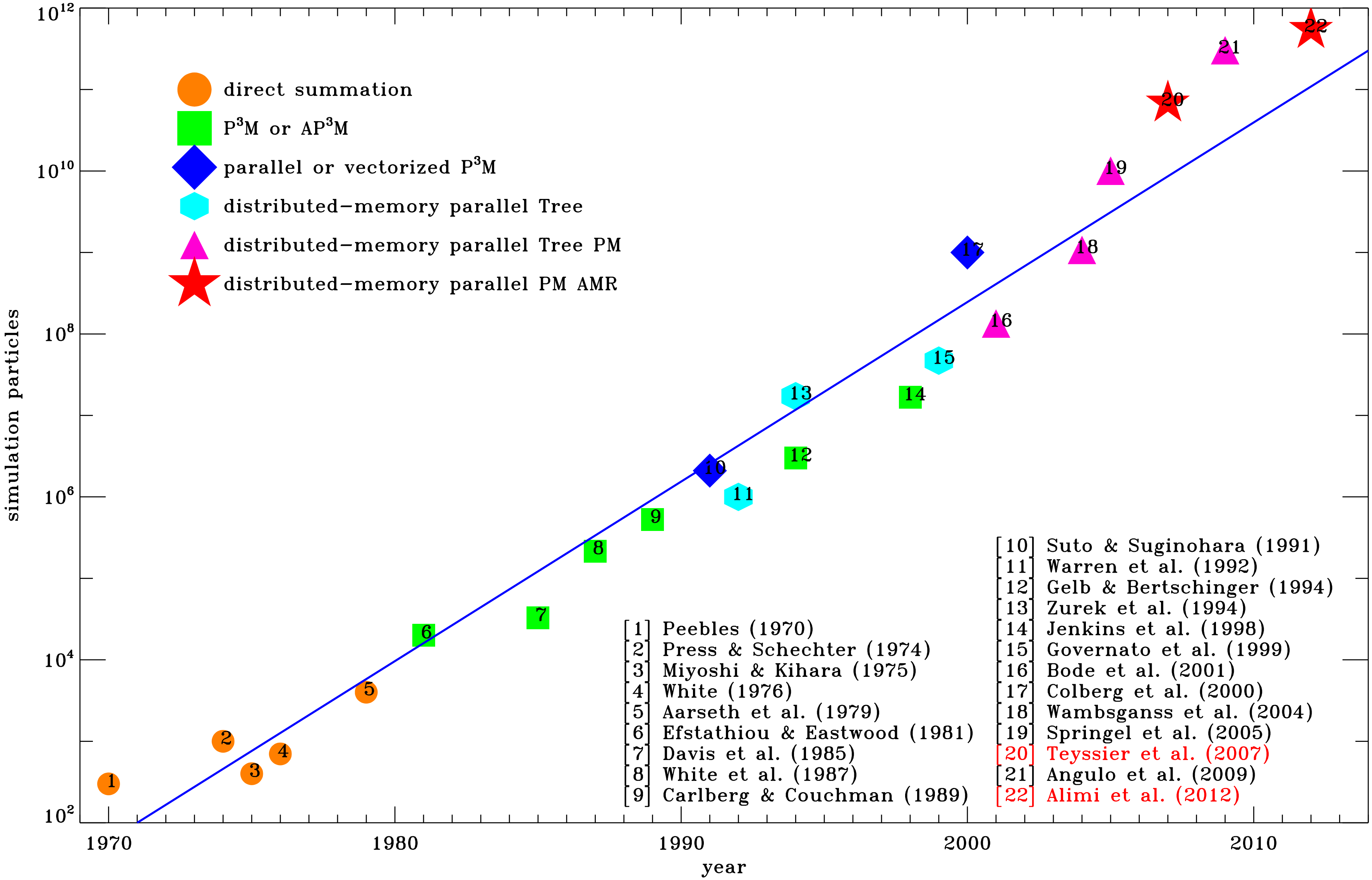
5: do the high resolution simulation and check for contamination

6: eventually, compute a better initial Lagrangian volume and re-do the simulation

