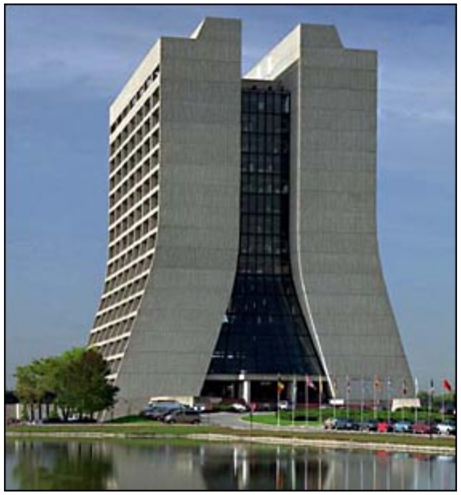
Adaptive Refinement Tree

Nick Gnedin

Theoretical Astrophysics Group

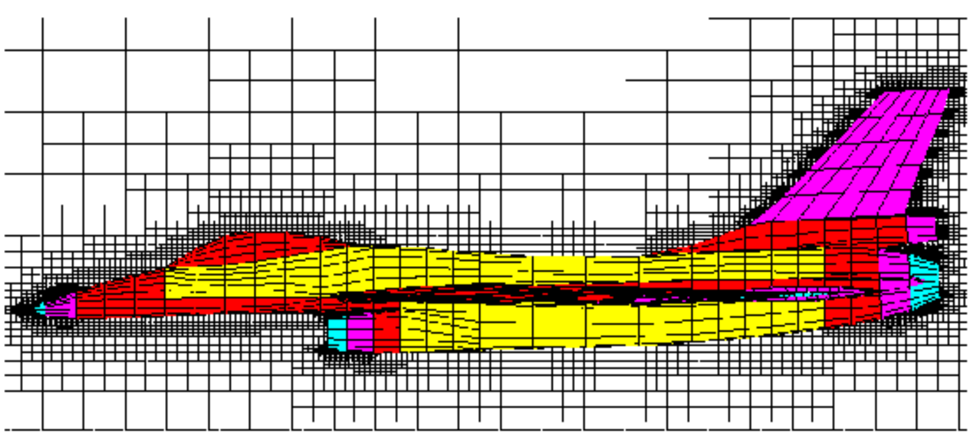


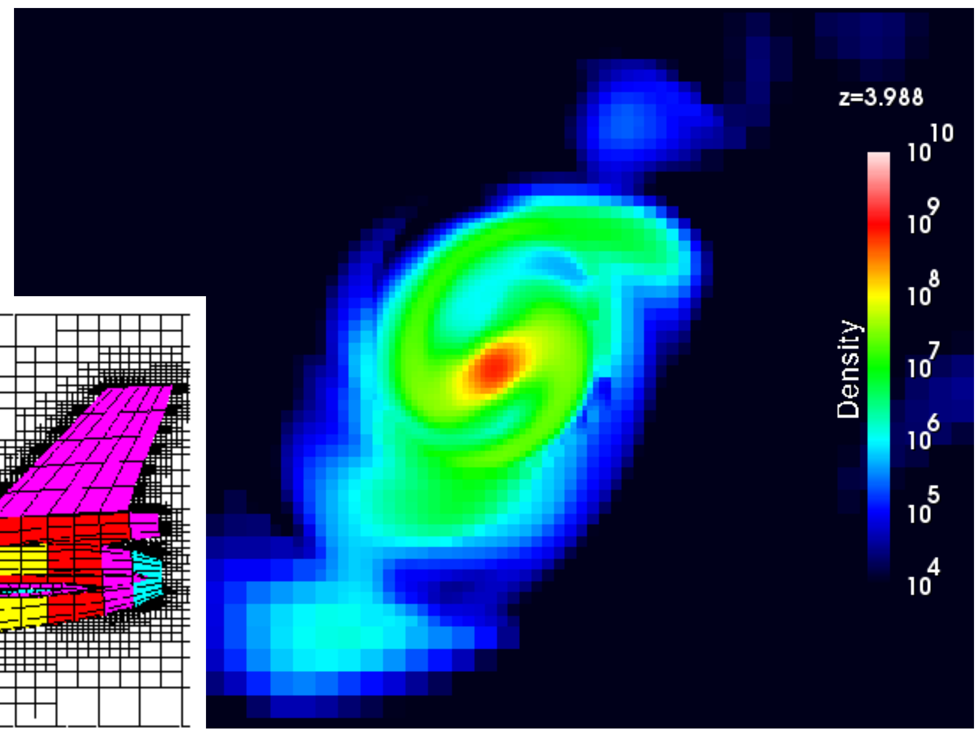
Fermilab

Cosmological Gas Dynamics

Little use of engineering expertise:

