The RAMSES code and related techniques

3. Gravity solvers

Romain Teyssier

University of Zurich

i r f u

cea

s a c la y

Outline

- The Vlassov-Poisson equation
- The Particle-Mesh method in a nut shell
- Symplectic Integrators
- Mass assignment and force interpolation schemes
- FFT-based Poisson solver
- P3M, AP3M, TPM, and related high-resolution schemes
- AMR schemes
- Relaxation Poisson solvers
- Multigrid Poisson solvers and AMR specifics

The Vlassov-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

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

3 solution strategies:

- pure fluid on a 6D grid
- pure N body using direct or Tree force computations
- mixture of the 2: the Particle-Mesh method

The Particle-In-Cell method

N body integrator coupled to a grid-based Poisson solver

- Compute the mass density field on the grid from the particle distribution
- Solve for the Poisson equation on the grid
- Interpolate the force back to the particle position

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

Harlow, F.H., “A Machine Calculation Method for Hydrodynamics Problems”, Los Alamos Scientific Laboratory report LAMS-1956 (1955)

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

Symplectic Time Integrator

Phase space: $q = \mathbf{x}_p \quad p = \mathbf{v}_p \qquad \dot{q} = p$

Hamiltonian: $\mathcal{H}(q,p) = p^2/2 + \Phi(q) \qquad \dot{p} = -\dfrac{\partial \Phi}{\partial q}$

The exact solution of an Hamiltonian system is area-preserving in phase-space (incompressible fluid in phase space).

The area is a Poincare invariant.

$$\mathbf{z} = (q,p) \qquad \mathbf{f}(\mathbf{z}) = (p, -\partial\Phi/\partial q) \qquad \dot{\mathbf{z}} = \mathbf{f}(\mathbf{z})$$

Define the orbits as: $\mathbf{z}(t) = \mathcal{F}(\mathbf{z_0})$

The area in phase-space is preserved if: $\det \dfrac{\partial \mathcal{F}}{\partial \mathbf{z}} = 1$

Classical First Order Time Integrators

- Explicit Euler: $\mathbf{z}^{n+1} = \mathbf{z}^n + \Delta t \mathbf{f}(\mathbf{z}^n)$ $\det \frac{\partial \mathcal{F}}{\partial \mathbf{z}} = 1 + \Delta t^2 \frac{\partial^2 \Phi}{\partial q^2}$

- Implicit Euler: $\mathbf{z}^{n+1} = \mathbf{z}^n + \Delta t \mathbf{f}(\mathbf{z}^{n+1})$ $\det \frac{\partial \mathcal{F}}{\partial \mathbf{z}} = \frac{1}{1 + \Delta t^2 \frac{\partial^2 \Phi}{\partial q^2}}$

- Symplectic Euler: $\mathbf{z}^{n+1} = \mathbf{z}^n + \Delta t \mathbf{f}(q^n, p^{n+1})$ $\det \frac{\partial \mathcal{F}}{\partial \mathbf{z}} = 1$

Second Order Symplectic Time Integrator

Kick-Drift-Kick algorithm (see also Leap Frog)

$$p^{n+1/2} = p^n - 0.5\Delta t \left(\frac{\partial \Phi}{\partial q}\right)^n$$

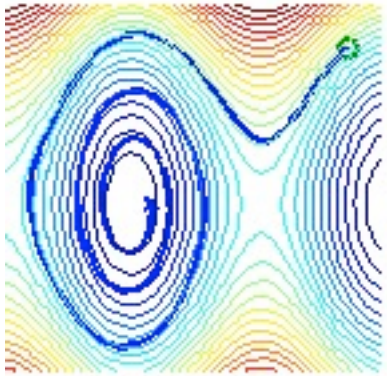
$$q^{n+1} = q^n + \Delta t p^{n+1/2}$$

$$p^{n+1} = p^{n+1/2} - 0.5\Delta t \left(\frac{\partial \Phi}{\partial q}\right)^{n+1}$$

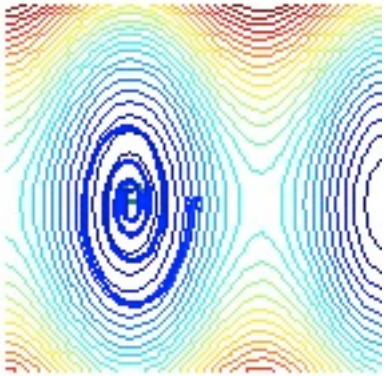
Exercise: check that the scheme is area-preserving.

Why use a symplectic integrator ?

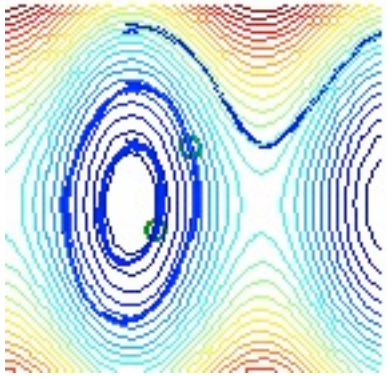
Explicit Euler



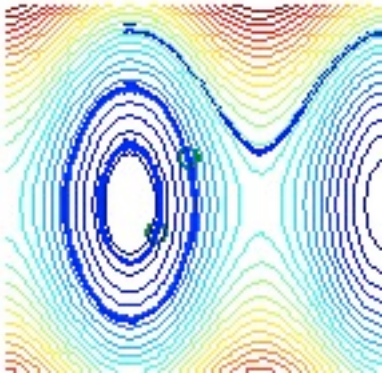
Implicit Euler



Symplectic Euler



Implicit Midpoint



Charge 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): $$S(x) = \frac{1}{\Delta x}\Delta\left(\frac{x}{\Delta x}\right)$$

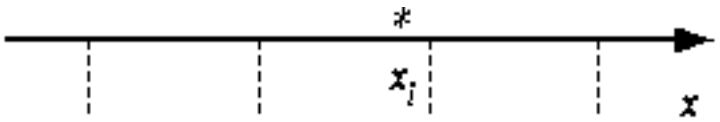
The contribution of each particle to the charge in the cell is:

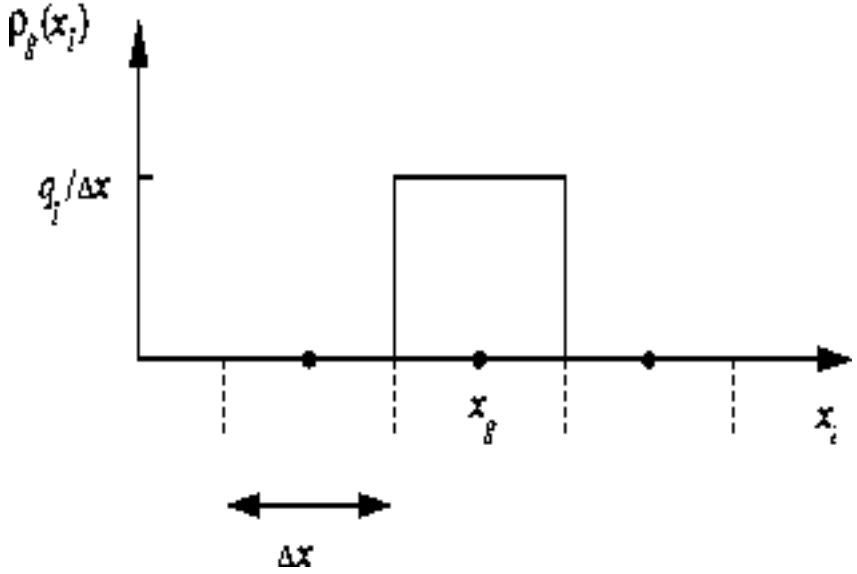
$$W^{\rho}(x_p - x_i) = \int_{x_i-\Delta x/2}^{x_i+\Delta x/2} S(x_p - x)dx$$

The total charge in the cell is: $$\rho_i = \frac{1}{\Delta x}\sum_{p=1}^{N_p} m_p W^{\rho}(x_p - x_i)$$

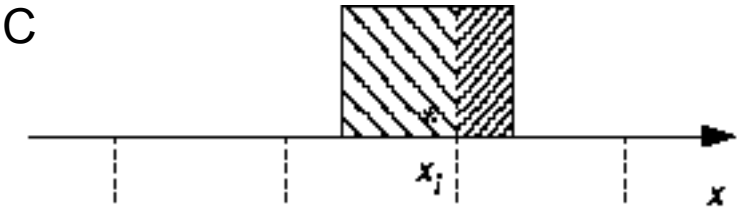
A Hierarchy of Shapes

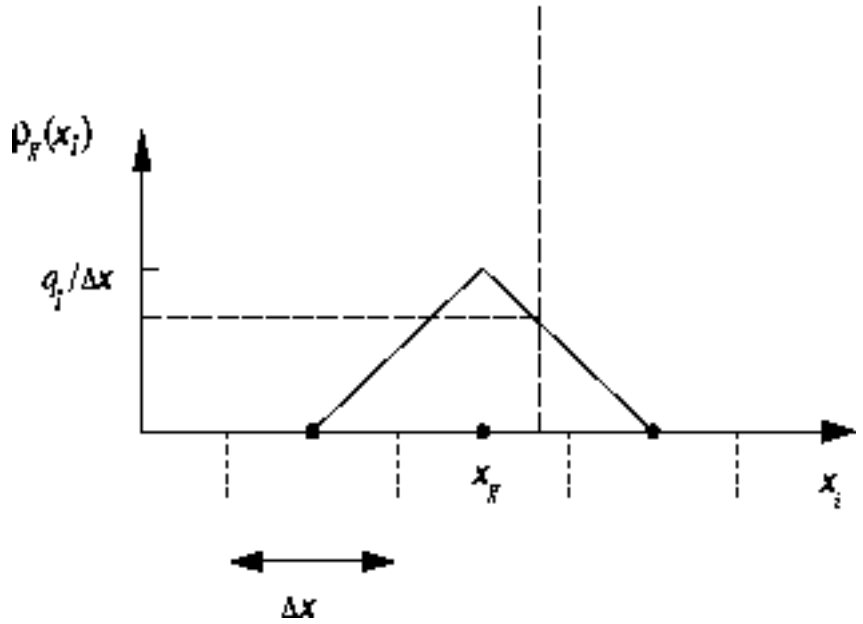
NGP





CIC





From NGP to CIC,

From CIC to TSC,

etc...

