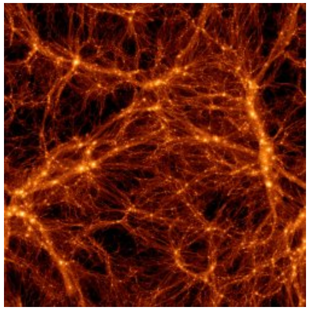
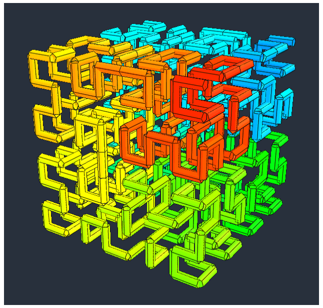
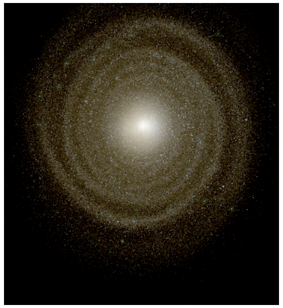
High-performance N-body simulations for modelling large-scale structures in the Universe

Romain Teyssier, Joachim Stadel, Doug Potter

University of Zurich







Outline

- Cosmological simulations: basic principles

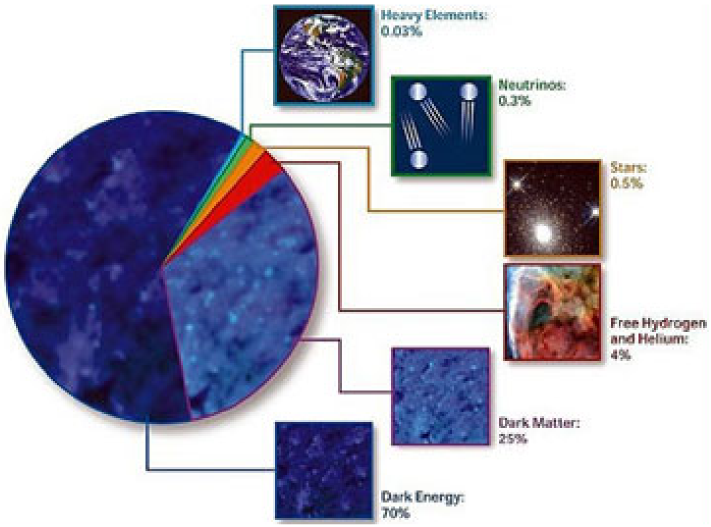
- Solving the Vlasov-Poisson equations with N body

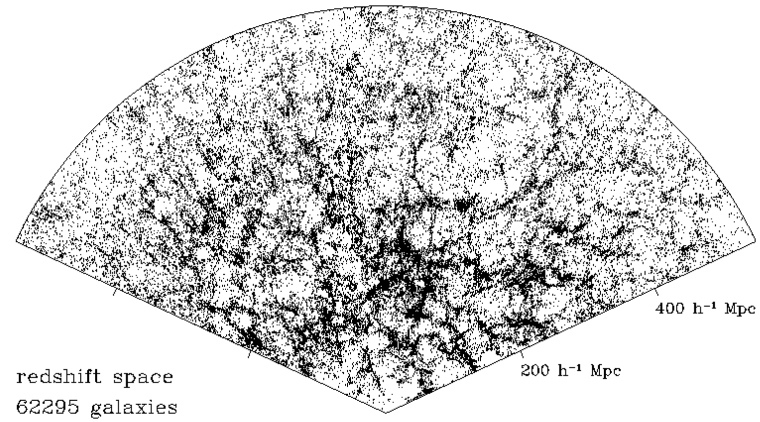
-Multigrid solver: beyond the Poisson equation

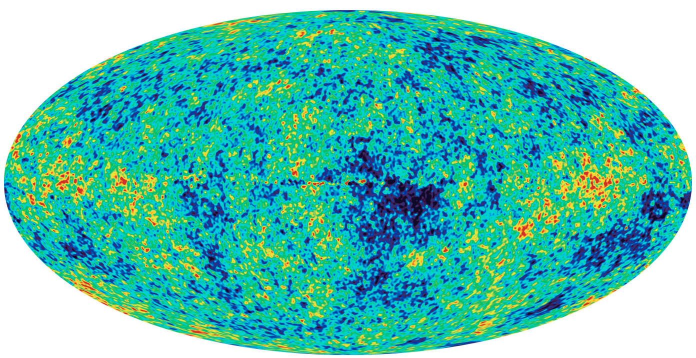
- A brief history of cosmological simulations

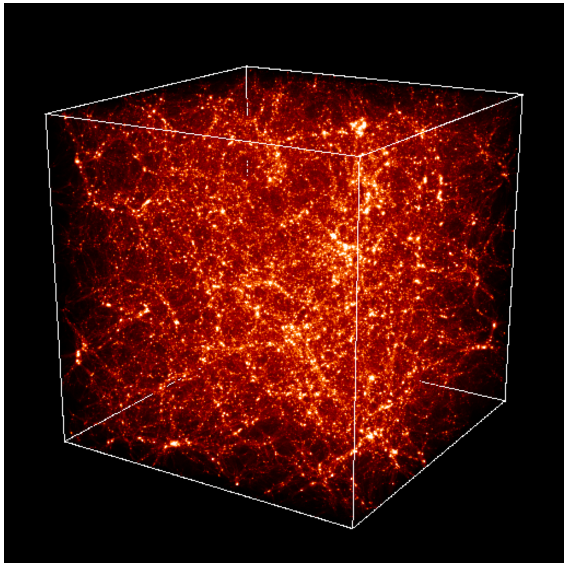
- The Euclid Flagship Simulation: a new high performance simulation in cosmology using fast multiple method and GPU computing

Cosmological simulations



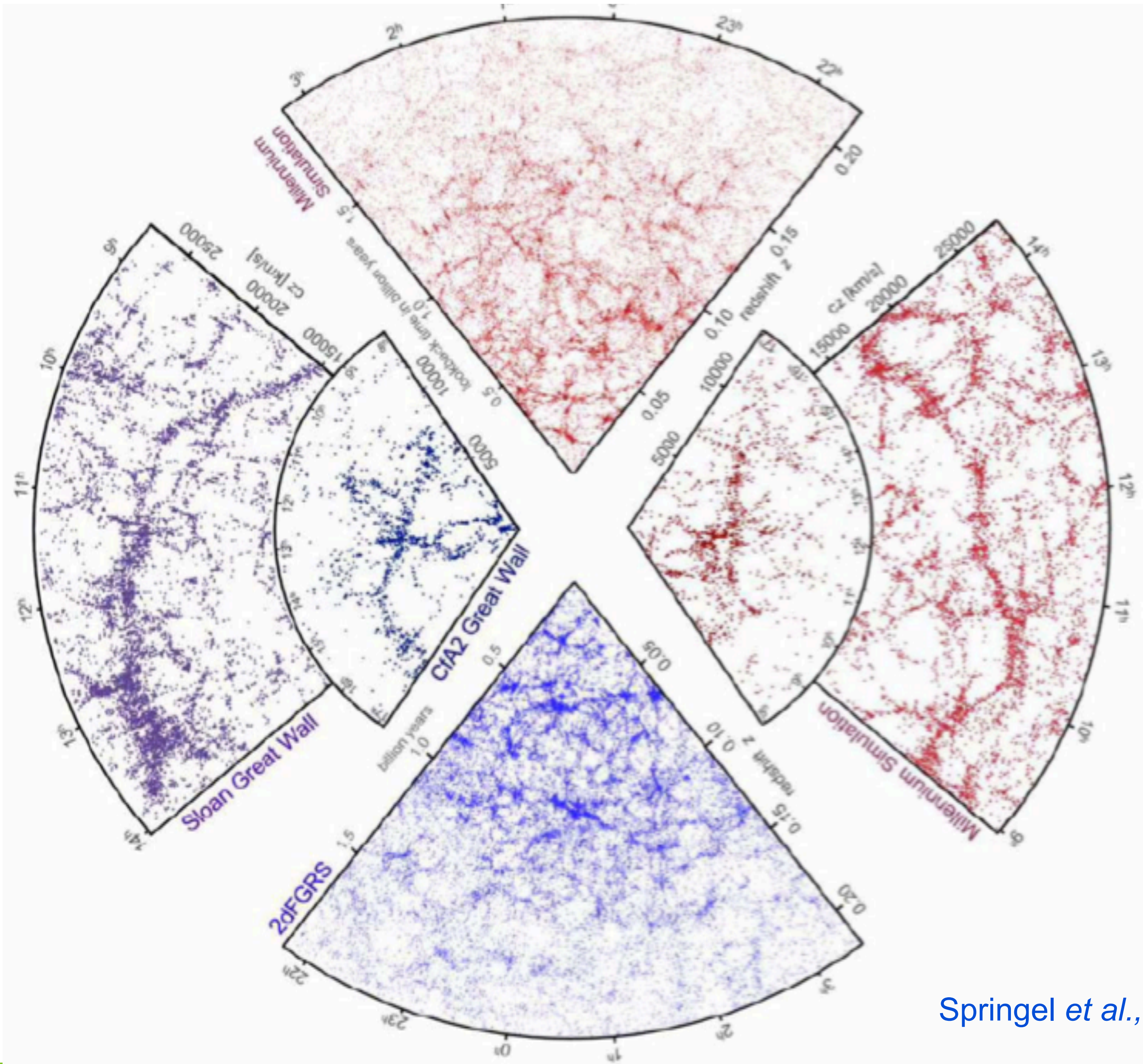






From Gaussian random fields to galaxies: non-linear dynamics of gravitational instability with N-body and hydrodynamics codes.