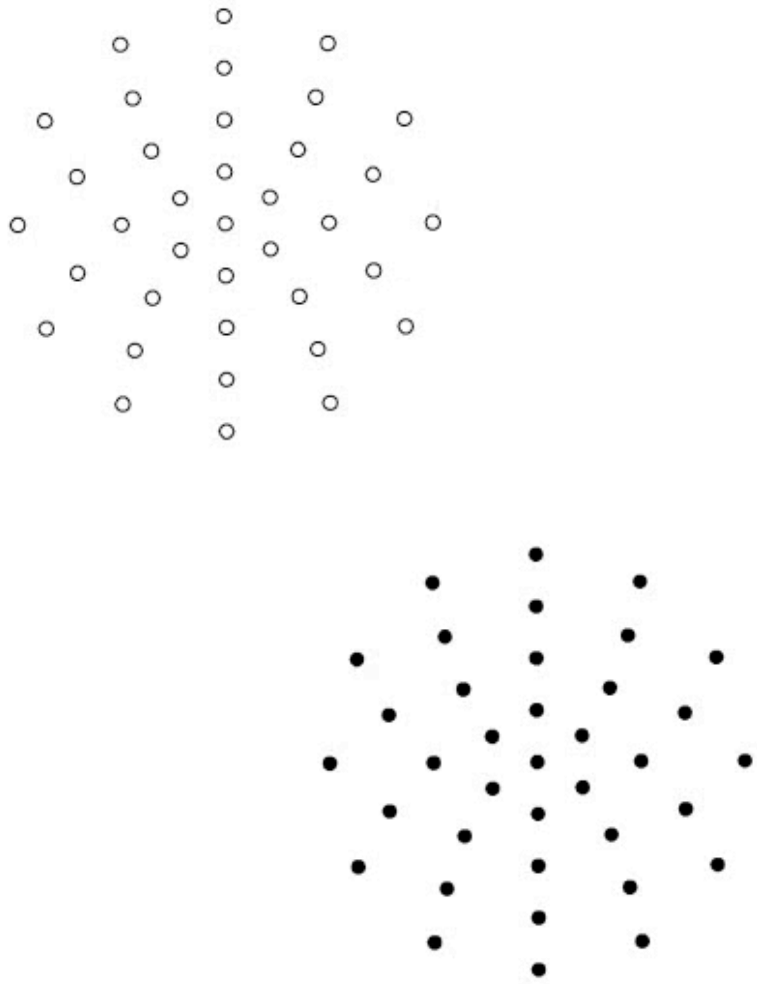




Cosmological N body simulations



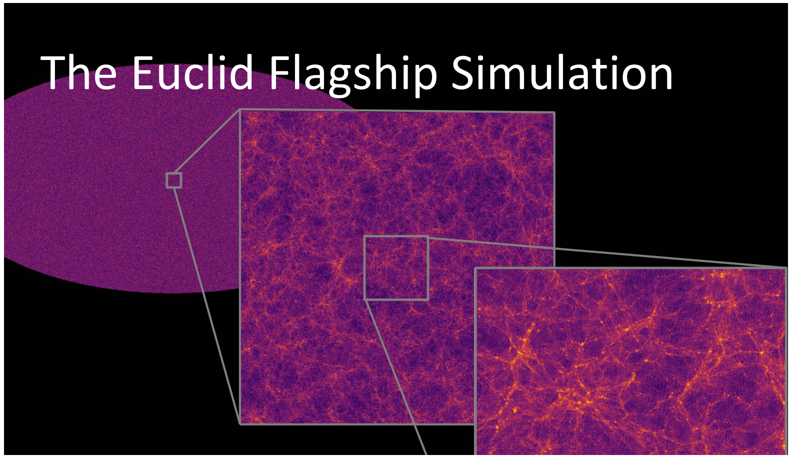
The first N body simulation ever



Cosmological simulations: computing requirements

Mock galaxy catalogues: one simulation every year with 10T particles. Galaxy population on the light cone with HOD/AM/SAM techniques with lensing maps.

Resources: 2 million node-hours (with GPU)



Potter, Stadel, Teyssier 2017

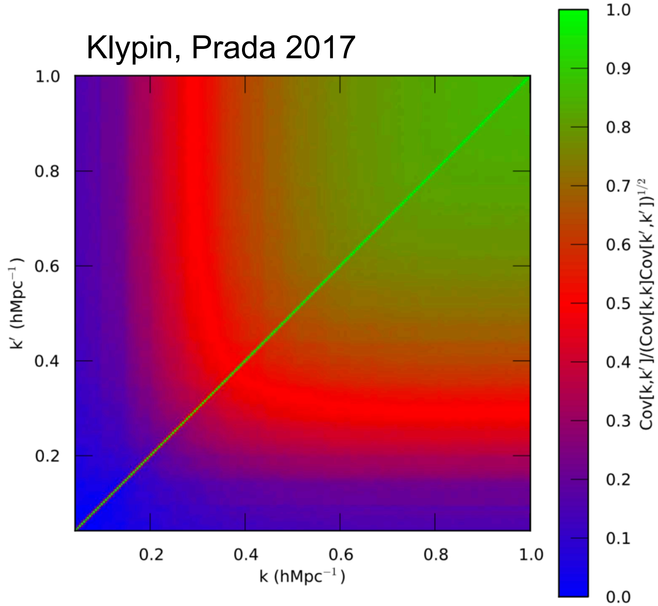
Emulators: 50 such simulations (one per cosmological parameter set)