Force interpolation

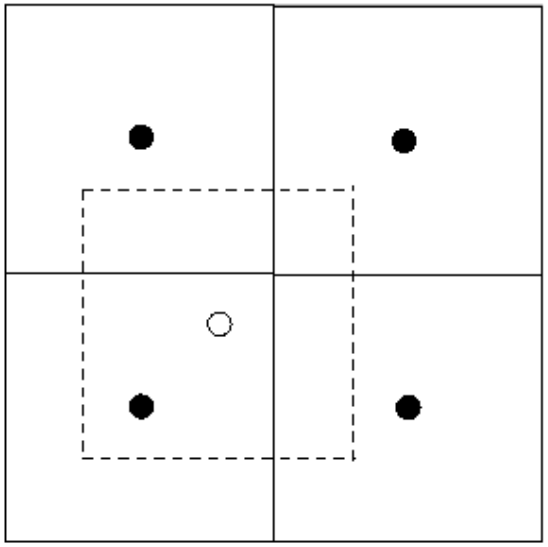
Use another interpolation scheme to get the mesh force at particle positions.

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

Momentum conservation is enforced if:

- 2 interacting particles see equal but opposite forces (exercise !)

- 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:

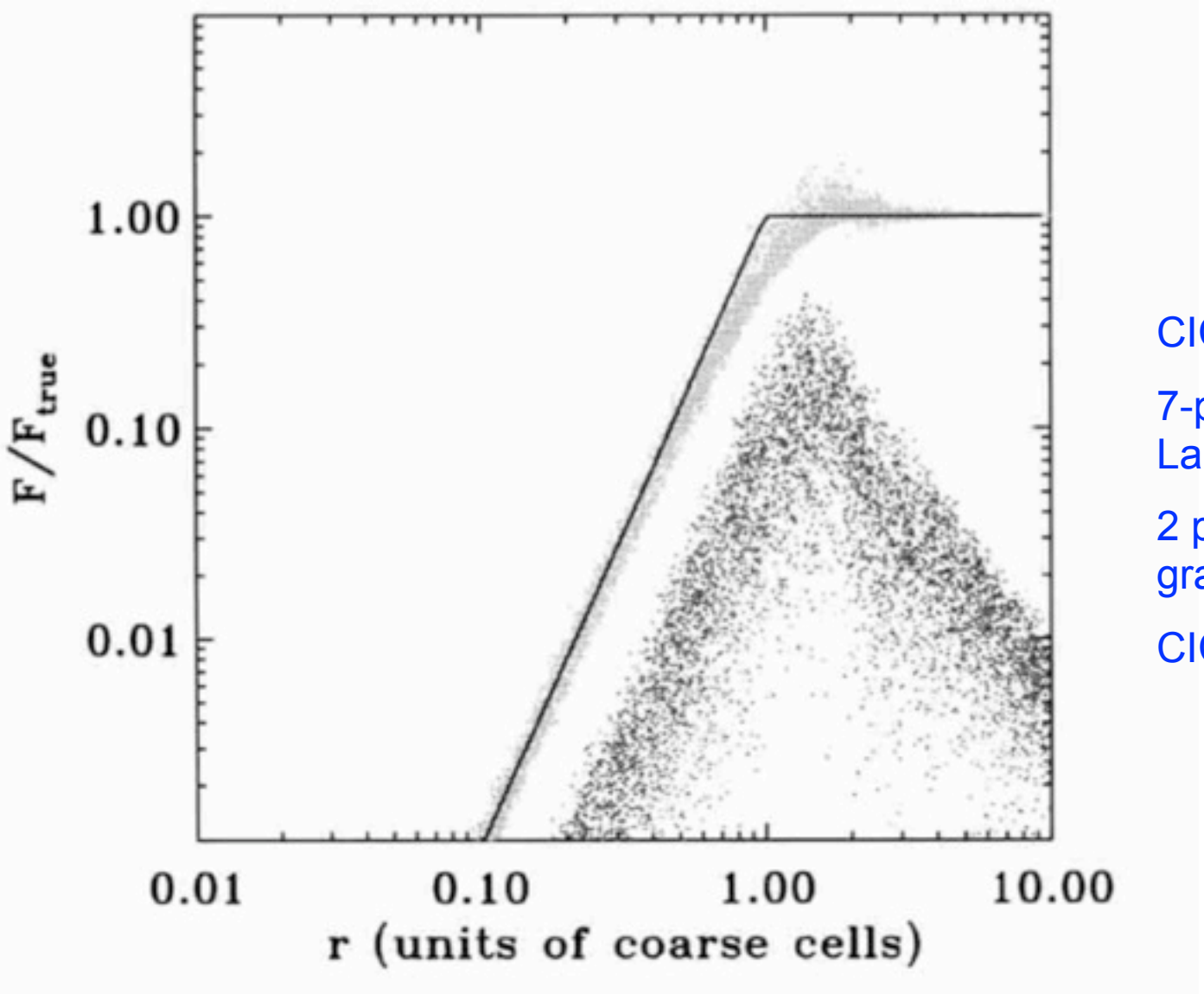
$$\Phi_{i+1} - 2\Phi_i + \Phi_{i-1} = \rho_i \Delta_x^2 \qquad \tilde{G}(k) = -\frac{\Delta x^2/4}{\sin(\frac{k\Delta x}{2})^2}$$

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

Final force is given by:

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

PM force accuracy



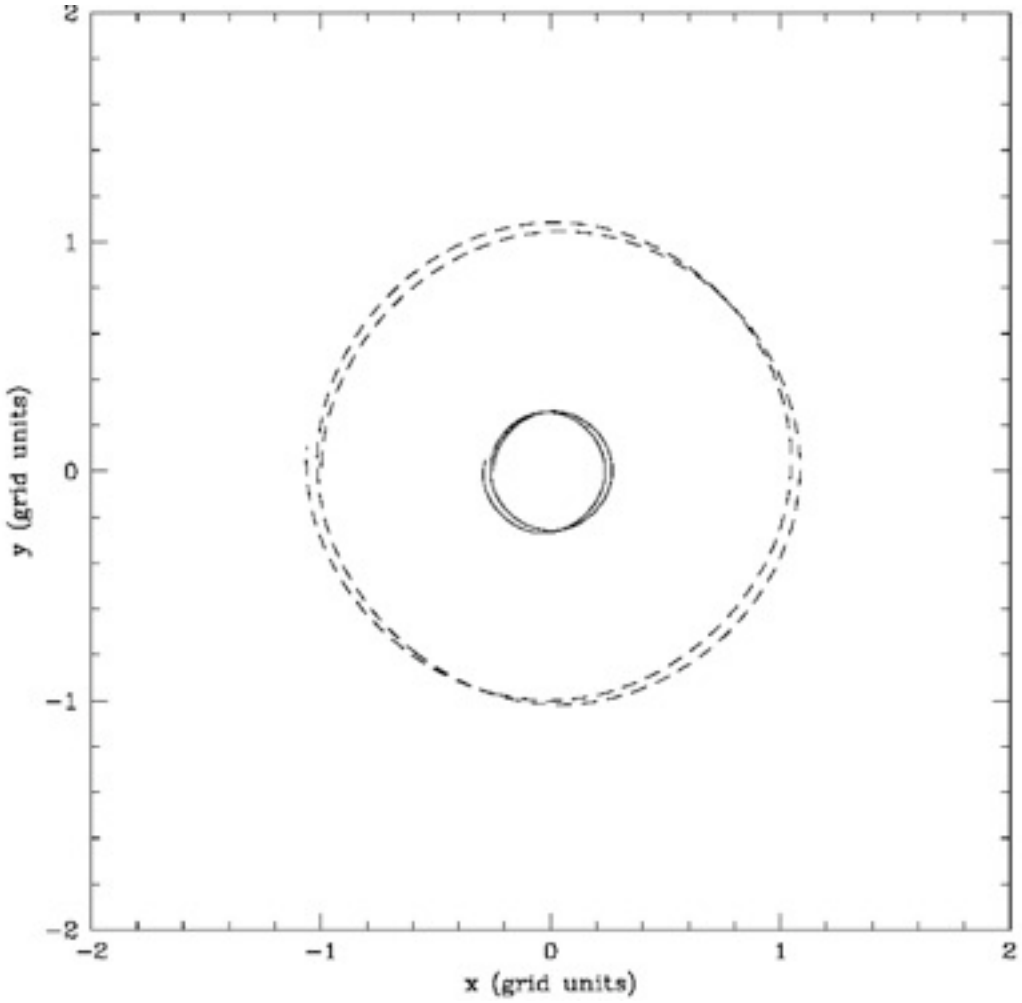
CIC

7-point Laplacian

2 points gradient

CIC^{-1}

Force + integrator accuracy



examples of particle's trajectory

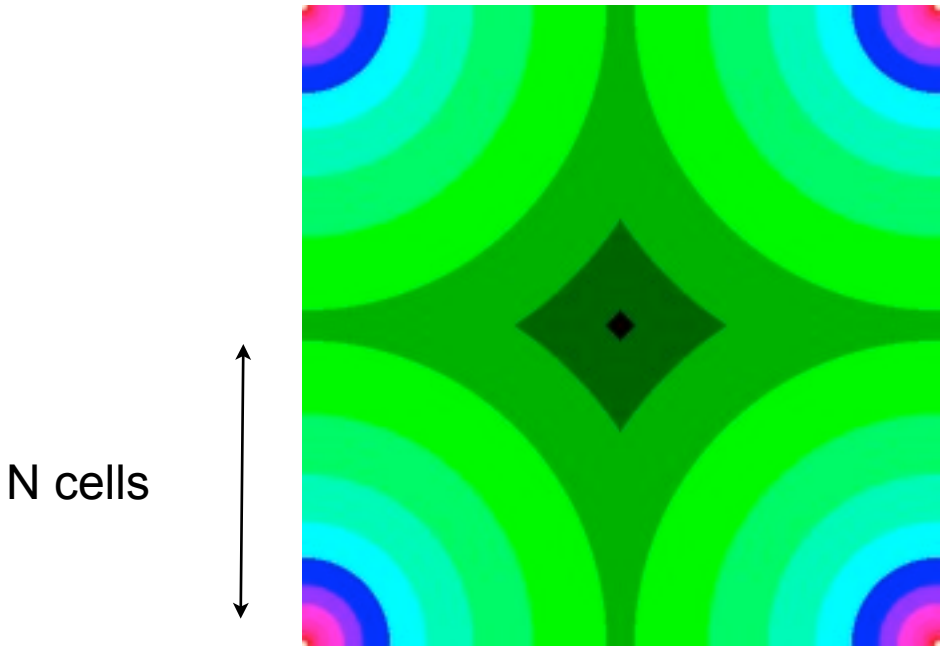
What about isolated boundary conditions ?

