Basic principles of cosmological simulations

Romain Teyssier
Outline

- Cosmological simulations: basic principles
- Vlasov-Poisson equations and symplectic integrators
- N body with the Adaptive Particle Mesh method
- Multigrid Poisson solver
- Cosmological initial conditions
- High performance computing in cosmology
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:

\[
\frac{df}{dt} = \frac{\partial}{\partial t} f(x, p, t) + \frac{p}{ma^2} \frac{\partial}{\partial x} f(x, p, t) - m \nabla_x \Phi(x) \frac{\partial}{\partial p} f(x, p, t) = 0
\]

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

Gravitational acceleration is given by the Poisson equation

\[
\Delta \Phi(x) = \frac{4\pi Gm}{a} \left( \int f(x, p, t) d^3p - \bar{n} \right),
\]

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 case for grids in cosmological simulations

Dark matter dynamics modeled using the Particle Mesh technique.
- High resolution is obtained using adaptively refined grids
- Refinements triggered by the local particle density: ensures low collision rates and high resolution
- Adaptive force softening (the local mesh size) but non-uniform force (poor energy conservation at coarse-fine boundaries)
- Multigrid solver for the Poisson equation is $O(N)$

Fluid dynamics modeled using finite-volume schemes.
- Developed decades go by an army of applied mathematicians and computational physicists
- Very good knowledge of the error budget
- Adaptive Mesh Refinement methodology for Godunov schemes invented in the late 1980’s by Colella and co-workers
- High-order methods are required (at least second order)
- One loses one order of accuracy at coarse-fine boundaries
The Particle-In-Cell method

N body integrator coupled to a grid-based Poisson solver

\[ \frac{dx_p}{dt} = \mathbf{v}_p \quad \text{and} \quad \frac{dv_p}{dt} = -\nabla_x \phi \]

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


The PIC or PM (Particle-Mesh) scheme has been applied to:
- Hydrodynamics (compressible, incompressible, MHD)
- Plasma physics
- Self-gravitating systems
Phase space: \( q = x_p \quad p = v_p \)

Hamiltonian: \( H(q, p) = \frac{p^2}{2} + \Phi(q) \)

The exact solution of an Hamiltonian system is energy-conserving and volume-preserving in phase-space (incompressible fluid in phase space).

The energy and the volume in phase-space are time-invariants.

\[ z = (q, p) \quad f(z) = (p, -\frac{\partial \Phi}{\partial q}) \quad \dot{z} = f(z) \]

Show that \( \nabla \cdot f = 0 \) and \( \frac{d}{dt} H = 0 \)

Using Reynold’s transport theorem in phase-space, show that the time derivative of any Lagrangian volume in phase-space is zero.
Classical First Order Time Integrators

For a symplectic map
the volume in phase-space is preserved if

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

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

- Symplectic Euler:
\[ z^{n+1} = z^n + \Delta t f(q^n, p^{n+1}) \]
\[ \det \frac{\partial F}{\partial z} = 1 \]
Kick-Drift-Kick algorithm (see also Leap Frog)

\begin{align*}
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}
\end{align*}

Exercise: check that the scheme is volume-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:

\[ W^p(x_p - x_i) = \int_{x_i - \Delta x/2}^{x_i + \Delta x/2} S(x_p - x) dx \]

The total charge in the cell is:

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

\[ \Delta_{ij}\Phi_j = \rho_i \]
\[ F_i = -\nabla_{ij}\Phi_j \]

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.

"Cloud-In-Cell" interpolation
Use of Fast Fourier Transform to solve for the Poisson equation

Poor’s man Poisson solver:

\[
\frac{\partial^2 \Phi}{\partial x^2} = \rho \\
-k^2 \tilde{\Phi}(k) = \tilde{\rho}(k) \\
\tilde{G}(k) = -\frac{1}{k^2} \\
\frac{\partial \Phi}{\partial x} = -F \\
-ik\tilde{\Phi}(k) = \tilde{F}(k) \\
\tilde{D}(k) = -ik
\]

Using finite difference approximations:

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

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.
Fine-level particles are temporarily passed to the coarse-level list to compute $\rho_c$. Fine-to-coarse information is mediated by particles only. Solve $\Delta \Phi_c = \rho_c$ on $\Omega_c$. Interpolate BC on $\partial \Omega_{c/f}$. Solve $\Delta \Phi_f = \rho_f$ on $\Omega_f$. 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 $\rho_c$ is added with CIC interpolation (proposed by Miniati & Colella, JCP, 2007).
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 \( \omega \approx \frac{2}{1 + \alpha \frac{\pi}{N}} \) \( (\alpha \text{ depends on dim. and BC}) \)

Similar performance with the Conjugate Gradient method. For a P\( \times \)P 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 ($N \log N$)!

Two-grid scheme

On the fine grid, define the residual and the error

1- Perform a few GS iterations (smoothing).