Coyote: Heitmann et al. 2014, Mira-Titan; Heitmann et al. 2016

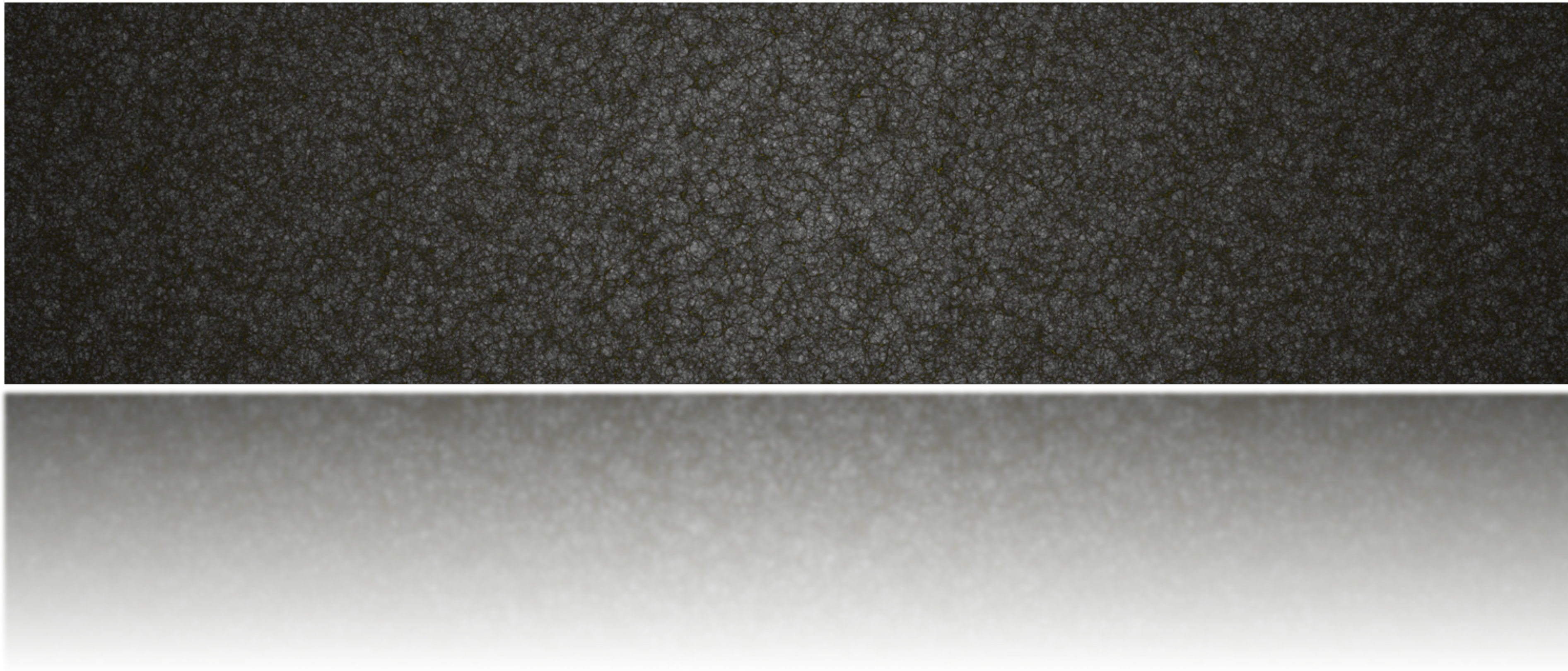
Covariance matrices: 3000 simulations with 8B particles every year.