Fourier analysis (and FFT) is more straightforwardly performed in a periodic system. How can one still use Fourier analysis in an isolated system ?

The “zero padding” trick (Hockney & Eatwood 81) for a NxN grid.

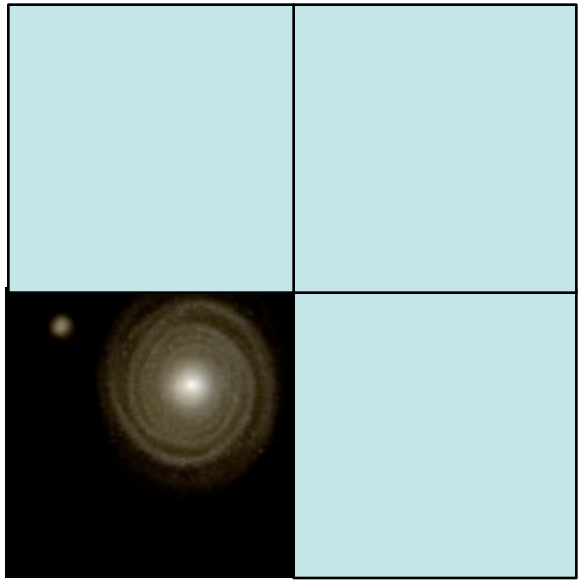
1- Compute the Green’s function in real space, in the NxN grid.

2- Duplicate and mirror the Green’s function in the 3 “ghost” quadrants, so that the Green’s function is now periodic in the 2Nx2N grid.



What about isolated boundary conditions ?

3- Set the density field to zero in the 3 ghost quadrants.



4- Perform the convolution using Fourier analysis on the 2Nx2N grid.

Very simple and flexible ! It is quite expensive (8 times longer) !

Alternative method:

James, R.A., “The solution of Poisson’s equation for isolated source distribution ”, JCP, **25**, 71, (1977)

Towards high-resolution: the P3M scheme

Decompose the total force in 2 components:

$$F_{hires}(r) = F_{PM}(r,a) + F_{PP}(r,\epsilon)$$

The PP force is a short-range interaction for $r < r_c$=2-3a:

$$F_{PP}(r,\epsilon) = \frac{1}{r^2+\epsilon^2} - F_{ref}(r,a)$$

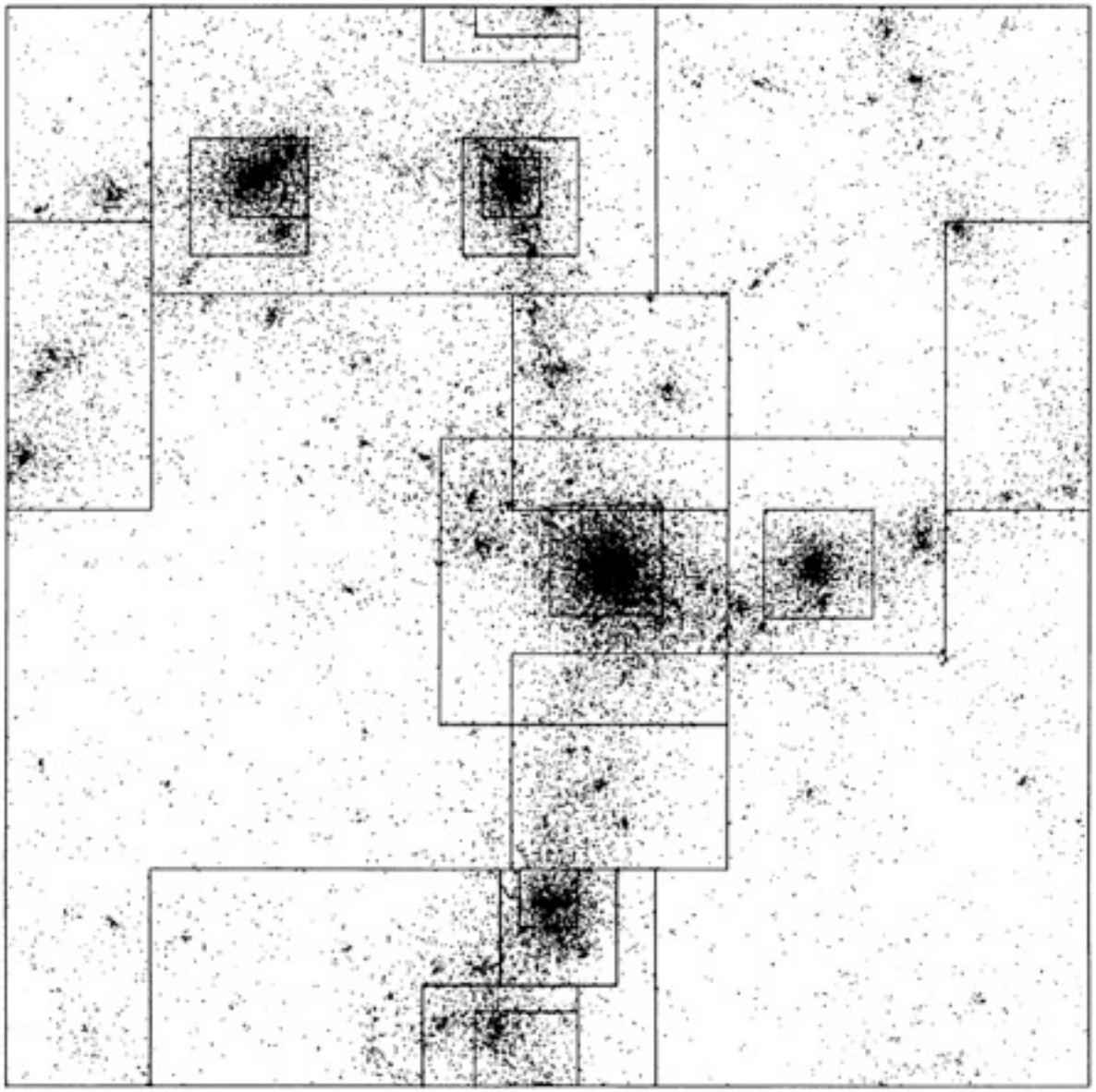
The PP force is computed with direct N^2 summation, for all particles within the cut-off radius.

Problem: for highly clustered configuration, the PP force dominates the CPU time in the P3M scheme.

Solution: adaptive grids: the AP3M scheme.

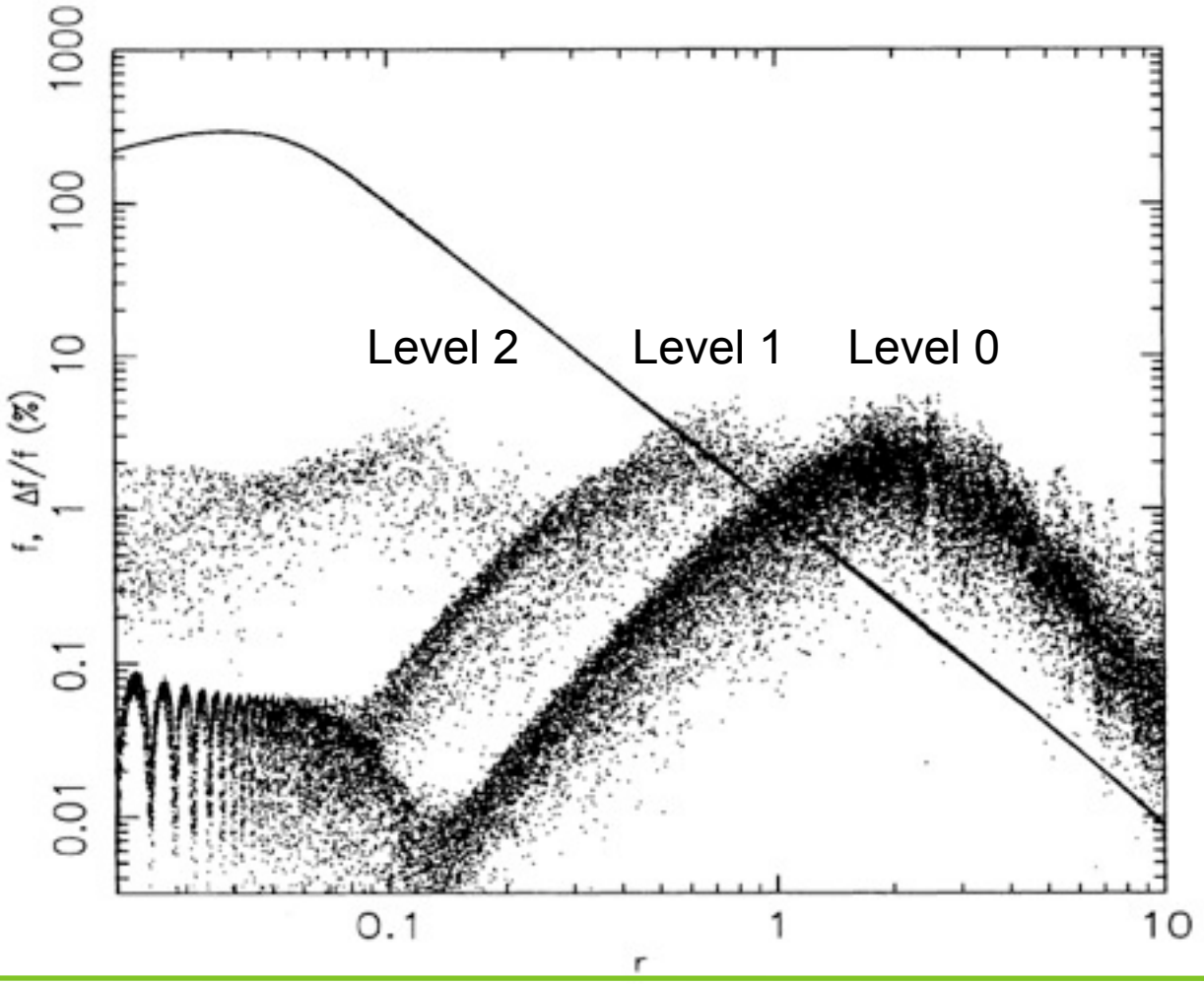
Couchman, H.M.P., “Mesh refined P3M: A fast adaptive N-body algorithm”, ApJ, **368**, 23, (1991)