Cosmological simulations



Springel *et al.*, Nature, 2006

The Vlasov-Poisson equation

Collisionless limit of the Boltzmann equation in an expanding universe:

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

Gravitational acceleration is given by the Poisson equation

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

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, tree codes or fast multipole methods (Barnes & Hut 1986; Bouchet & Hernquist 1988, Dehnen 2014)

- mixture of the 2: the Particle-Mesh method, the P3M method (Hockney & Eastwood 1988), the Adaptive PM and P3M methods ()

The N body method in a nutshell

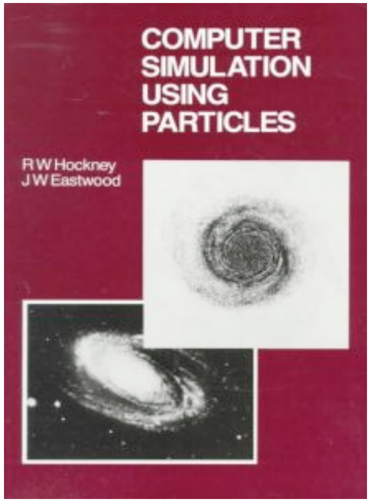
N body integrator coupled to a grid-based Poisson solver

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

1- Deposit the mass on a grid from the particle distribution

2- Solve (directly or on the mesh) for the gravitational potential

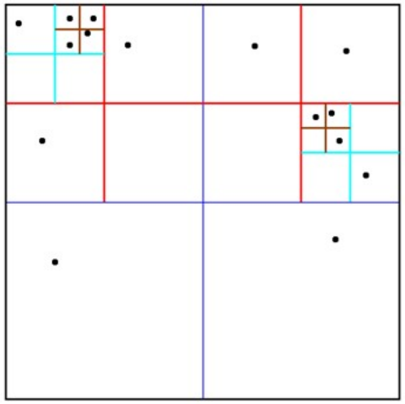
3- Interpolate the force back to the particle position



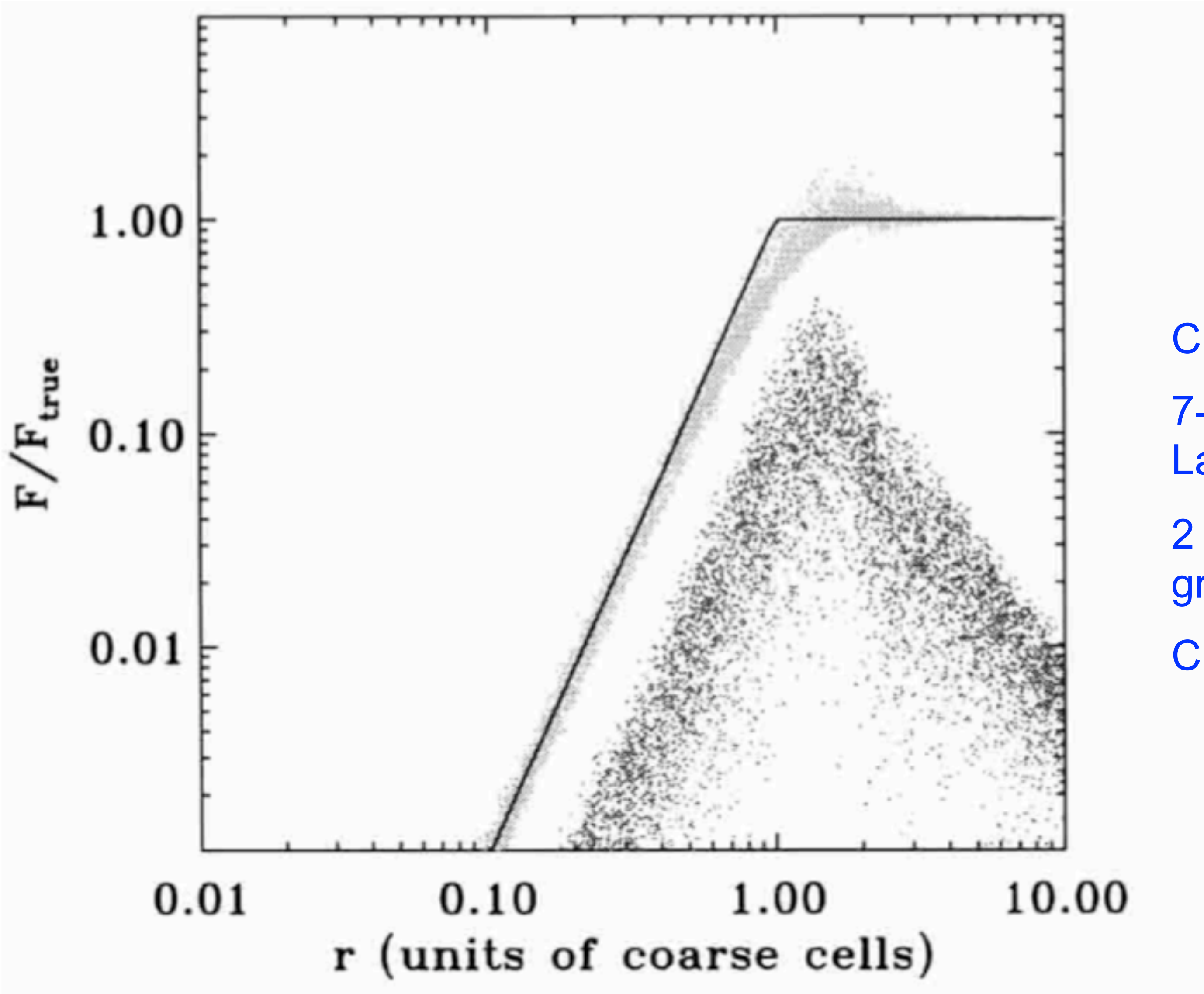
Gravity solvers:

- direct force or Particle-Particle (PP) interactions: order N^2
- tree code or Particle-Cell interactions: order N log N
- Particle-Mesh and FFT (PM): order N log N
- Adaptive Mesh and Multigrid: order N
- Fast Multipole Method or Cell-Cell interactions: order N

Adaptive quadtree where no square con



Overall PM force accuracy



CIC

7-point Laplacian

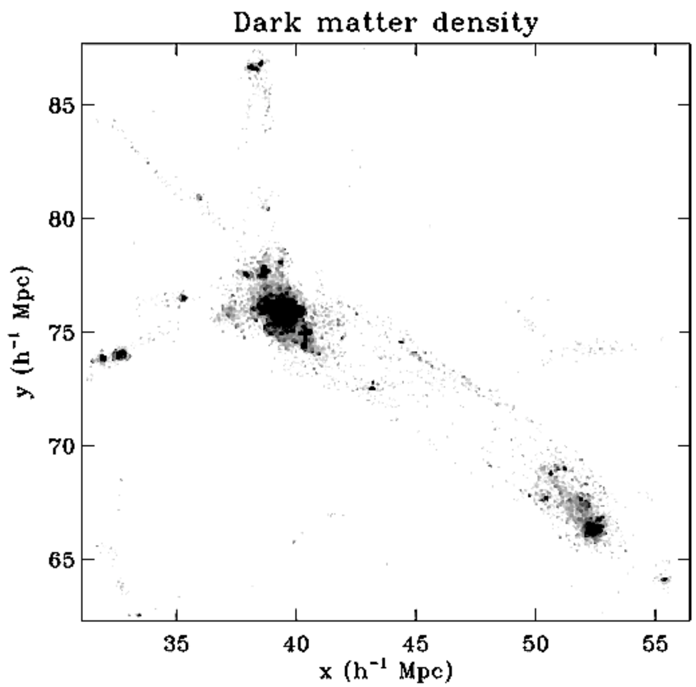
2 points gradient

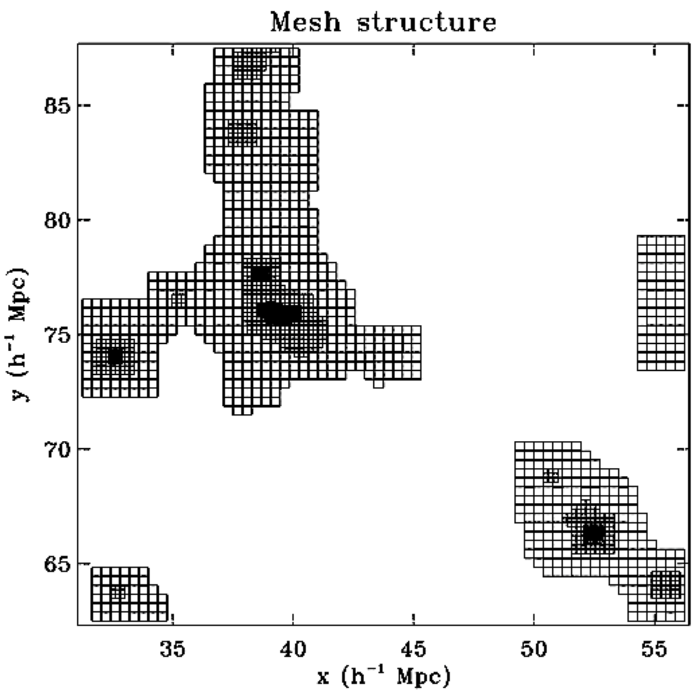
CIC^{-1}

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.





ART: a one-way interface Poisson solver

Kravtsov et al. 1997, ApJS, 111, 73

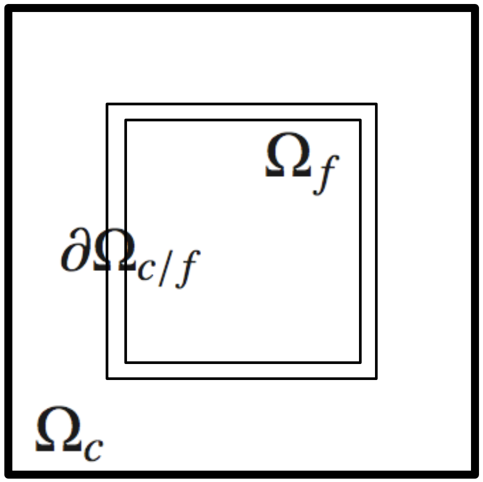
Fine-level particles are temporarily passed to the coarse-level list to compute ρ_c .

Fine-to-coarse information is mediated by particles only.

Solve $\Delta\Phi_c = \rho_c$ on Ω_c

Interpolate BC on $\partial\Omega_{c/f}$

Solve $\Delta\Phi_f = \rho_f$ on Ω_f



Coarse-to-fine information is propagated by the potential through Dirichlet BCs.

Third order interpolation ensures that truncation errors are globally second order.

Self-gravity with gas dynamics: in each coarse-level cell, the center-of-mass is computed and its contribution to ρ_c is added with CIC interpolation (proposed by Miniati & Colella, JCP, 2007)

The multigrid method: two-grid scheme

On the fine grid, define the residual $r_\ell^n = \Delta_\ell \Phi_\ell^n - \rho_\ell$

and the error $e_\ell^n = \Phi_\ell^n - \Phi_\ell^\infty$

$$\Delta_\ell e_\ell^n = r_\ell^n$$

1- Perform a few GS iterations (smoothing).