Resources: 2 million node-hours (with GPU)

Klypin, Prada 2017

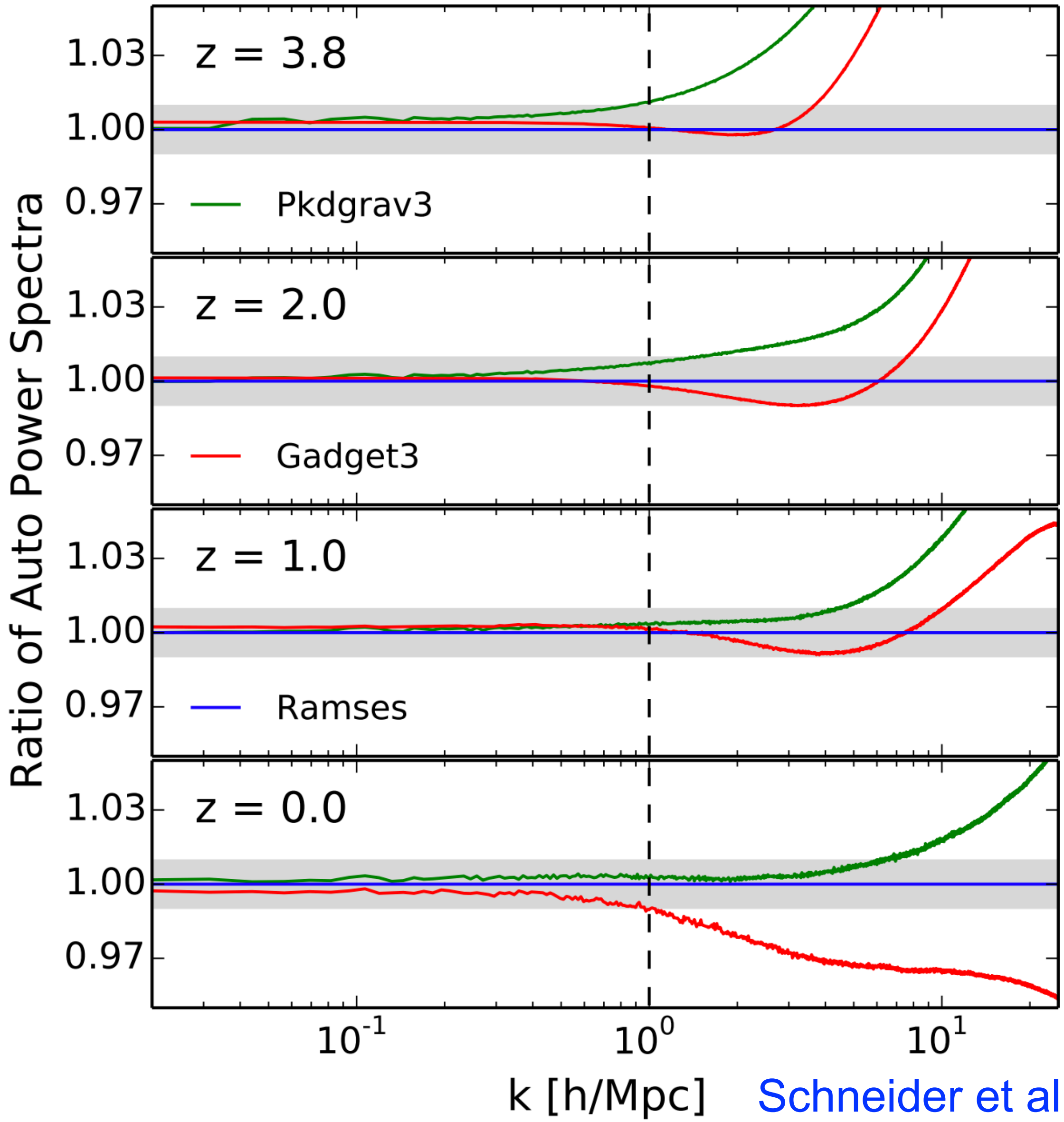


The EUCLID Flagship simulation (version 2)



- Massive neutrinos and GR effects (linear)
- Self-consistent light cone (no more back-scaling)
- 4.1 trillion particles (resolution 10^9 h^{-1}Msol)
- 3’600 h^{-1}Mpc box size
- 835’000 GPU node-hours on Piz Daint