Adaptive grids: the AP3M scheme



Adaptive grids: the AP3M scheme

Decompose the force recursively in smaller and smaller scale components. Only the coarse grid PM force is long-range. Requires force shaping at each level, and the use of an isolated boundary conditions Poisson solver.



High-resolution with constant force softening

Class of high-resolution N-body solvers based on the Particle In Cell method, with a constant (small scale) force softening:

- P3M: Hockney & Eastwood 81

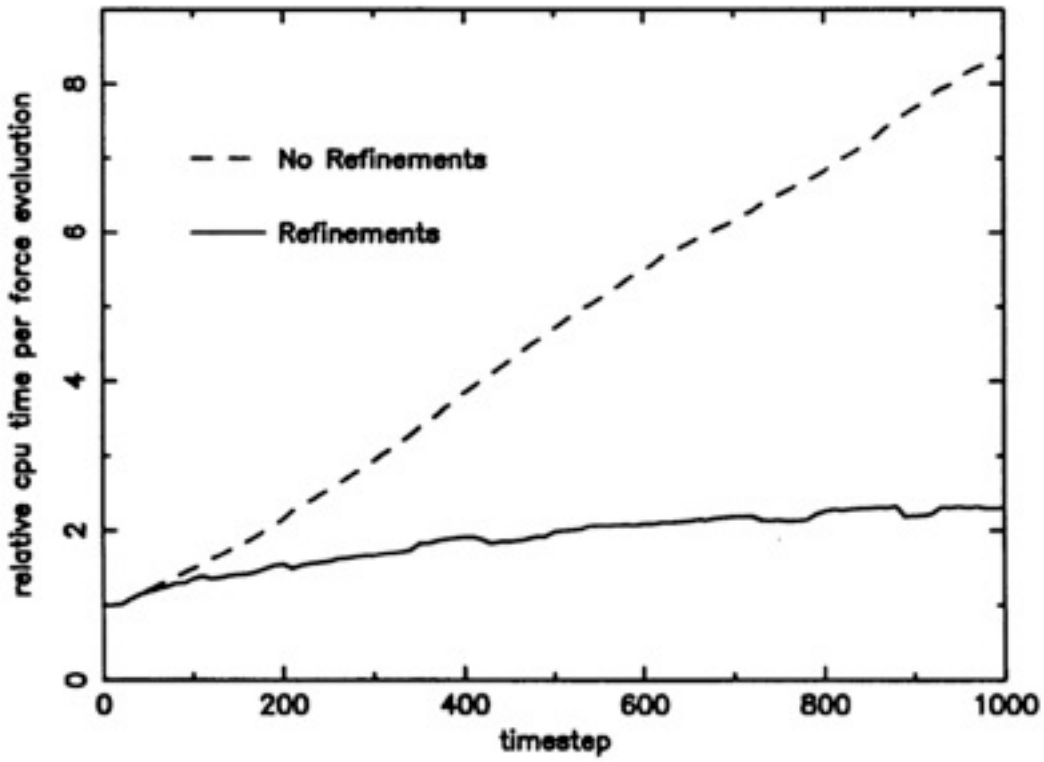
- AP3M: Couchman 91

- TreePM: Bode & Ostriker 2003

Speed-up of the P3M scheme

using adaptive grids

(from Couchman et al. 95)



High-resolution with constant force softening

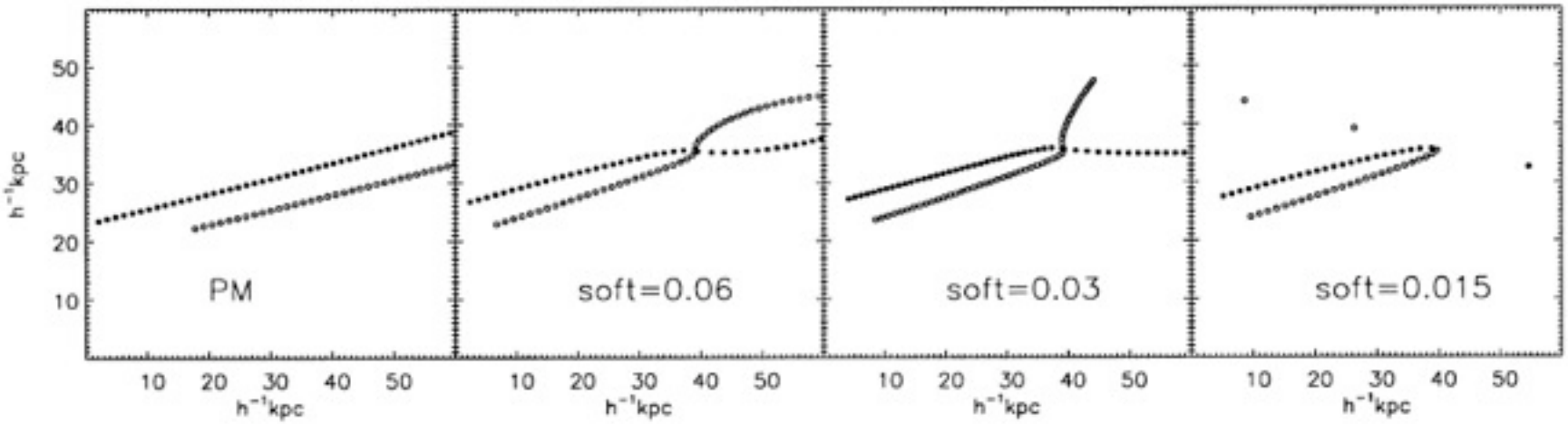
Problem: how do we choose the force softening length ?

Usually, one has $\epsilon = \frac{1}{10}\Delta x$ down to $\frac{1}{100}\Delta x$

where Δx is the mean inter-particular distance.

Pros: constant softening length ensures that the integration is symplectic, which results in good energy conservation

Cons: smoothing length is usually too small in low density regions and too large in high density regions.



Examples of 2-body scattering in low density region (from Knebe et al. 2000)

High-resolution with constant force softening

Particle discreteness effects show up quite dramatically in Warm Dark Matter simulations (from Wang & White, MNRAS, 2007)

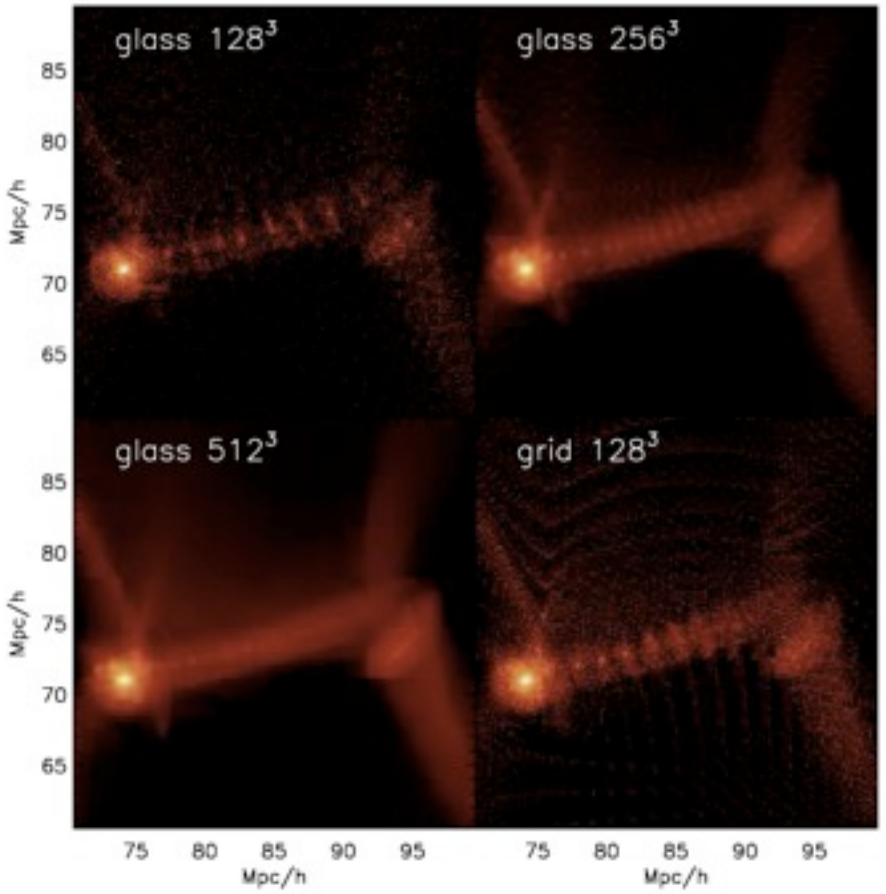
Very slow convergence N^(-1/3)

These effects can be neglected if

$$\epsilon \simeq 2d$$

d: *local* inter-particular spacing

Adaptive force softening ?