- very high resolution required
- complex physics
- gas is gravitating
- no solid boundaries





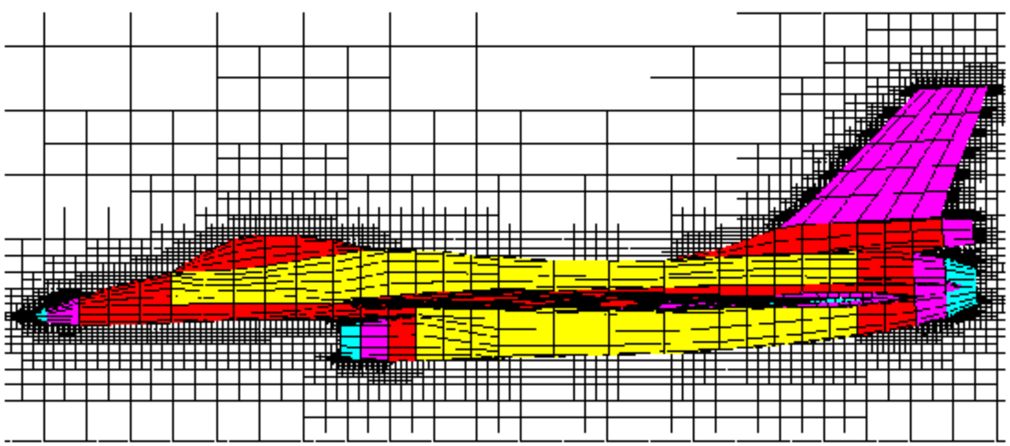
The AMR Approach

- Efficient, reliable finite element methods *for uniform grids* have been developed for solving the Poisson and gasdynamics equations.

- The *Adaptive Mesh Refinement* (AMR) methods increase the dynamic range of grid-based numerical algorithms beyond the limits imposed by existing hardware.

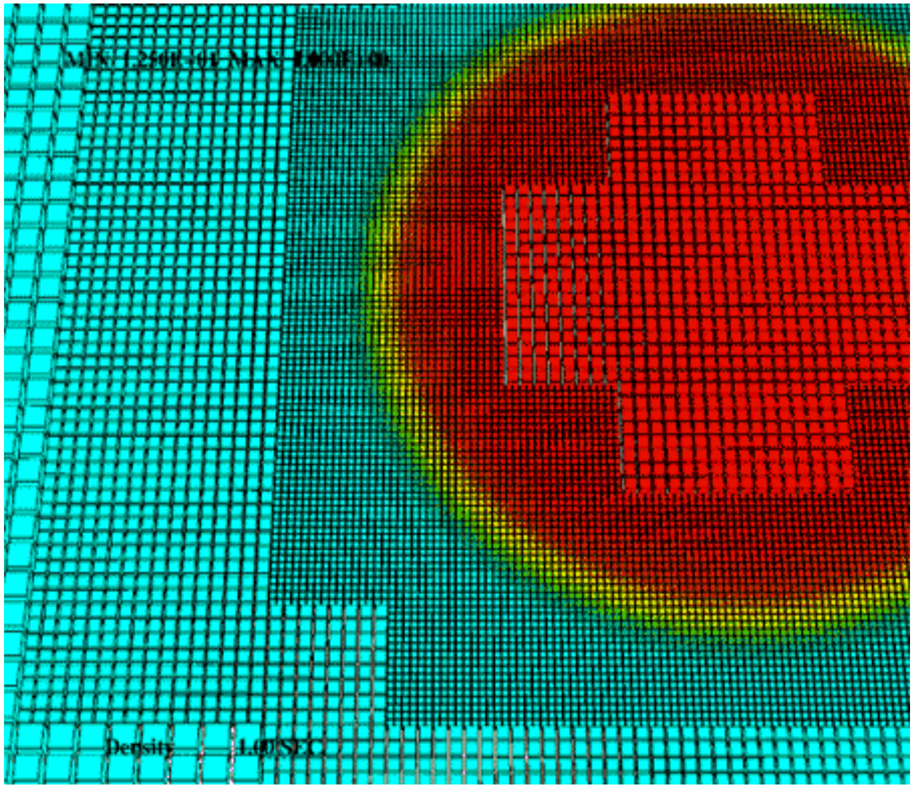
- The methods have numerous applications in different fields of physics, engineering, etc.