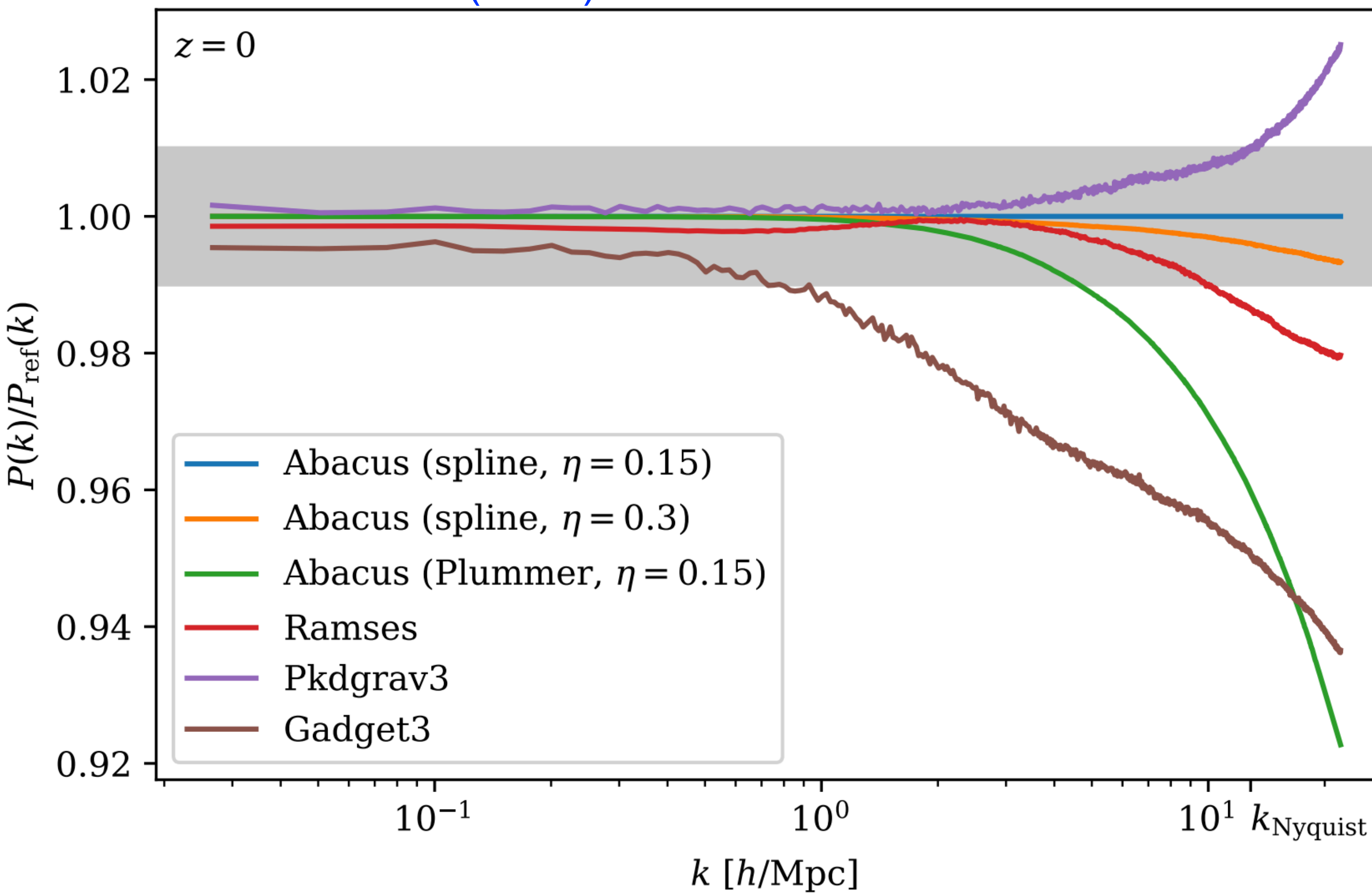
Systematic errors in N body codes ?



Schneider et al. (2016)

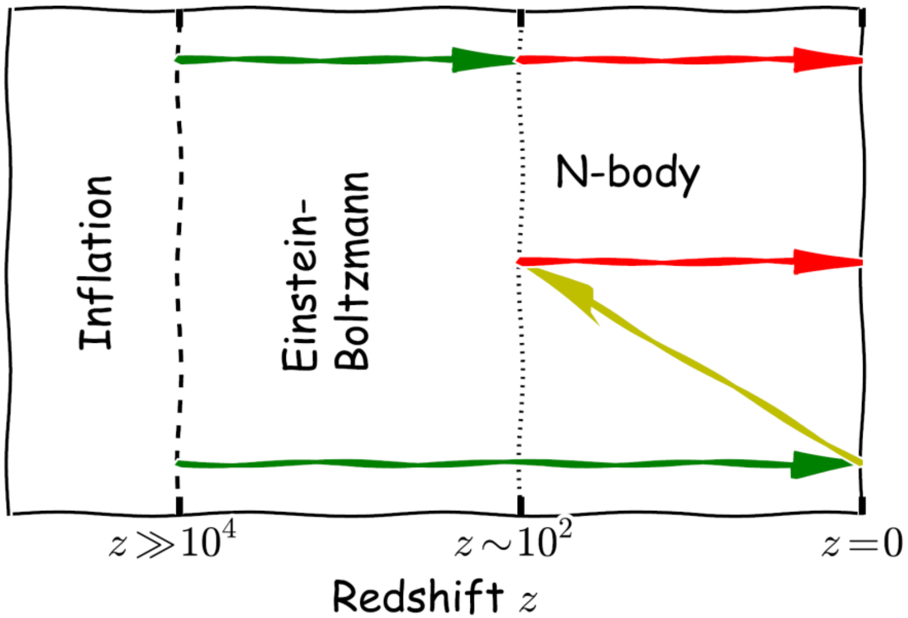
Systematic errors in N body codes ?