Splinter, R.J., Melott, A.L., Shandarin, S.F., Suto, Y., " Fundamental Discreteness Limitations of Cosmological N-Body Clustering Simulations", ApJ, **497**, 38, (1998)

Romeo, A.B., Agertz, O., Moore, B., Stadel, J., "Discreteness Effects in ΛCDM Simulations: A Wavelet-Statistical View", ApJ, **686**, 1, (2008)

PM with Adaptive Mesh Refinement

Idea: use the PM algorithm on a set of adaptively refined grid. Similar to the AP3M algorithm, without the short range PP part.

Method: each cell is recursively refined if the number of particle per cell exceed some threshold (around 10 particles).

2 different methods for the Poisson solver:

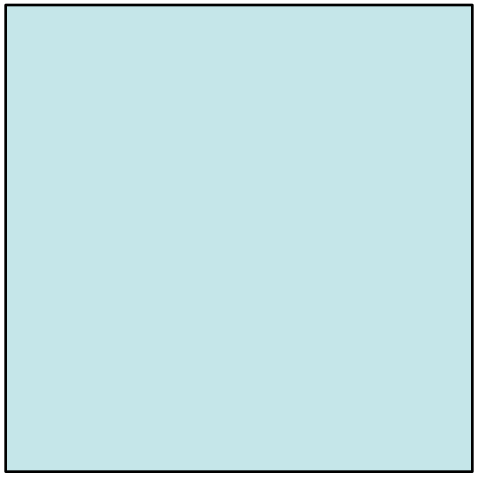
- **the PANDORA code**: Villumsen, J.W., “A New Hierarchical PM Code for Very Large Scale Cosmological N-body Simulations”, ApJS, **71**, 407, (1989)

- **the ART code**: Kravtsov, A.V., Klypin, A.A., Khokhlov, A.M., “ART: a new high-resolution N-body code for cosmological simulations”, ApJS, **111**, 73, (1997)

A lot of different codes: ENZO (AP3M), ART, RAMSES (ART), PANDORA, MLAPM (ART), FLASH (unclear), CHARM (ART)...

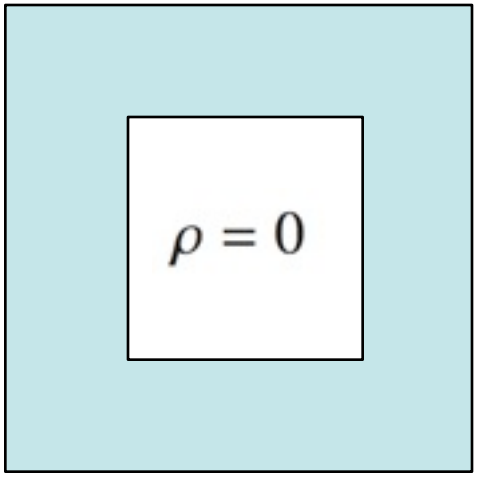
Poisson solver with AMR: the PANDORA scheme

Isolated BC



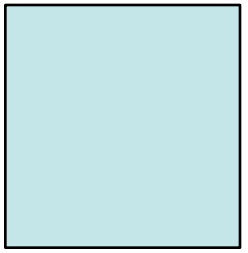
Step 1: compute the force on the coarse level due to *all* particles and interpolate to coarse-level particles

Isolated BC



Step 2: compute the force on the coarse level due to coarse-level particles and interpolate to fine-level particles

Isolated BC

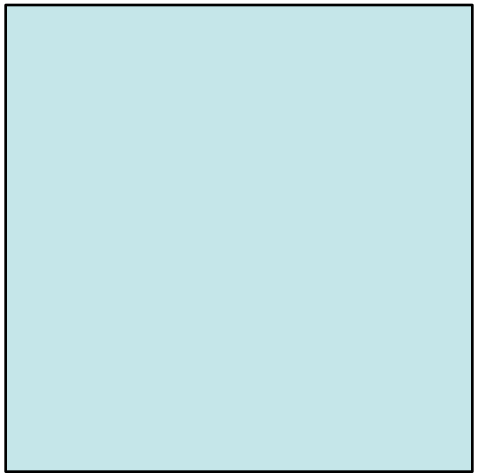


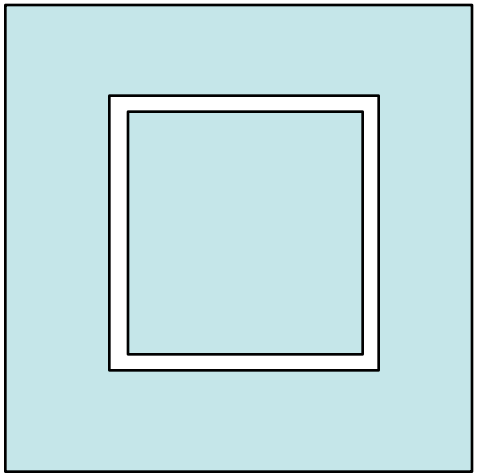
Step 3: compute the force on the fine level due to fine-level particles and interpolate to fine-level particles

Only for rectangular grids !

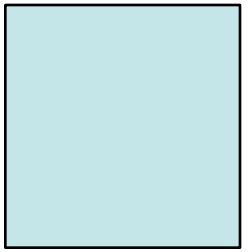
Poisson solver with AMR: the ART scheme

Periodic BC





Dirichlet BC



Step 1: compute the force on the coarse level due to *all* particles and interpolate to coarse-level particles

Step 2: interpolate coarse-level potential at the coarse-fine boundary

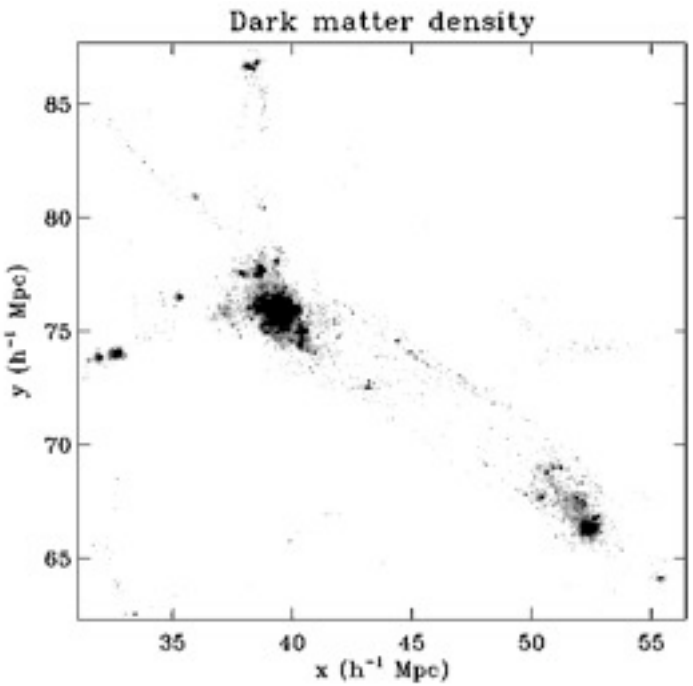
Step 3: compute the force on the fine level due to fine-level particles and interpolate to fine-level particles

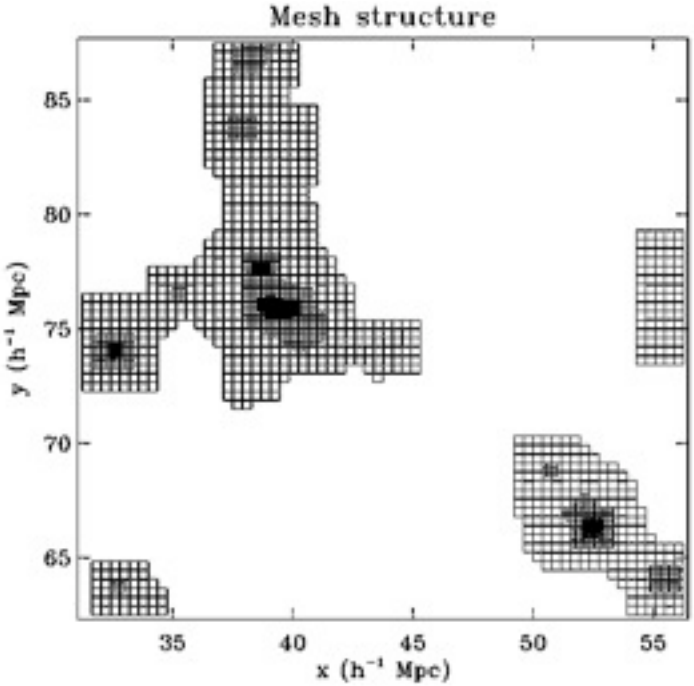
Valid for any grid geometry !

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

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

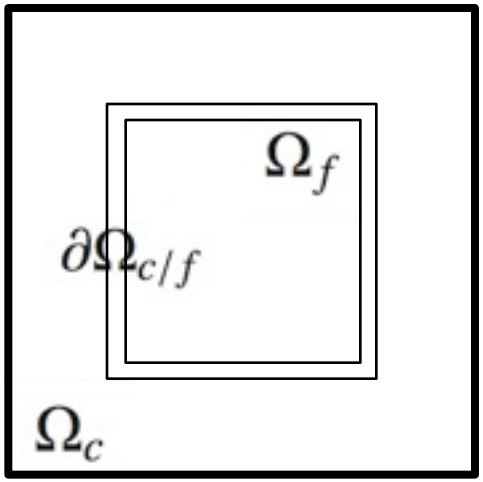
Fine-to-coarse information is mediated by particles only.

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.



A variant proposed by Miniati & Colella, JCP, 2007: in each coarse-level cell, the center-of-mass is computed and its contribution to ρ_c is added with CIC interpolation. Useful for self-gravity and gas dynamics.