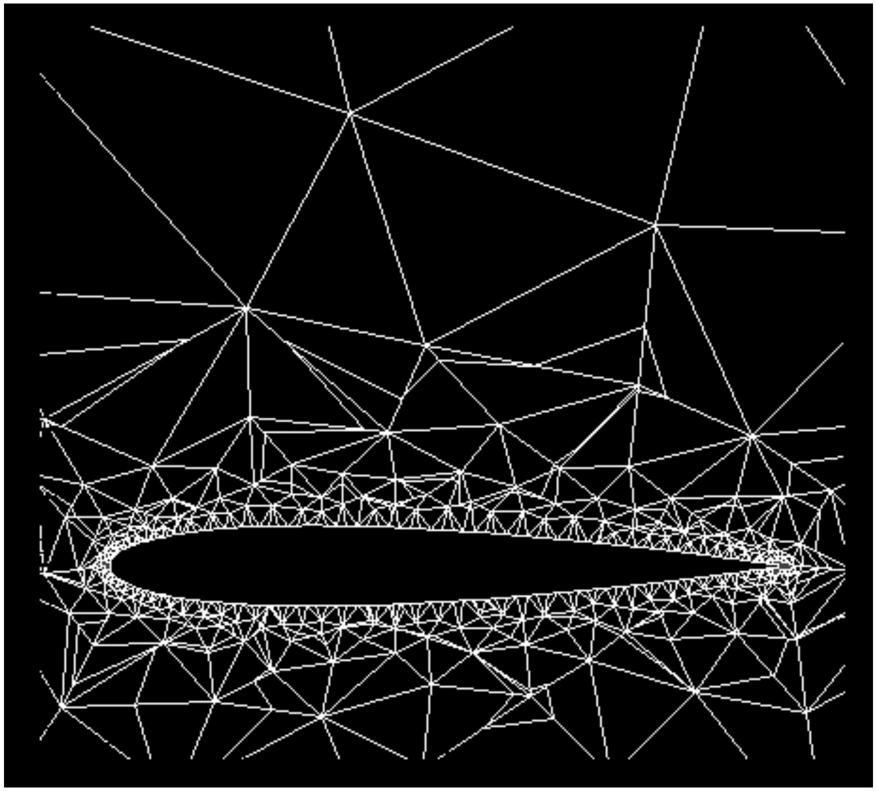
- Now gaining popularity in astrophysics and cosmology