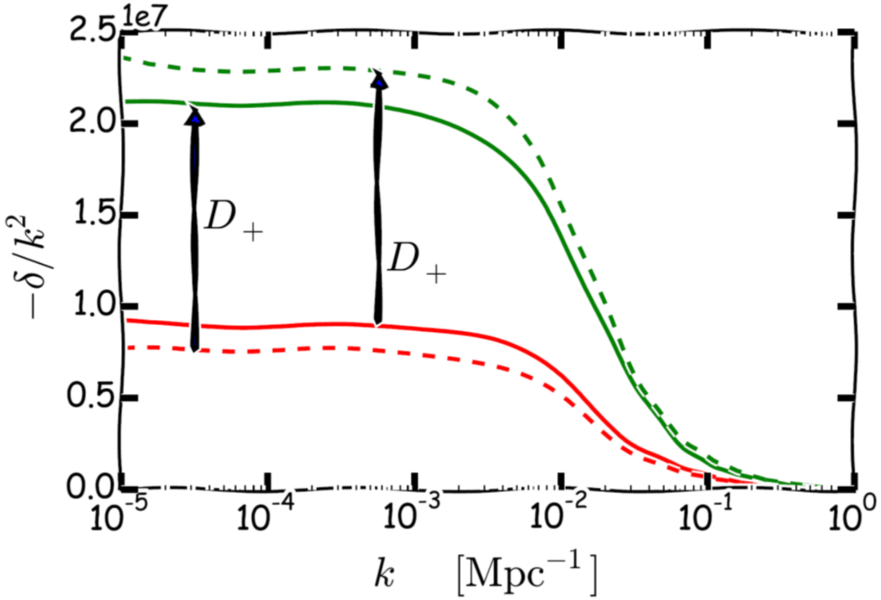
Garrison et al. (2018)



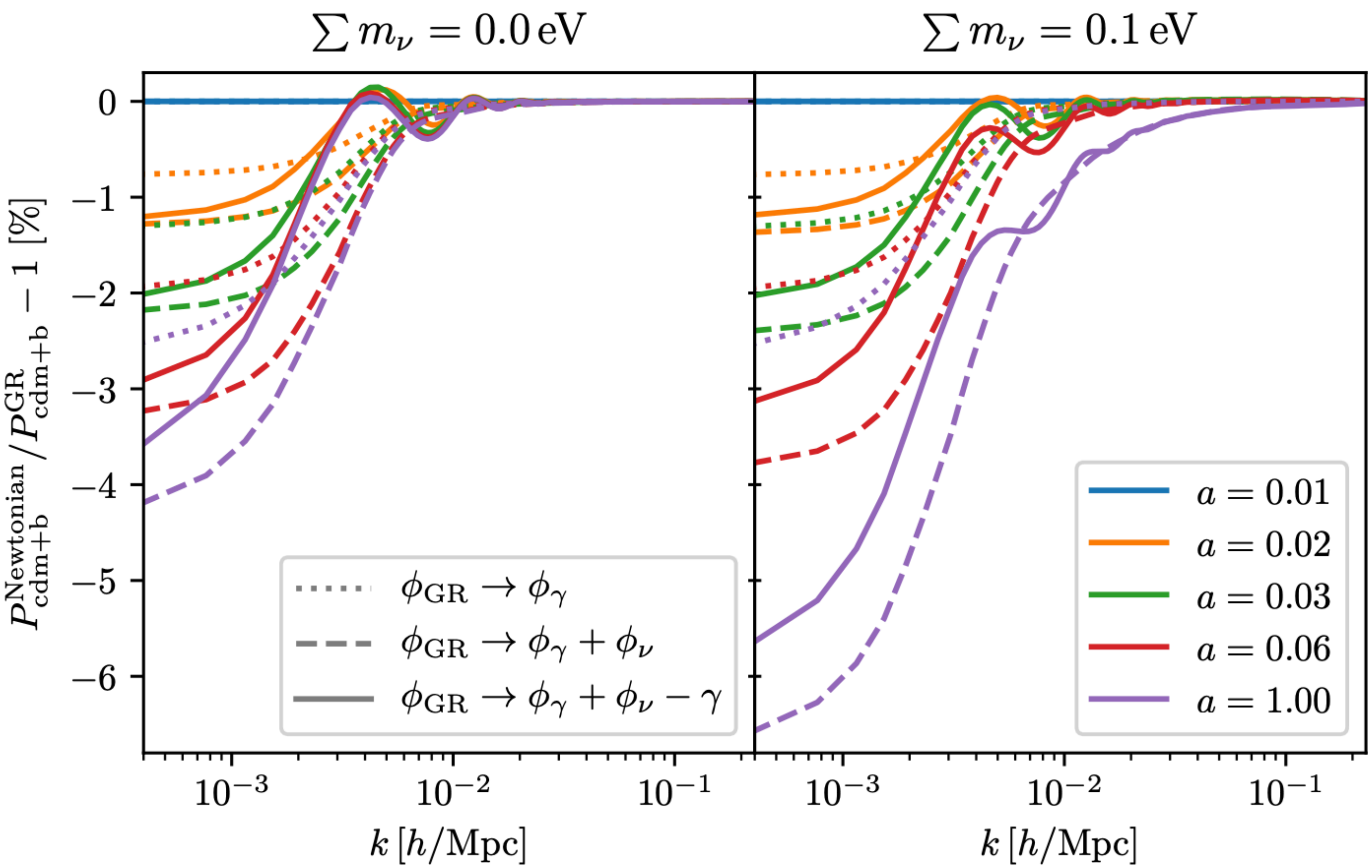
Back-scaled or forward initial conditions?