A two-way interface Poisson solver for AMR

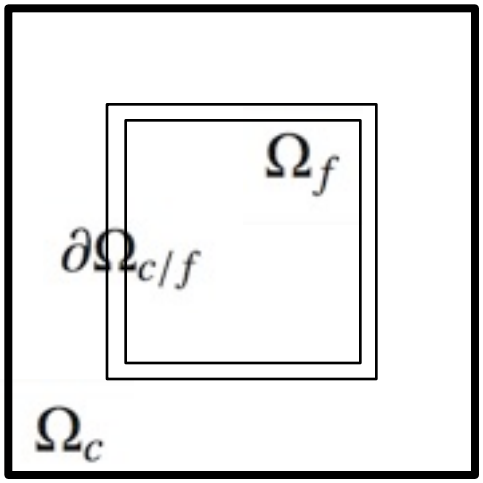
Proposed by Colella and co-workers for incompressible fluid dynamics and gravity solvers for compressible fluids.

Solve $\Delta\Phi_c = \rho_c$ on Ω_c - Ω_f

Impose Dirichlet and Neuman potential matching at $\partial\Omega_{c/f}$

Solve $\Delta\Phi_f = \rho_f$ on Ω_f

Iterate until global convergence



Applied to the Particle Mesh scheme: Miniati, F., Colella, P., " Block structured adaptive mesh and time refinement for hybrid, hyperbolic + N-body systems", JCP, **227**, 400, (2007)

In case of adaptive time steps, revert to one-way coupling.

Relaxation solvers for the Poisson equation

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

Simplest scheme: the Jacobi or Gauss-Seidel 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.

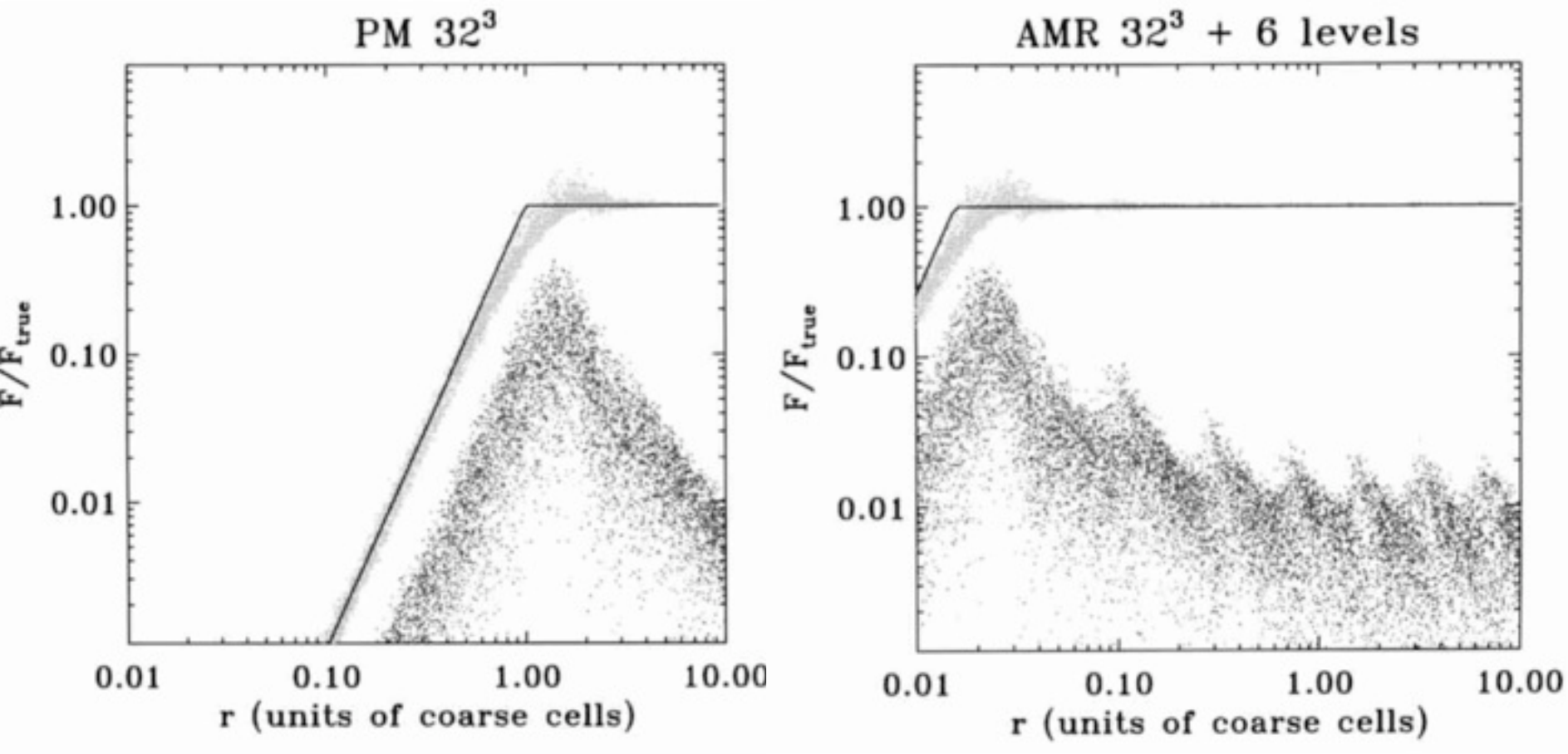
$$\phi_{i,j}^{n+1} = \omega\phi_{i,j}^{n} + (1-\omega)\phi_{i,j}^{n+1} \quad \text{with} \quad 1 < \omega < 2$$

Fastest convergence for $\omega \simeq \dfrac{2}{1+\alpha\frac{\pi}{N}}$

Similar performance with the Conjugate Gradient method. For a NxN grid: exact convergence in N^2 iterations,

In practice, order N iterations are necessary to reach the level of truncation errors `epsilon=1e-3 to 1e-4`

PM-AMR force accuracy



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 lnN or even N.

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

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 J/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:

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

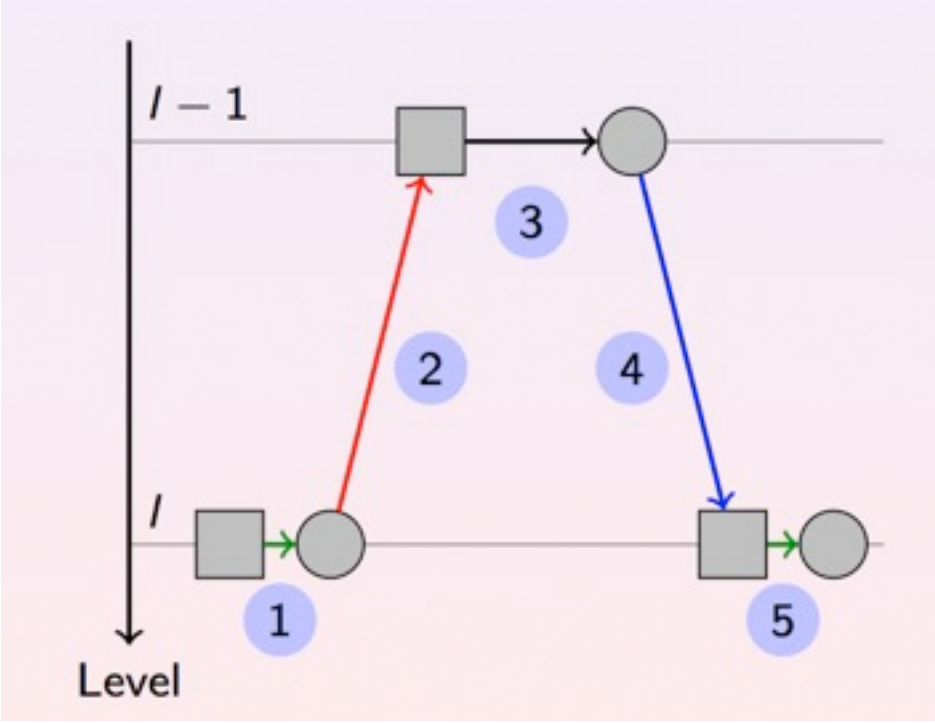
4- Prolong back the error to the fine grid:

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

5- Correct the fine grid solution:

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

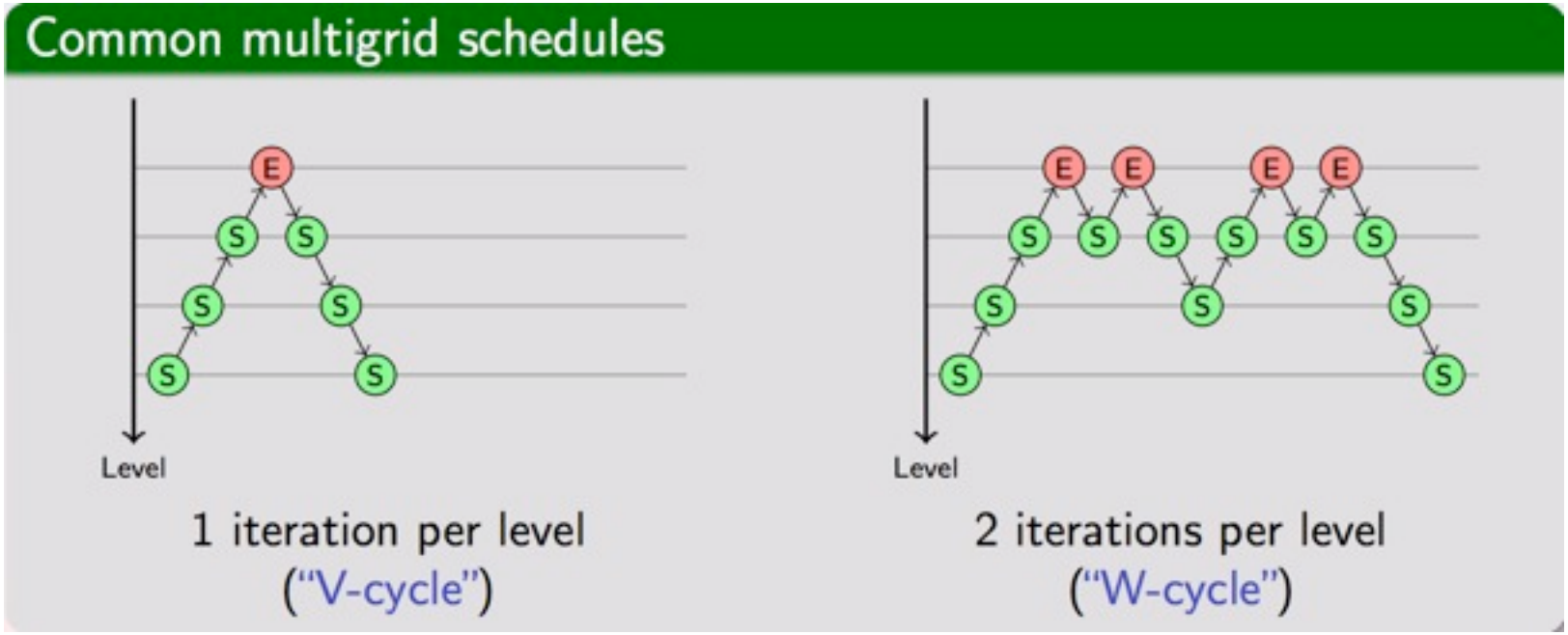
and perform a few J/GS iteration.