2- Restrict the residual to the coarse grid:

$$r_\ell^n \longrightarrow r_{\ell-1}^0$$

3- Solve for the coarse grid system:

$$\Delta_{\ell-1} e_{\ell-1} = r_{\ell-1}$$

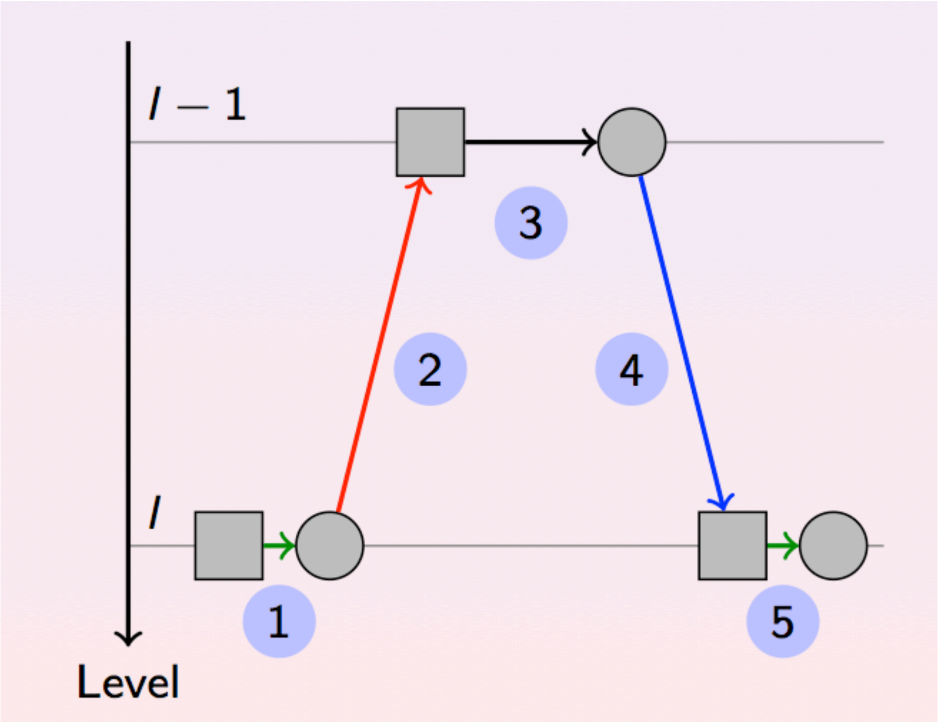
4- Prolong back the error to the fine grid:

$$e_{\ell-1}^\infty \longrightarrow e_\ell^{n+1}$$

5- Correct the fine grid solution:

$$\Phi_\ell^{n+1} = \Phi_\ell^n + e_\ell^{n+1}$$

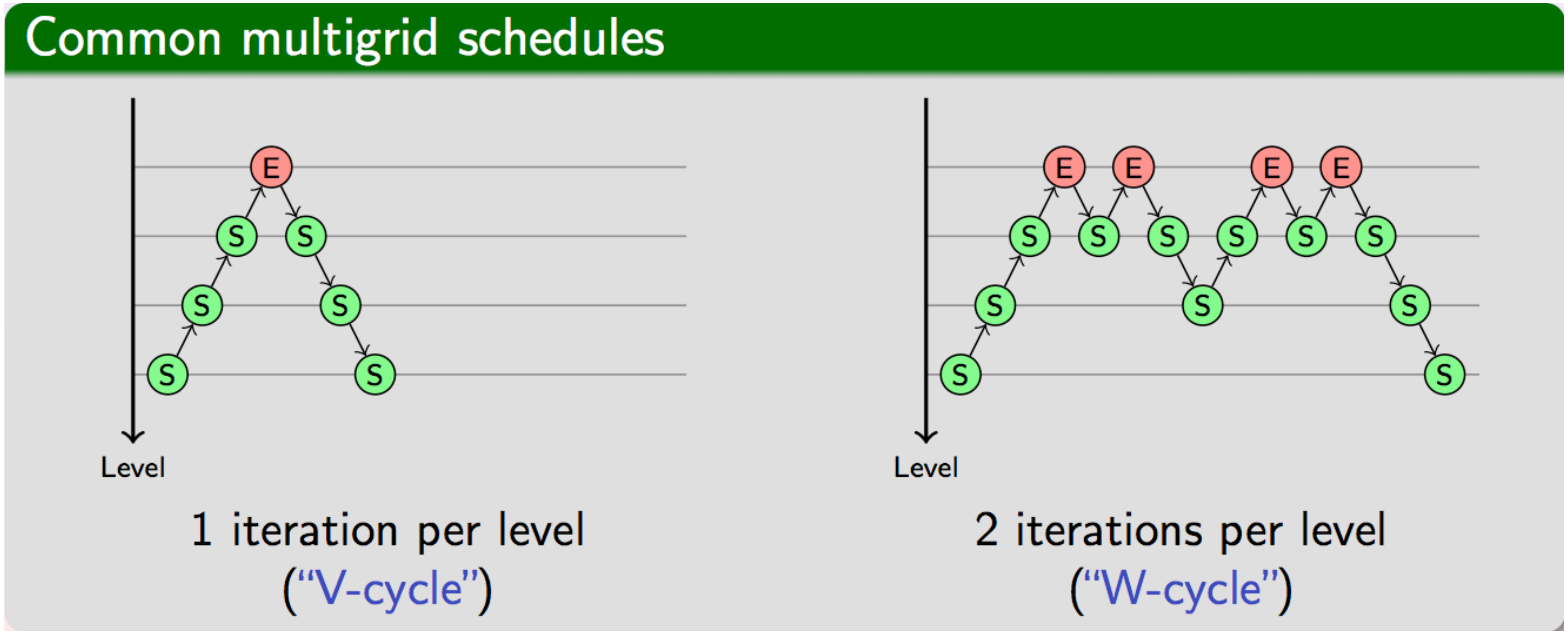
and perform a few GS iteration.



The multigrid method: V or W cycles

Recursively apply the 2-grid scheme. Solve for the exact solution only at the coarsest level.

Iterate once or twice before going to the finer level.



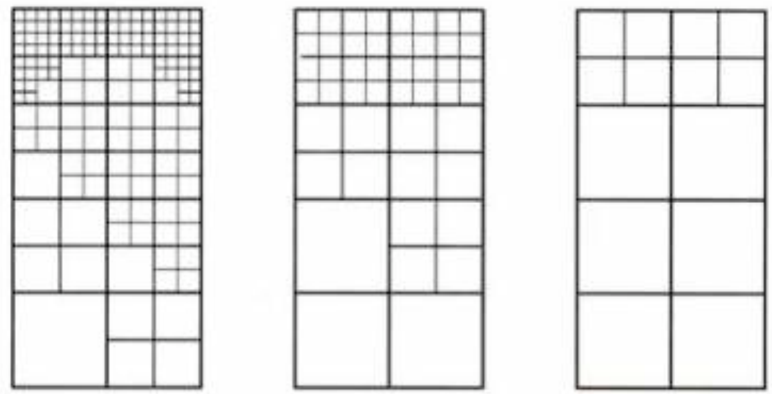
Converge in very few iterations, independently of grid size.

Quasi-insensitive to the quality of the initial guess.

The multigrid method and AMR

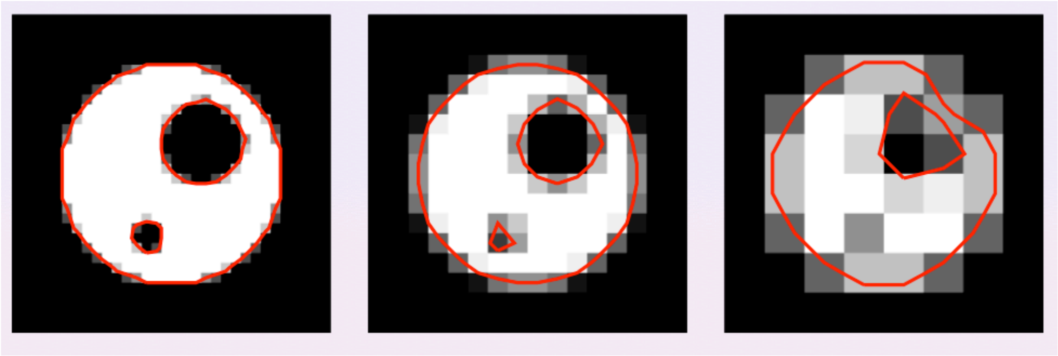
Two-way coupling: one needs to define a ensemble of AMR grids, each AMR grid corresponds to a level in the multigrid hierarchy.