Fidler *et al.* 2017





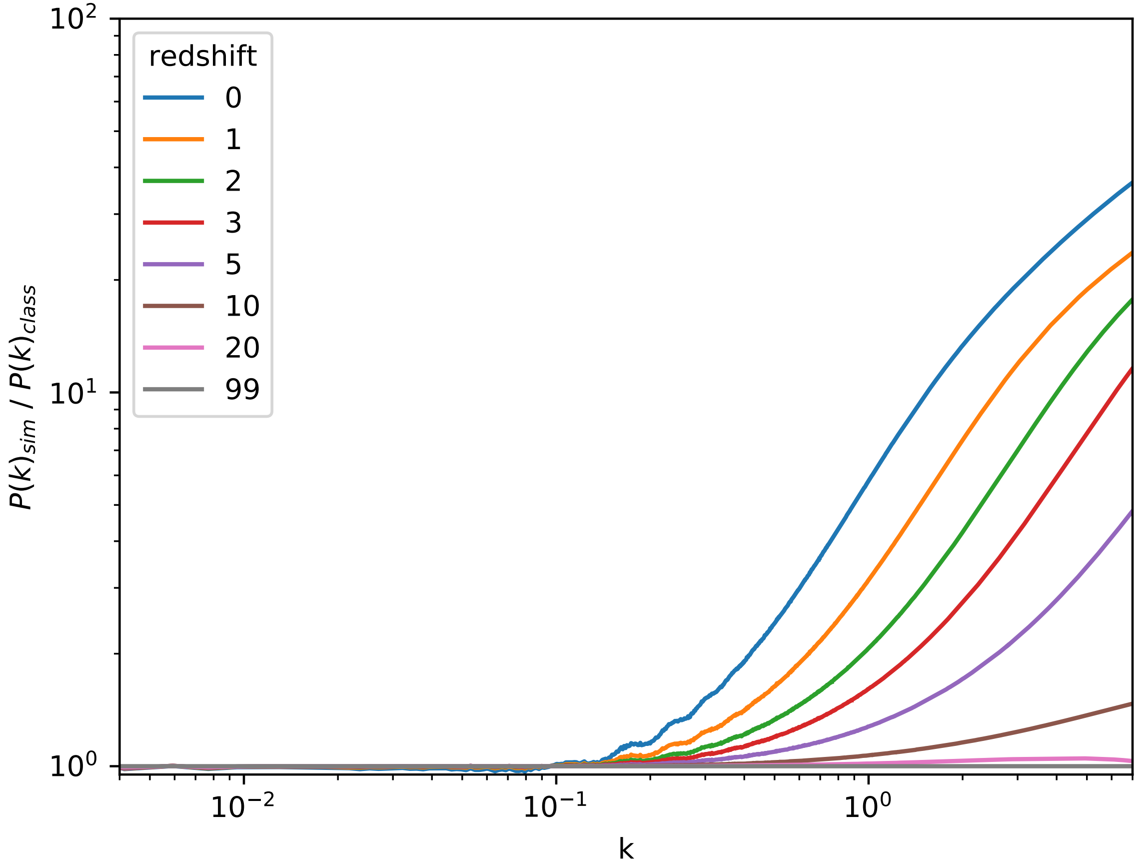
Evolution of GR corrections over time



N-body gauge with photons, neutrinos and metric linear fluctuations generated with CLASS
Tram *et al.* 2018