2- Restrict the residual to the coarse grid:

\[ r^{n}_{l} \rightarrow r^{0}_{l-1} \]

3- Solve for the coarse grid system:

\[ \Delta_{l-1}e^{n}_{l-1} = r^{n}_{l-1} \]

4- Prolong back the error to the fine grid:

\[ e^{\infty}_{l-1} \rightarrow e^{n+1}_{l} \]

5- Correct the fine grid solution:

\[ \Phi^{n+1}_{l} = \Phi^{n}_{l} + e^{n+1}_{l} \]

and perform a few GS iteration.
Multigrid scheme

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


**One-way coupling:** for each AMR level, one needs to design a multigrid scheme for arbitrary-shaped boundary conditions.
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
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. More robust but slower convergence (Guillet & Teyssier, JCP, 2010)
Beyond standard gravity models?

RAMSES with f(R) modified gravity

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.

Baojiu Li et al. 2011
Overall Adaptive PM force accuracy

PM $32^3$

AMR $32^3 + 6$ levels
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: a new 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

Cosmological inputs are:
- 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

- direct summation
- $P^3M$ or $AP^3M$
- parallel or vectorized $P^3M$
- distributed-memory parallel Tree
- distributed-memory parallel Tree PM
- distributed-memory parallel PM AMR

1. Peebles (1970)
3. Miyoshi & Kihara (1975)
4. White (1976)
5. Aarseth et al. (1979)
7. Davis et al. (1985)
8. White et al. (1987)
10. Suto & Sugino (1991)
11. Warren et al. (1992)
15. Governato et al. (1999)
16. Bode et al. (2001)
17. Colberg et al. (2000)
19. Springel et al. (2005)
20. Teyssier et al. (2007)
22. Alimi et al. (2012)
The first N body simulation ever
AMR codes in cosmology

Codes developed specifically for cosmology

**ENZO**: Greg Bryan, Michael Norman, Tom Abel *(Bryan et al. 1995)*

**ART**: Andrey Kravtsov, Anatoly Klypin *(Kravtsov et al. 1997)*

**RAMSES**: Romain Teyssier *(Teyssier 2002)*

Codes adapted from other fields of astrophysics

**FLASH**: The Flash group *(PARAMESH lib)*

**CHARM**: Francesco Miniati *(CHOMBO lib, Phil Colella)*
RAMSES: parallel Adaptive Mesh Refinement

- Graded octree structure: the cartesian mesh is refined on a cell by cell basis.
- Full connectivity: each oct have direct access to neighboring parent cells and to children octs (memory overhead 2 integers per cell).
- Optimize the mesh adaptivity to complex geometry but CPU overhead can be as large as 50%.

**N body module**: Particle-Mesh method on AMR grids. Poisson equation solved using a multigrid solver.

**Hydro module**: unsplit second order Godunov method (MUSCL) with various Riemann solvers and slope limiters. New CT-based MHD solver.

**Time integration**: single time step or sub-cycling.

**Other**: Radiative cooling/heating, star formation, stellar and AGN feedback, radiative transfer and MHD

MPI-based parallel computing using time-dependent domain decomposition based on Peano-Hilbert cell ordering.

Download at [http://www.itp.uzh.ch/~teyssier/Site/RAMSES.html](http://www.itp.uzh.ch/~teyssier/Site/RAMSES.html)
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
Parallel computing with locally essential trees

Parallel computing with a domain decomposition based on the *Hilbert curve*.
Tested and operational with MPI up to 76'000 core.
Scaling depends on problem size and complexity.

Each processor octree is surrounded by ghost cells (local copy of distant processor octrees) so that the resulting local octree contains all the necessary information.

Domain decomposition over 3 processors

Locally essential tree in processor #1

Locally essential tree in processor #2

Locally essential tree in processor #3
Time integration: recursive sub-cycling of fine levels
- froze coarse level during fine level solves (looses one order of accuracy)
- average Godunov fluxes in time at coarse fine boundaries

$$\left( F_{i+1/2,j}^{n+1/2,\ell} \right) = \frac{1}{\Delta t_{1}^{\ell+1} + \Delta t_{2}^{\ell+1}} \left( \Delta t_{1}^{\ell+1} \left( \frac{F_{i+1/2,j-1/4}^{n+1/4,\ell+1} + F_{i+1/2,j+1/4}^{n+1/4,\ell+1}}{2} \right) + \Delta t_{2}^{\ell+1} \left( \frac{F_{i+1/2,j-1/4}^{n+3/4,\ell+1} + F_{i+1/2,j+1/4}^{n+3/4,\ell+1}}{2} \right) \right)$$

Adaptive time stepping results in a 10x speed-up of the execution
Load balancing issues in RAMSES

Adaptive time steps and global domain decomp. lead to poor load balance.

For each problem, there is a plateau in the parallel efficiency.
Hybrid OpenMP and MPI approach

Poisson solver on 8-core nodes

- **MPI alone**
- **MPI + 4 OMP threads**
- **MPI + 8 OMP threads**

**Time** vs **Number of cores**
Galaxy formation simulations

Stars

Copyright R. Teyssier (2008)

Gas

Copyright R. Teyssier (2008)