Berger, M., “An adaptive multigrid method for the Euler equations”, Springer Berlin



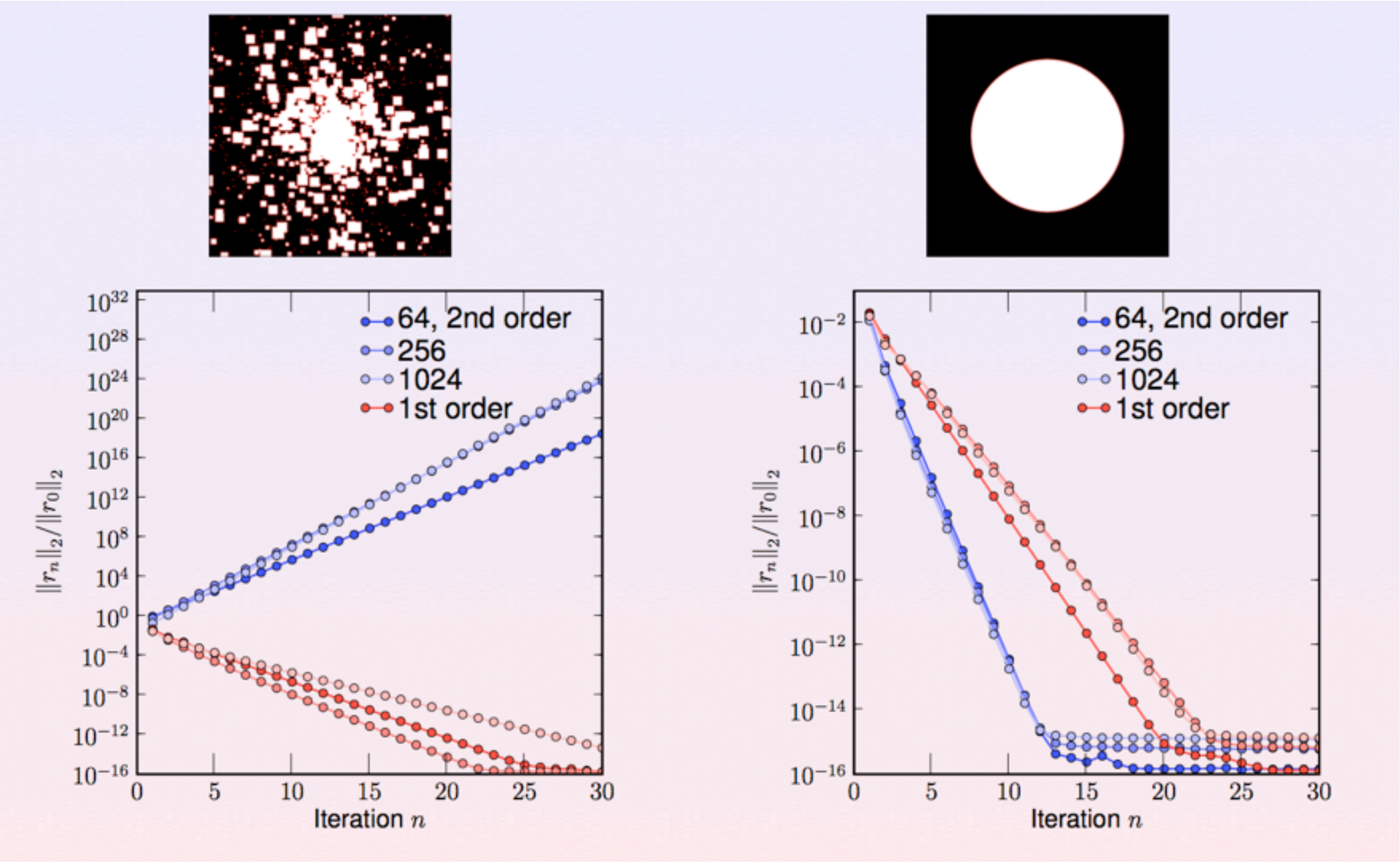
Level 3 Level 2 Level 1

One-way coupling: for each AMR level, one needs to design a multigrid scheme for arbitrary-shaped boundary conditions.

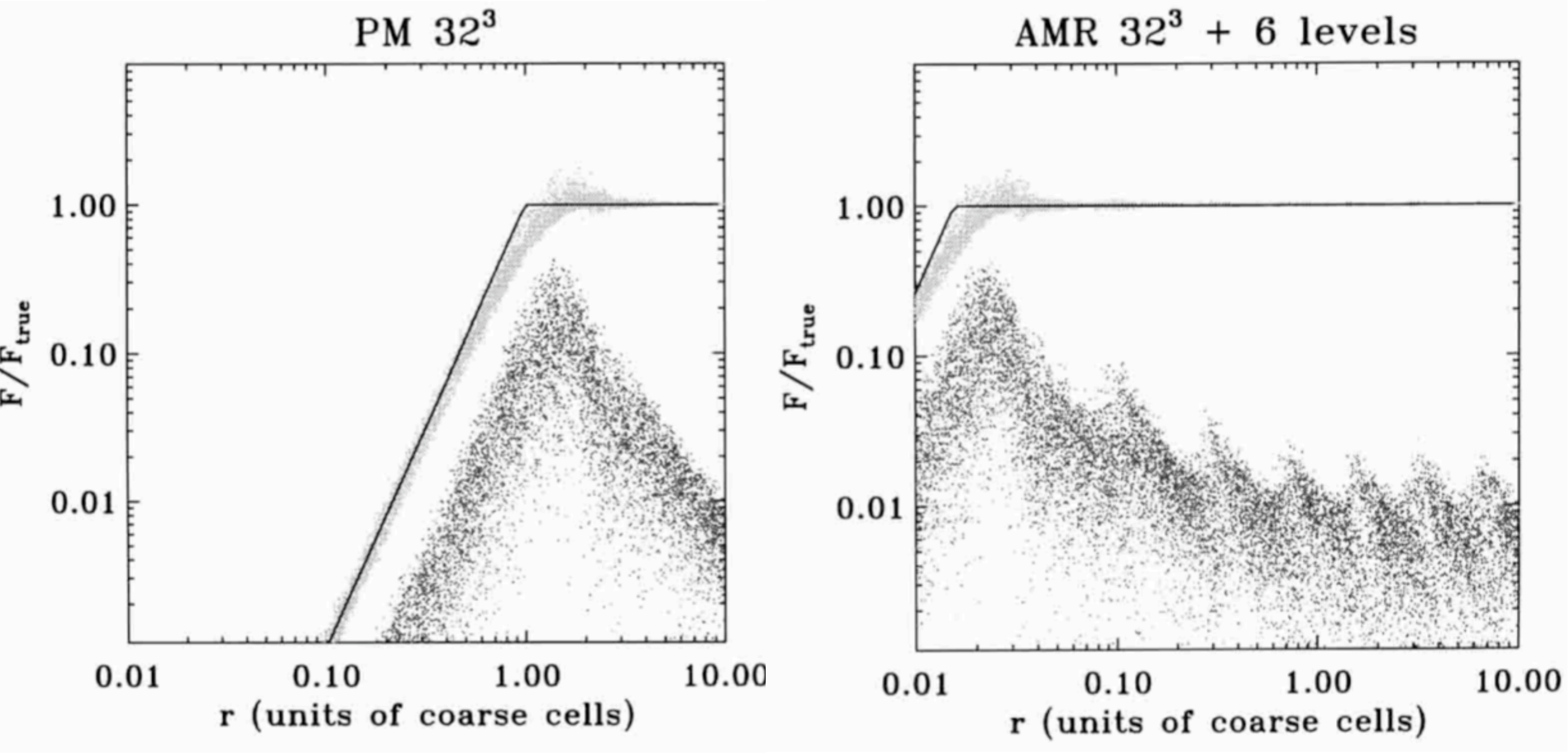


Level 3 Level 2 Level 1

Performance of multigrid on arbitrary domains



Overall Adaptive PM force accuracy



General relativistic effects on large scales

Poisson equation in the weak field limit is valid only deep within the horizon. Close to the horizon, the finite speed of light affects the clustering. It follows a diffusion equation (e.g. Llinares 2017)

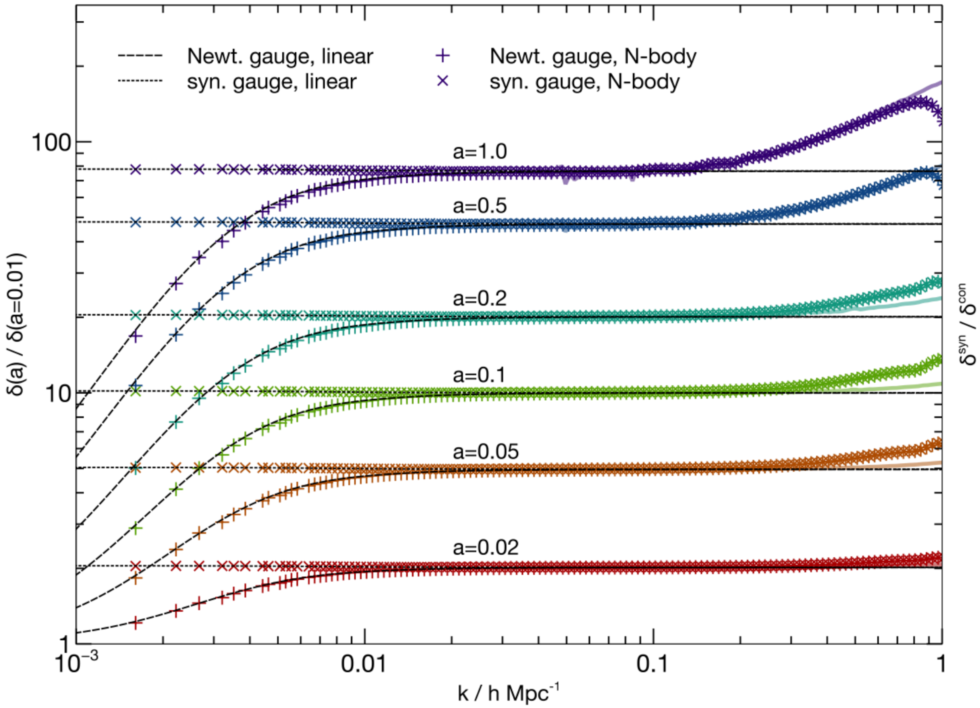
$$3H\dot{\Phi} - c^2\nabla^2\Phi = -c^2\frac{3H_0^2\Omega_m}{2a}\delta,$$

Using an implicit time integration, this diffusion equation can be solved using a multigrid solver.