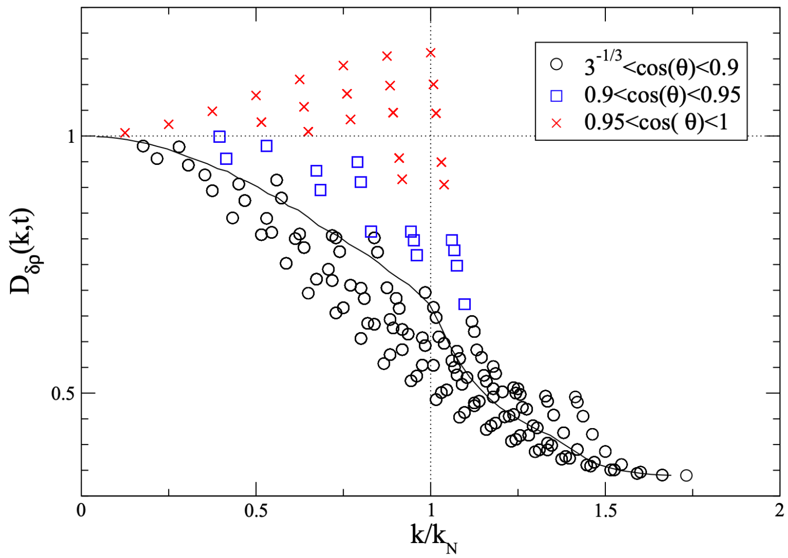
Bridging properly linear and non-linear scales

Flagship 2 power spectrum non-linear correction

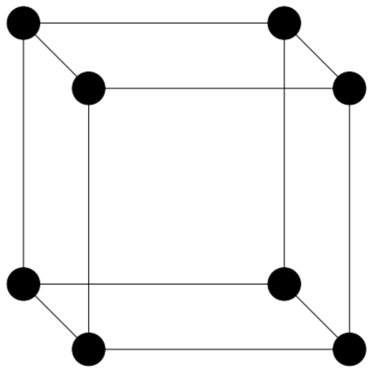


Particle discreteness effects

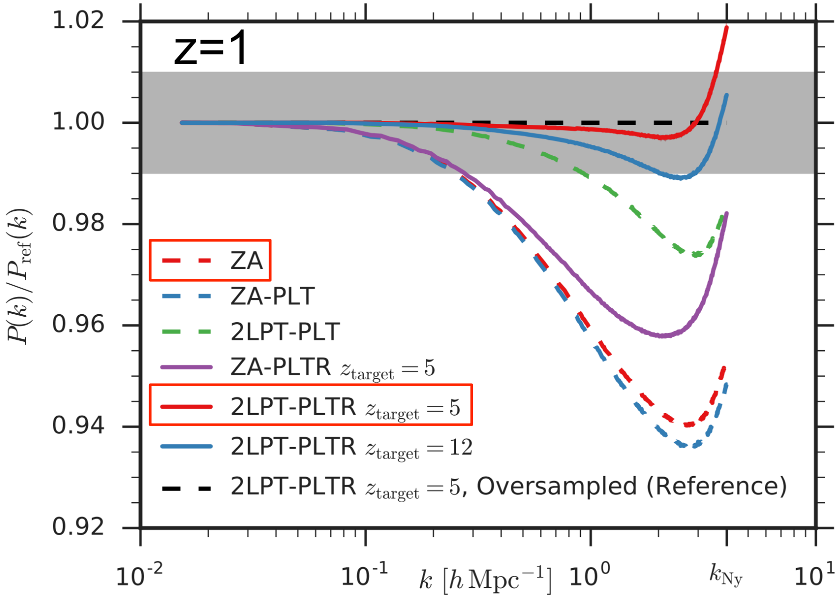
Joyce *et al.* 2008



simple cubic lattice



Garrison *et al.* 2018



Particle linear theory can be used to rescale the initial conditions and correct from discreteness effects.

Sub-percent accuracy down to the Nyqvist frequency