Slide courtesy A. Kravtsov

Structured vs unstructured AMR



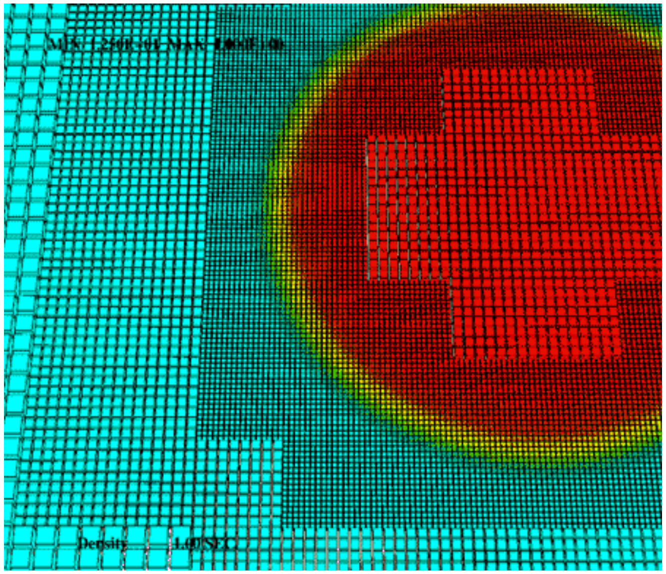


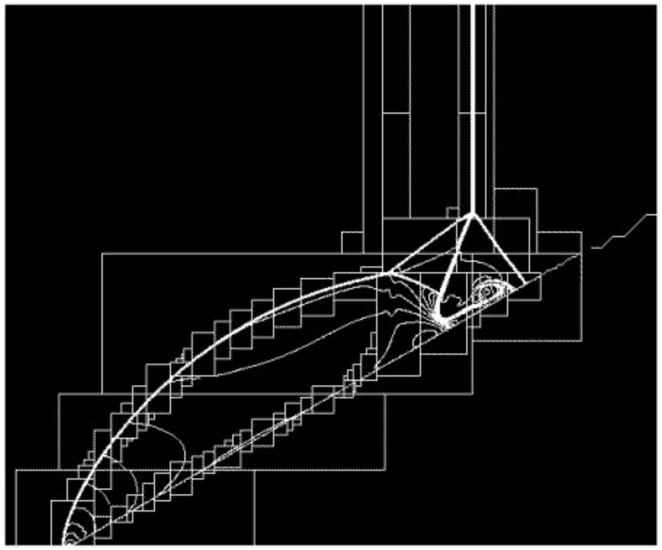
Structured: hierarchy of rectangular grids or irregularly shaped meshes of cubic cells

Unstructured: highly flexible refinement meshes, efficient for cases of complicated region geometry and boundaries; more sophisticated data structures and algorithms

Slide courtesy A. Kravtsov

Structured AMR





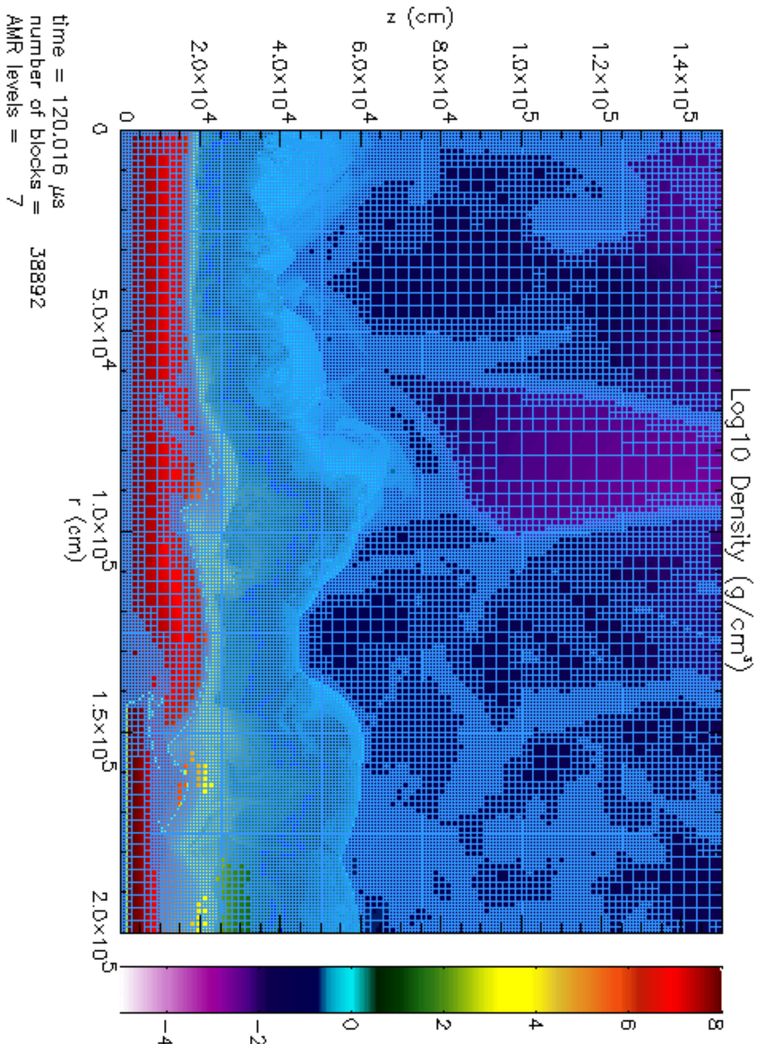
- Is most widely used in astrophysics, as there is no particular motivation to use the unstructured meshes. The finite difference techniques are simpler and often more stable and accurate with cubic cells