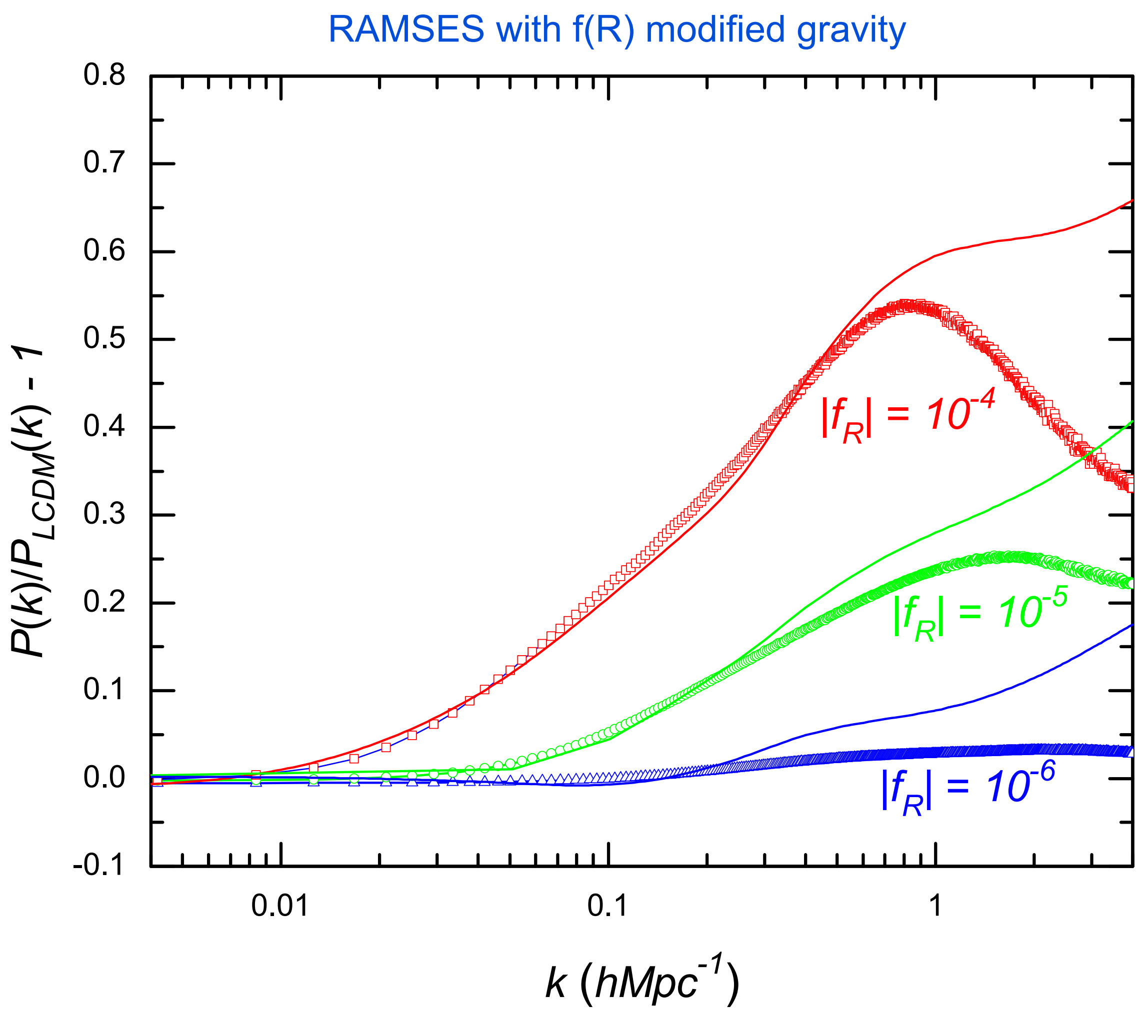
Hahn & Paranjape (2016) showed that it can be approximated by a screened potential on large scale, hence solvable with the fast multiple method.

$$G_{\rm P}(k) = -1/k^2\,,$$

$$G_{\rm H}(k) = -1/(k^2 + \ell^{-2})$$



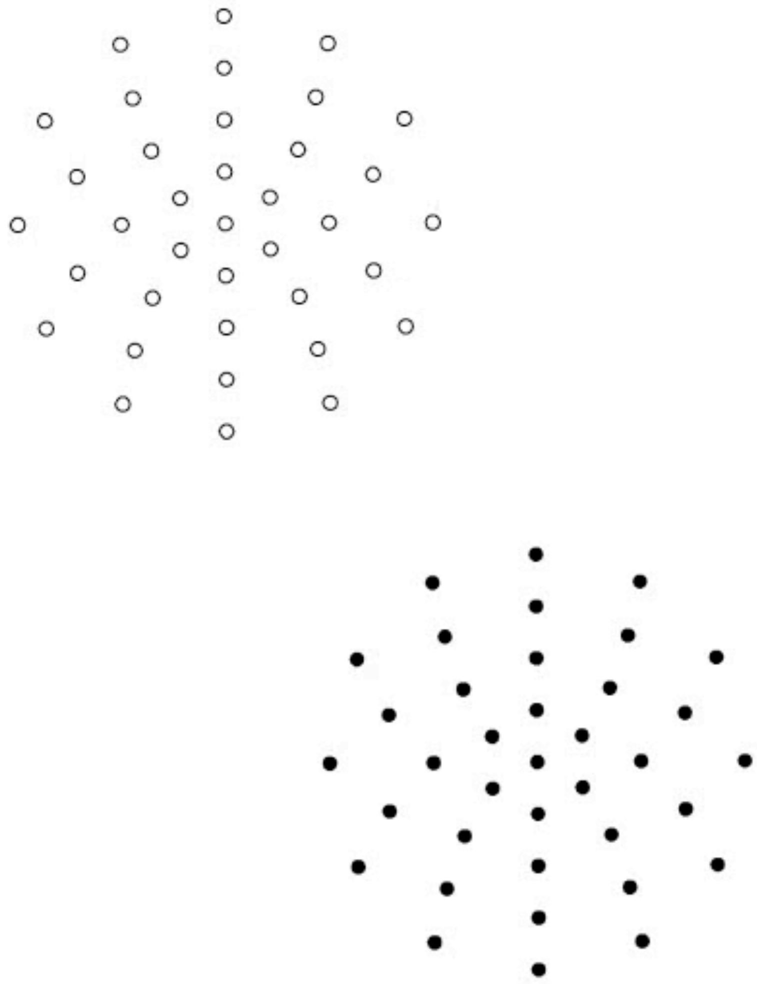
Beyond standard gravity models ?



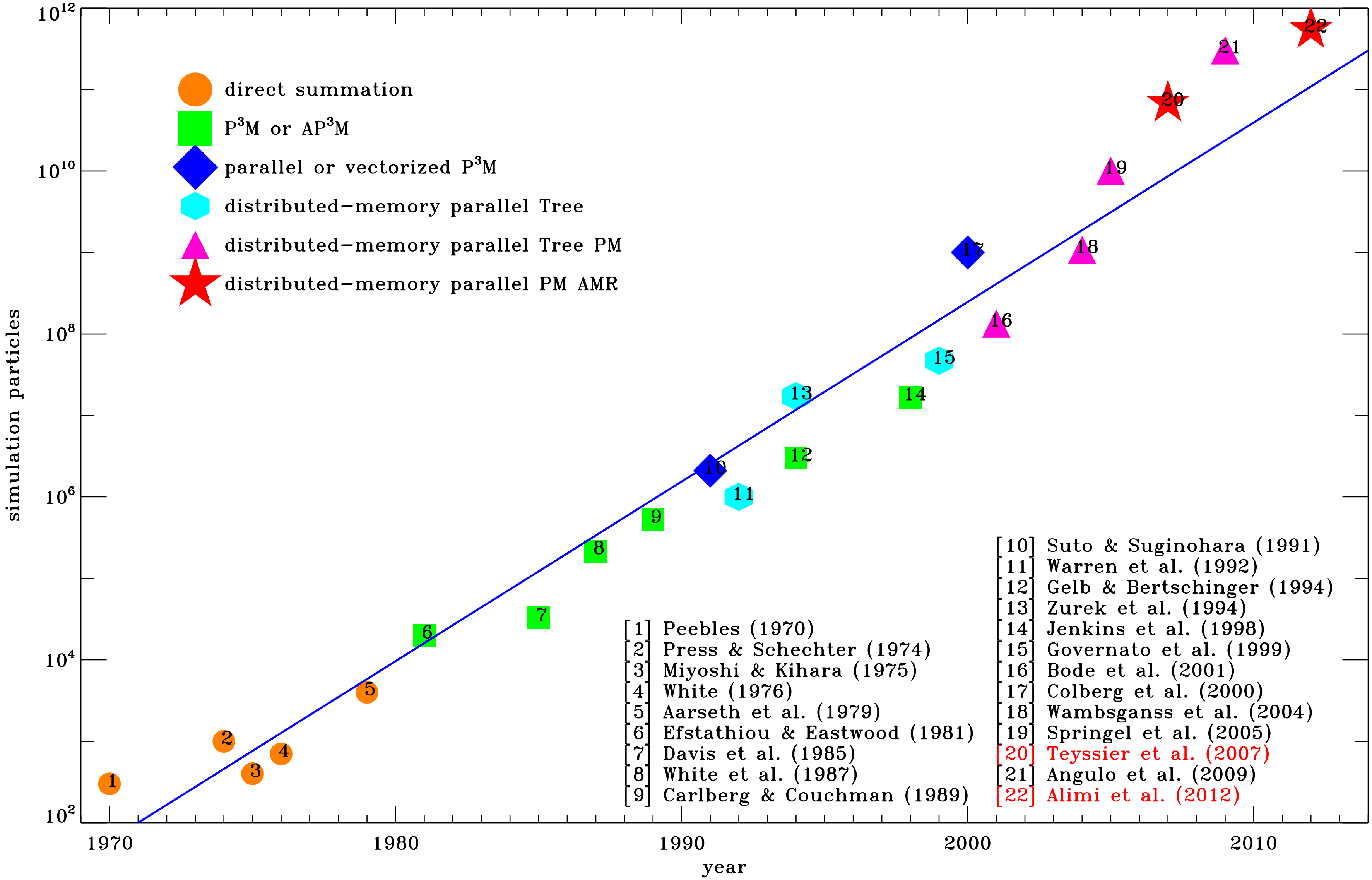
Baojiu Li *et al.* 2011

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.

The first N body simulation ever



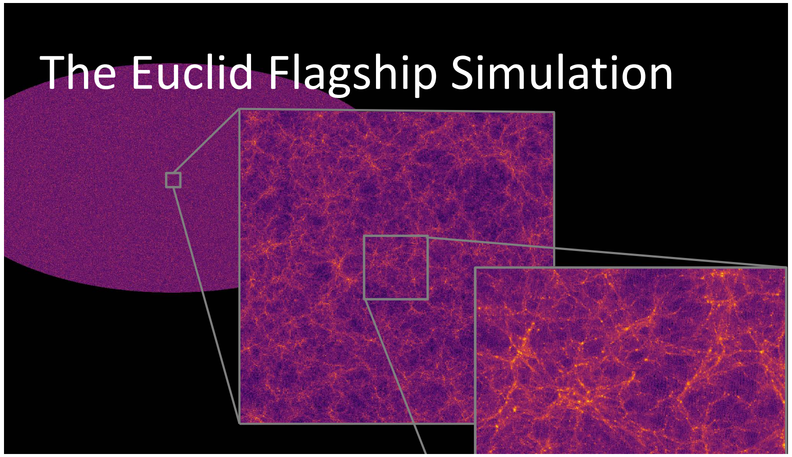
Cosmological N body simulations



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.