Multigrid scheme

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

Iterate one or twice before going to the finer level.



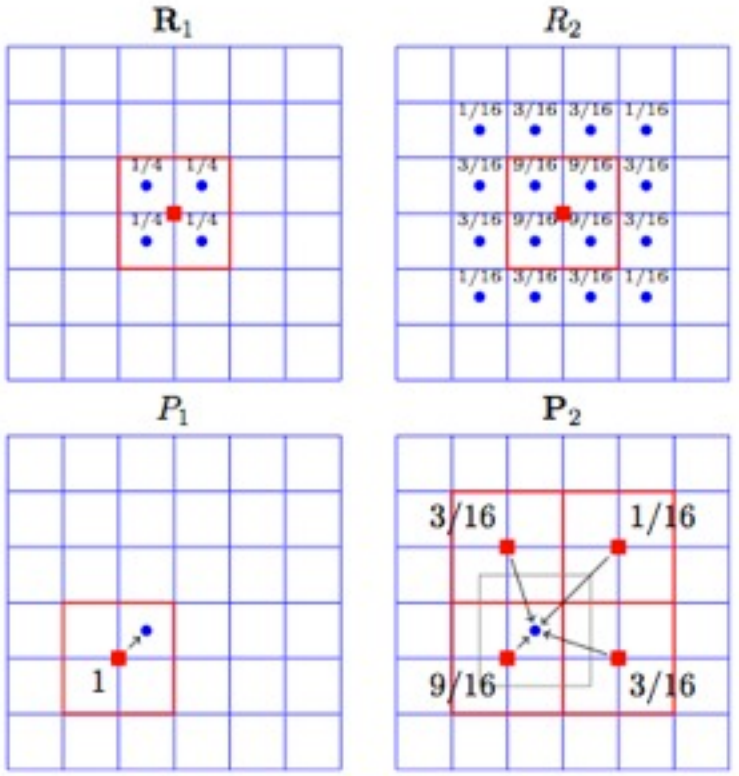
Converge in very few iterations, independently of grid size.

Quasi-insensitive to the quality of the initial guess.

Restriction and prolongation operators

Restriction: smooth or average the solution. Propagate the information from small to large scales. Use Cloud-In-Cell on the coarse grid.

Prolongation: interpolate the solution. Warning: this creates new information. Use Cloud-In-Cell from the coarse grid.



Rules:

- order(R)+order(P) > order(Δ)

- $R^T P = I$

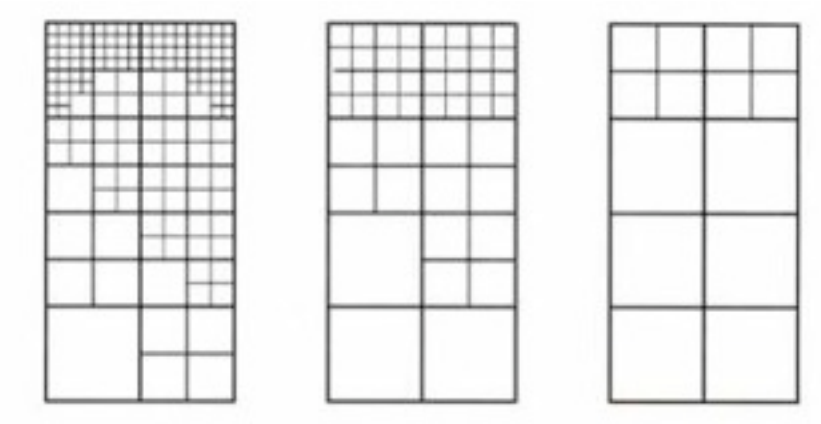
- $\Delta_{\ell-1} = R \Delta_\ell P$

In RAMSES, we use R_1 and P_2

Multigrid for AMR

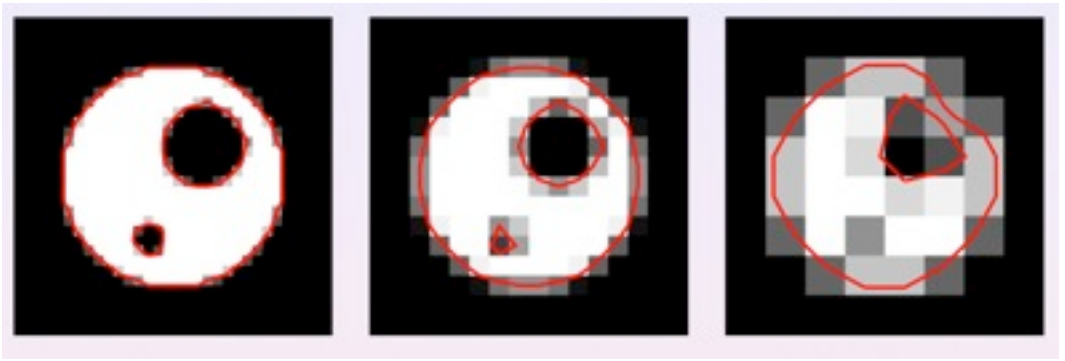
Two-way coupling: one needs to define a ensemble of AMR grids, each AMR grid corresponds to a level in the multigrid hierarchy.

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



Level 3 Level 2 Level 1

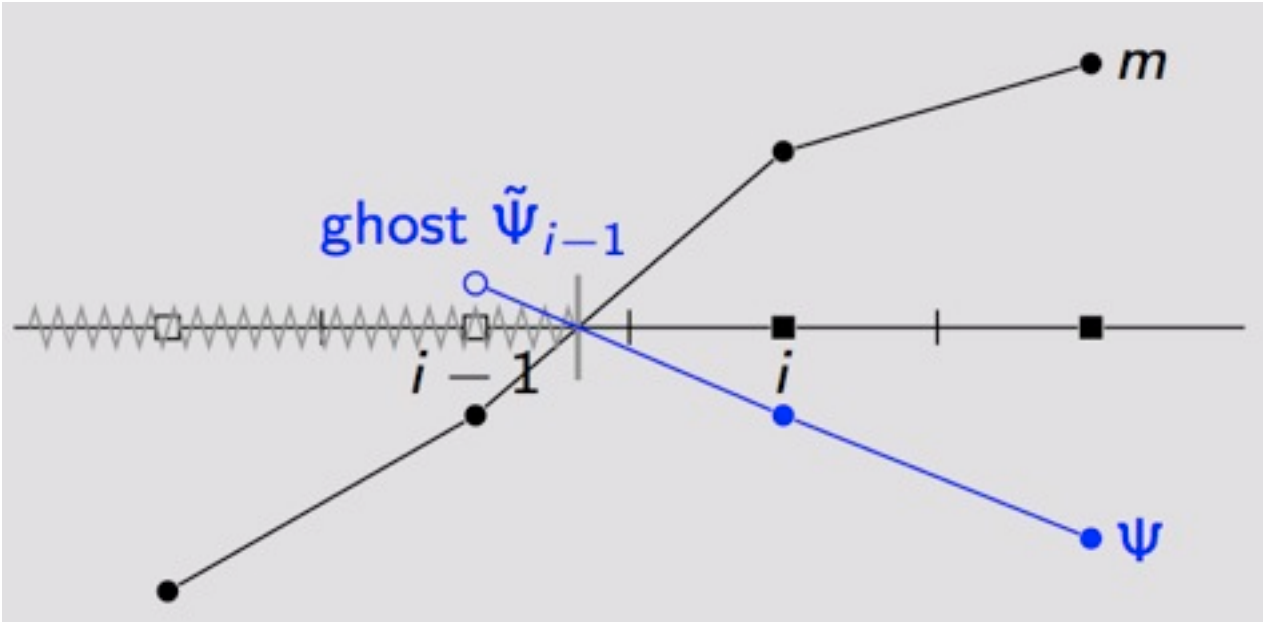
One-way coupling: for each AMR level, one needs to design a multigrid scheme for arbitrary-shaped boundary conditions.