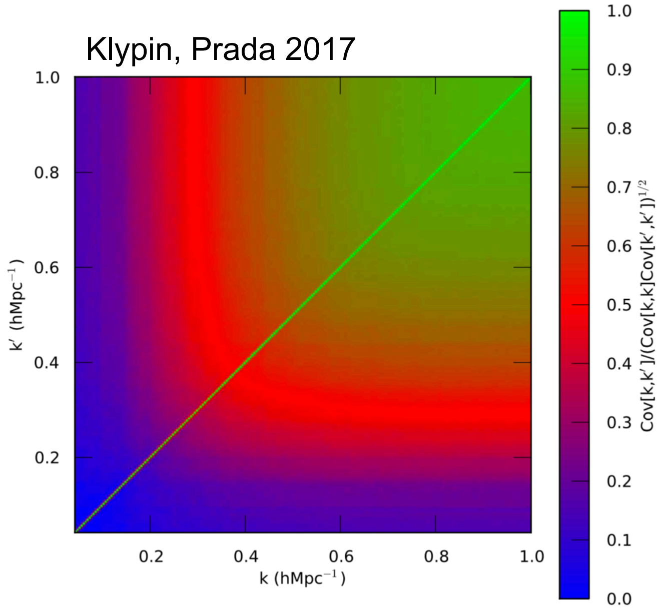
Resources: 2 million node-hours (with GPU)



Potter, Stadel, Teyssier 2017

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

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

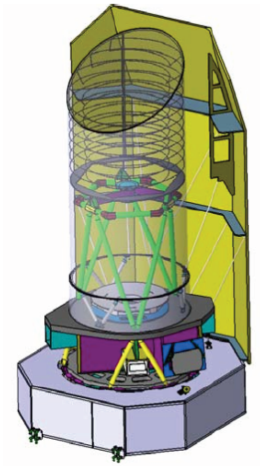


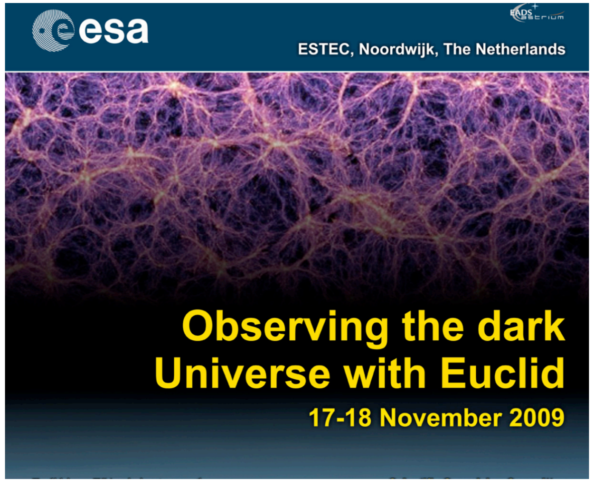
Covariance matrices: 3000 simulations with 8B particles every year.

Resources: 2 million node-hours (with GPU)

Very large N body simulations with RAMSES

Billion dollar experiments need support from HPC



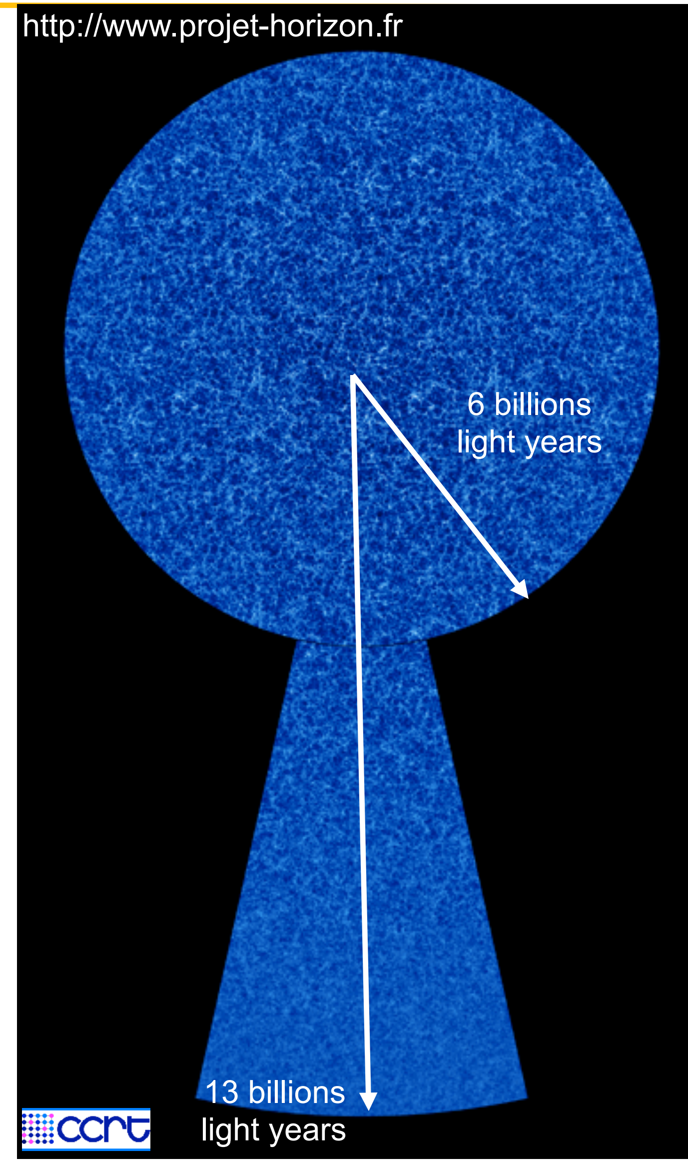


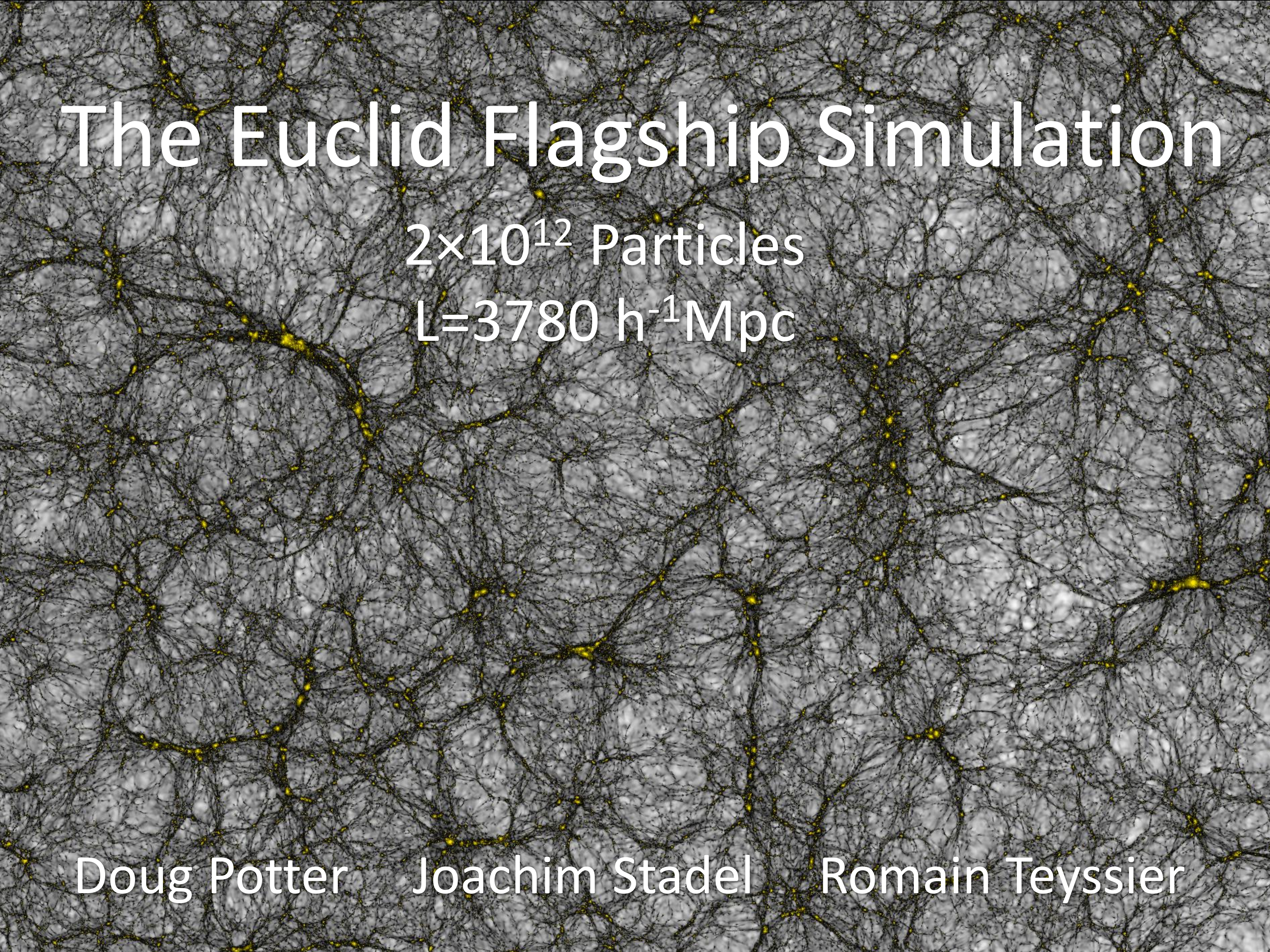
The Horizon simulation (2 Gpc/h)

70 billion dark matter particles and 140 billion AMR cells
6144 core
2.4 GB per core
Wall-clock time 2 month
performed in 2007 on the CCRT BULL Novascale 3045

The DEUS simulation (21 Gpc/h)

550 billion dark matter particles and 2 trillion AMR cells
76032 core
4 GB per core
Wall-clock time 1 week
performed in 2012 on the CCRT BULL Bullx S6010




The Euclid Flagship Simulation
2×10^12 Particles
L=3780 h^-1Mpc
Doug Potter Joachim Stadel Romain Teyssier

The pkdgrav3 N-Body Code

1. Fast Multipole Method, O(N), 5^{th} order in Φ
2. GPU Acceleration (Hybrid Computing)
3. Hierarchical Block Time-Stepping
4. Dual tree gravity calculation for very active particles
5. Very efficient memory usage per particle
6. On-the-fly analysis
7. Asynchronous direct I/O for checkpoints, the light cone data and halo catalogs.
8. Available on www.pkdgrav.org (bitbucket.org)

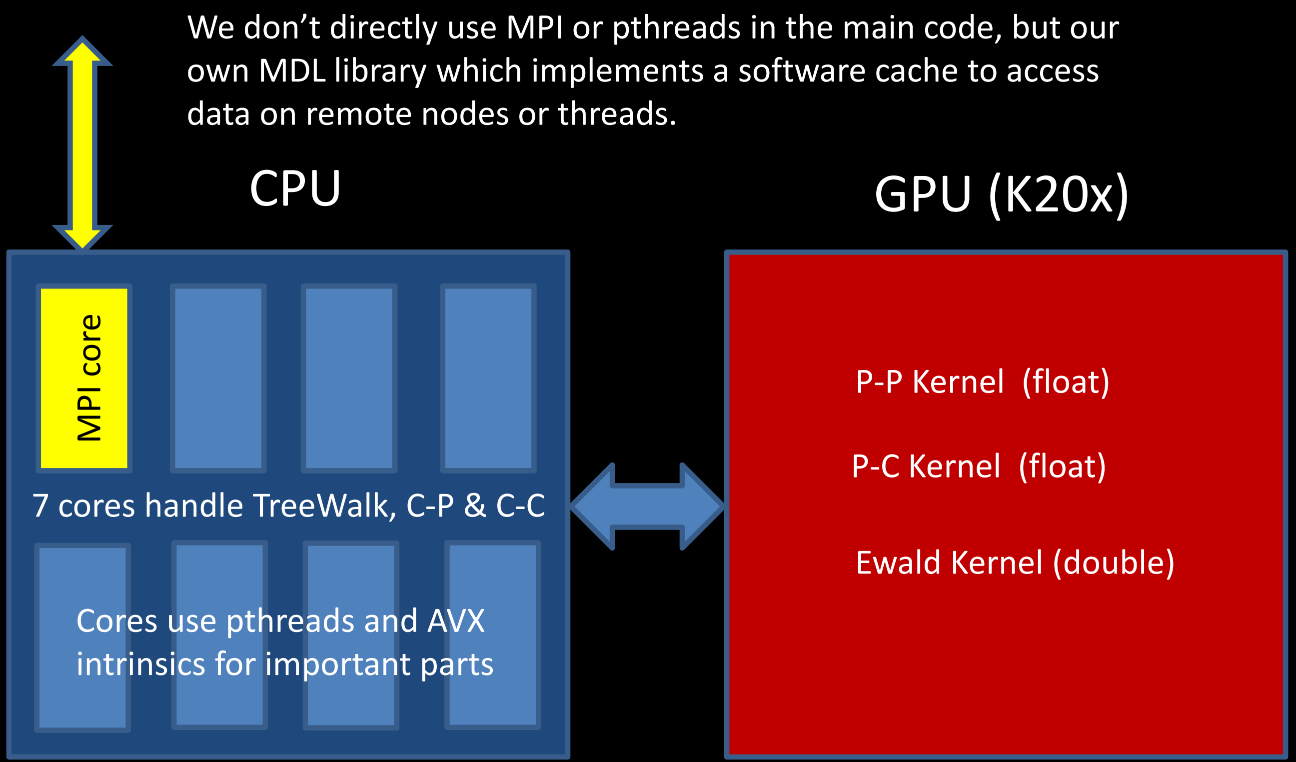
Piz Daint – over 5000 GPU Nodes

Swiss National Computing Center (CSCS) in Lugano, Switzerland

3rd Fastest Computer in the World Now!

GPU Hybrid Computing

Piz Daint example

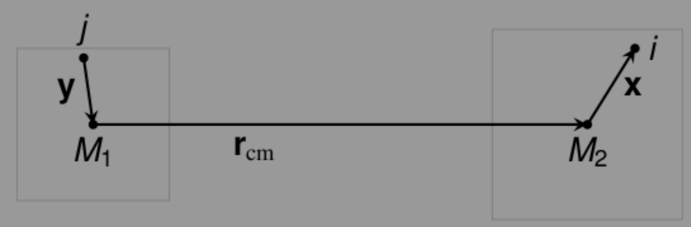


Fast Multipole Method - O(N)

Quick explanation of FMM

O(**10**^{6}) particles

O(**10**^{6}) particles



Direct **O**(**10**^{12}) interactions to calculate! $O(N^2)$ code.

Tree Use a multipole approximation for the mass at M_2 to calculate the force at each j: **O**(**10**^{6}) interactions to calculate. $O(N \log N)$ code.

FMM Use a multipole approx for the mass at M_2 to approximate the "potential landscape" at M_1 (n^{th} order gradients of the potential): **O**(**1**) interaction to calculate. $O(N)$ code!

Memory Usage in pkdgrav3

0.5 billion particles can fit on a 32 Gbyte Node like Piz Daint

28 bytes persistent

6 bits: old rung 24: group id	
pos[0]	int32_t
pos[1]	int32_t
pos[2]	int32_t
vel[0]	float
vel[1]	float
vel[2]	float

<28 bytes / particle

Tree Cells
Binary Tree

4th order
Multipoles
(float prec)

~5 bytes / particle

Cache/Buffers

0-8 bytes ephemeral

Group finding
Other analysis

ClAoS is used for the particle and cell memory which makes moving particles around simple

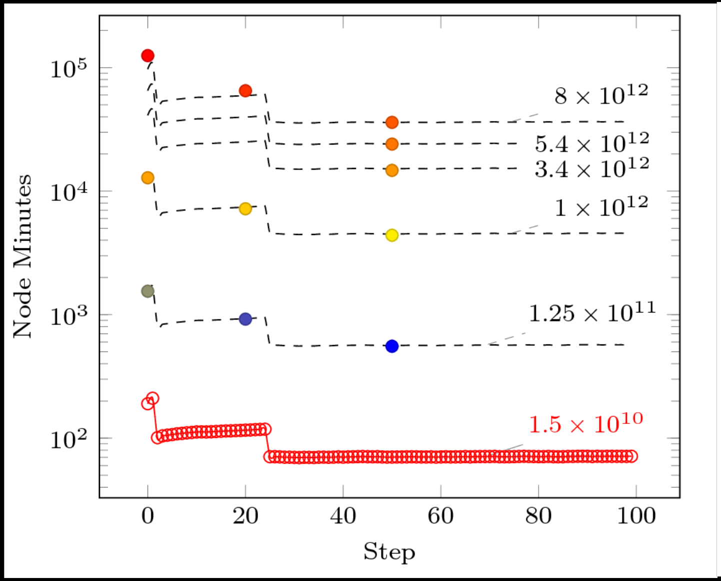
AoSoA is used for all interaction lists which are built by the TreeWalk algorithm.

Reducing memory usage increases the capability of existing machines, but also increases performance somewhat. Simulations are limited more by memory footprint.

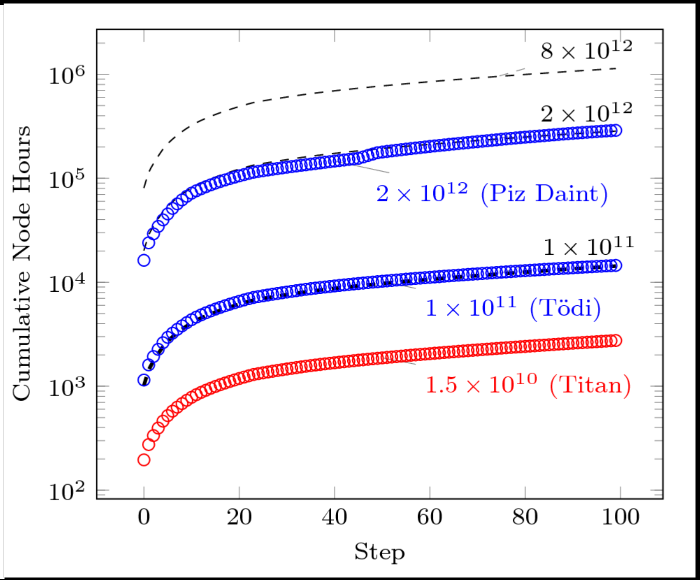
see Potter+ 2017 for details

Benchmarking on Titan and Piz Daint

Nearly Perfect Weak Scaling makes performance prediction very accurate for these simulations.
120 seconds for an all N gravity solve!