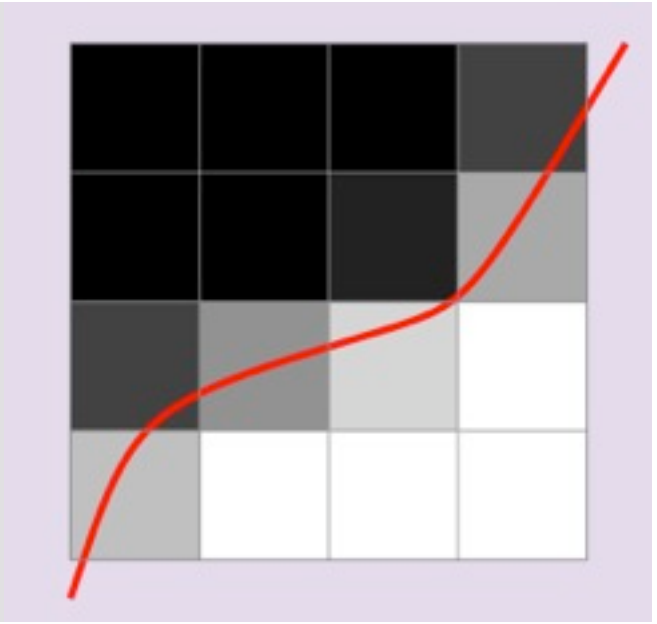


Level 3 Level 2 Level 1

Boundary-capturing technique

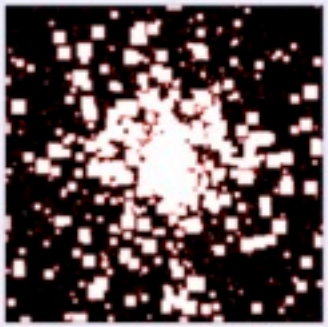
On each level, the boundary is defined as the zero-level set of a domain-fitted function (distance to the interface or volume fraction.

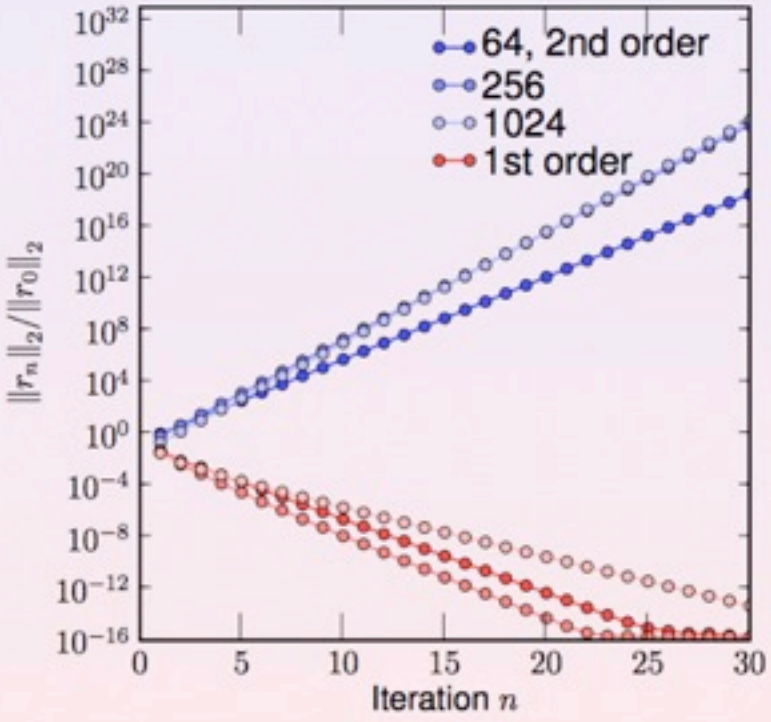


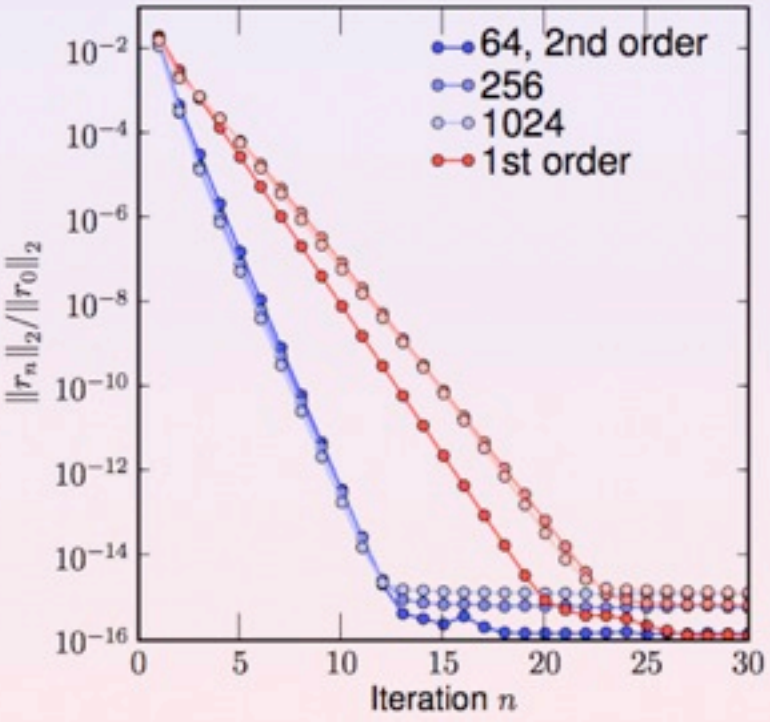


Boundary condition is enforced by linear extrapolation: second order boundary reconstruction as in Gibou (2002).

Performance of multigrid on arbitrary domains

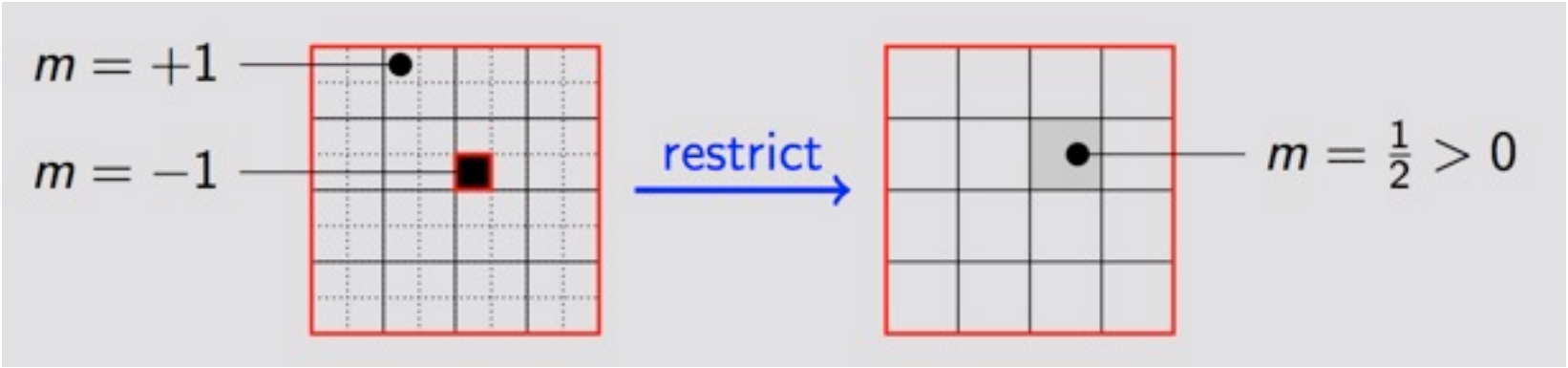






Small island-problem on complex boundaries

In some cases (holes), the boundary disappears at coarse levels.
Poisson solution on coarse levels is different from the one at fine levels.
Multigrid diverges ! (Brandt 1995).



A simple solution: Switch to first order boundary-capturing. Slower convergence, partially compensated by W-cycles (Guillet 2009)

Conclusion

- Using force shaping and small scale PP corrections, the Particle Mesh method can reach very high resolution and accuracy. The AP3M method is the ultimate N-body solver for this family. Not well for arbitrary mesh geometry.

- Question: constant force softening versus adaptive force softening ?

- PM-AMR has proven to be a powerful alternative.

- Two distinct approaches for AMR: one-way versus two-way coupling.

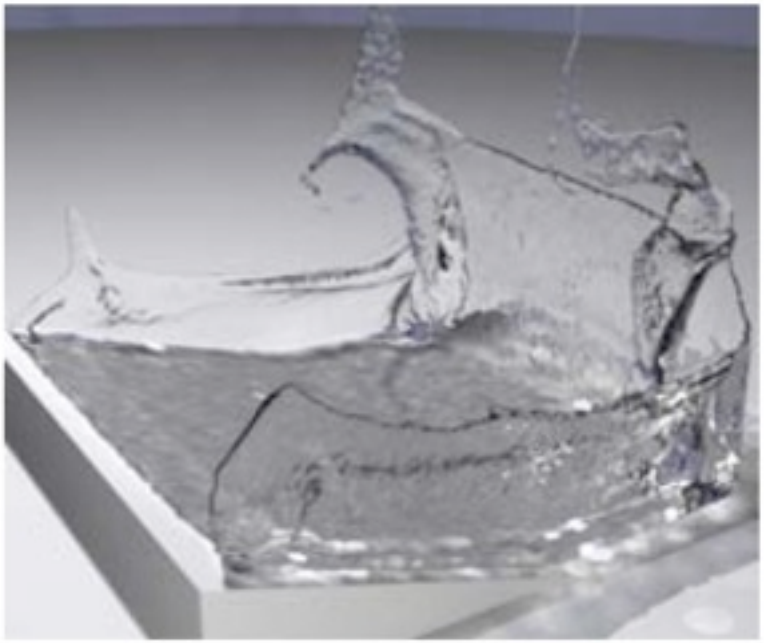
- PM-AMR reaches a similar accuracy than other codes (P3M or Tree), although force discontinuities can be an issue.

- PM-AMR is a very efficient method when used in conjunction with AMR-based multigrid solvers.

- RAMSES gravity solver implements most of these schemes.

A side note...

PIC scheme is used also for compressible and incompressible fluid flows simulations (10-20 particle per cell are needed together with adaptive grids)



Harlow, F.H., “A Machine Calculation Method for Hydrodynamics Problems”, Los Alamos Scientific Laboratory report LAMS-1956 (1955)

Brackbill, J.U., Ruppel, H.M., “FLIP: A Method for Adaptively Zoned, Particle-in-Cell Calculations of Fluid Flows in Two Dimensions”, JCP, **65**, 314 (1986)