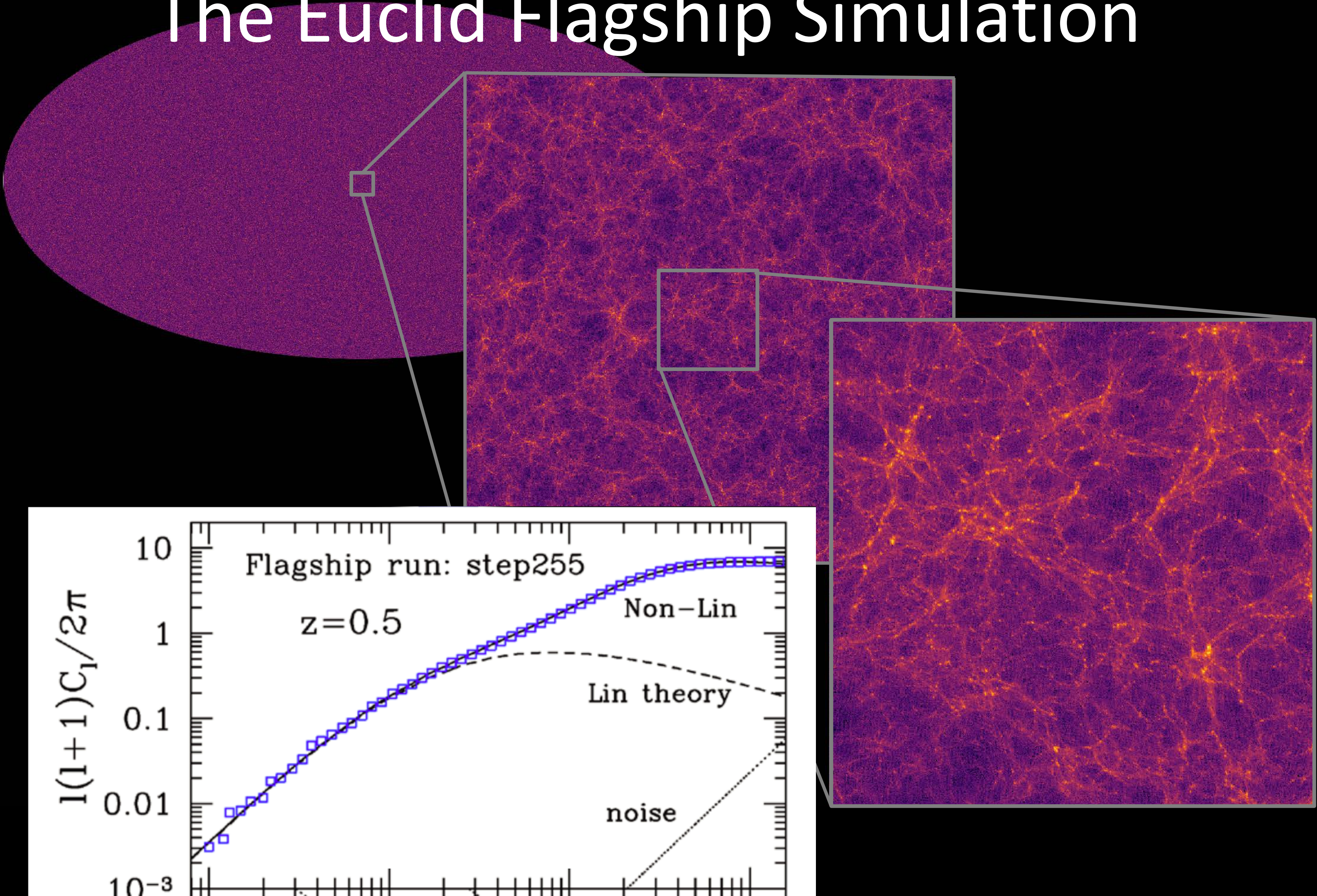


We show that it is quite feasible to run 8 trillion particles on Titan with a little over 1 million node hours. **10 PFlops**

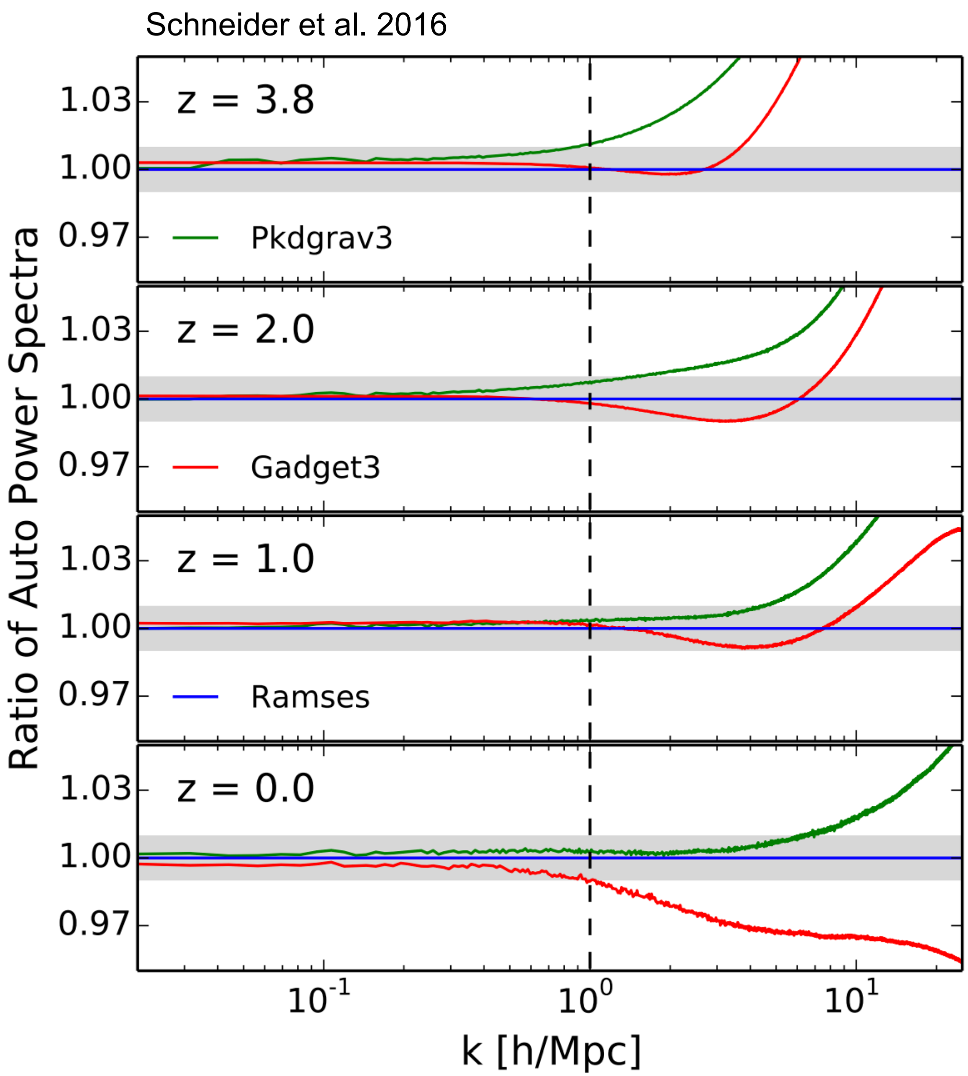


see Potter+ 2017 for details

The Euclid Flagship Simulation



Accuracy of large scale structure simulations

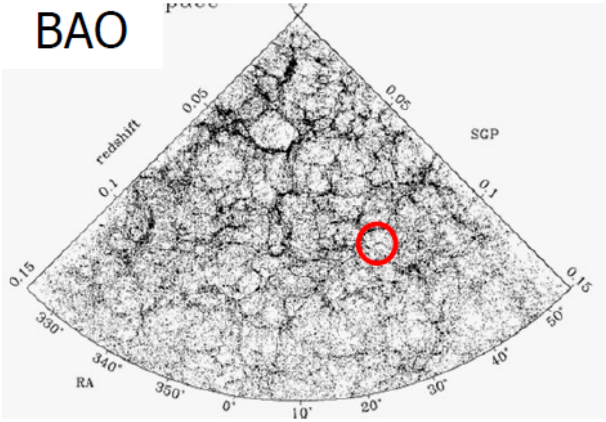


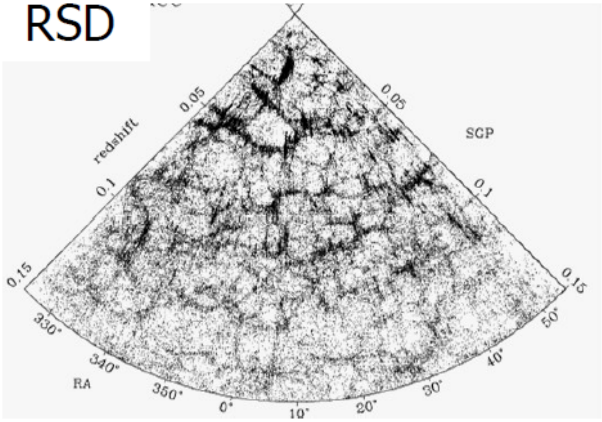
Euclid: a space mission to map the dark universe

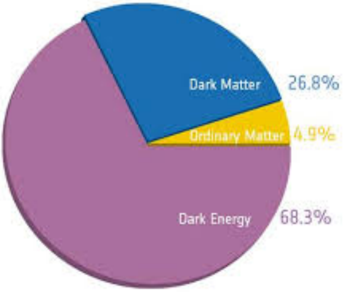
BAO, RSD and WL over 15,000 deg^2

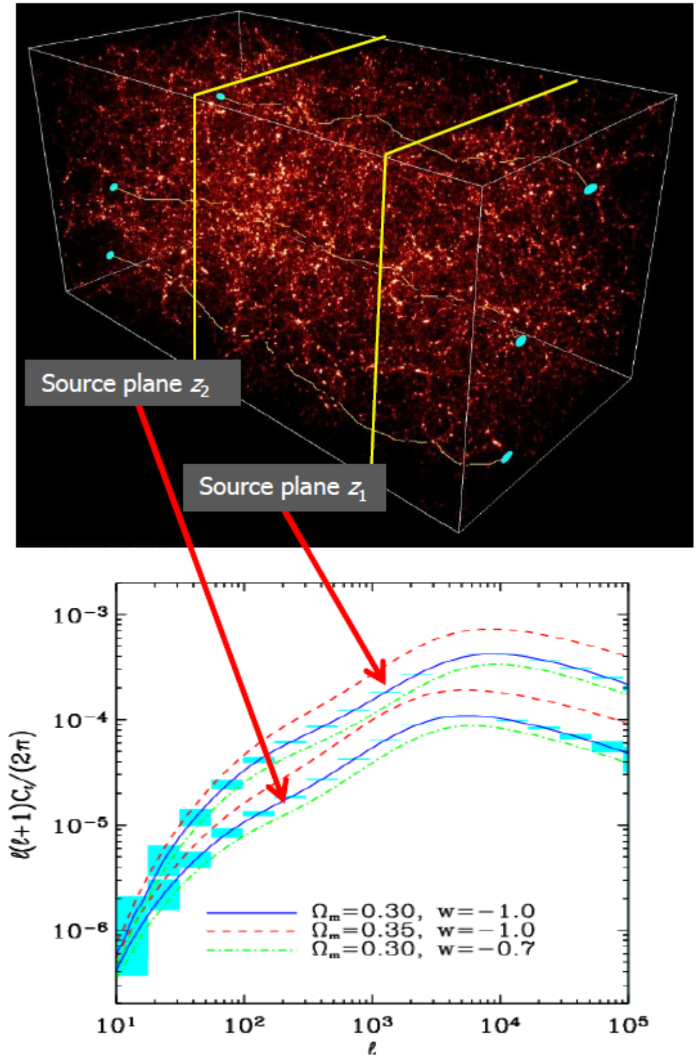
50 million galaxies with redshifts

1.5 billion sources with shapes, 10 slices









Thank you !