- First methods developed by Marsha Berger, Joseph Oliger, and Phillip Colella in the early 1980s

- The first algorithms used rectangular grids organized in hierarchy of meshes with levels of the hierarchy defined by the grid cell size (resolution)

Slide courtesy A. Kravtsov

Refining cell by cell



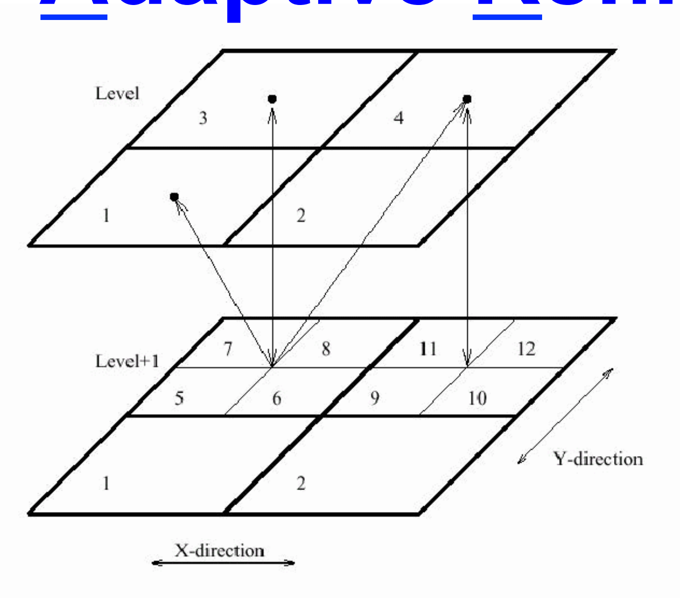
An AMR simulation with the Flash code

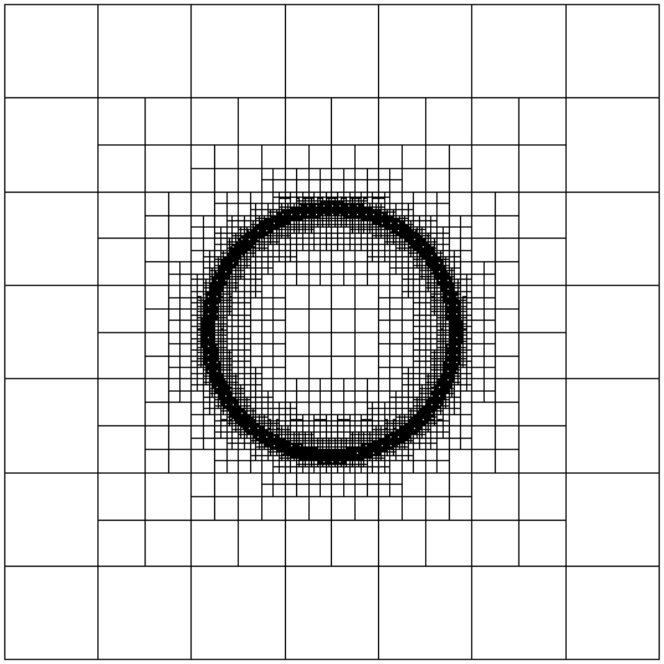
- Taken to its limit, one can think of tree of individual cells or small groups of cells (quads – 4 cells in 2D, octs – 8 cells in 3D)

- In this case, the refinement can be controlled on the level of individual cells, which allows meshes with complicated geometries to match complicated features in the systems (shocks, filaments etc.)

Slide courtesy A. Kravtsov

Adaptive Refinement Tree (ART) code

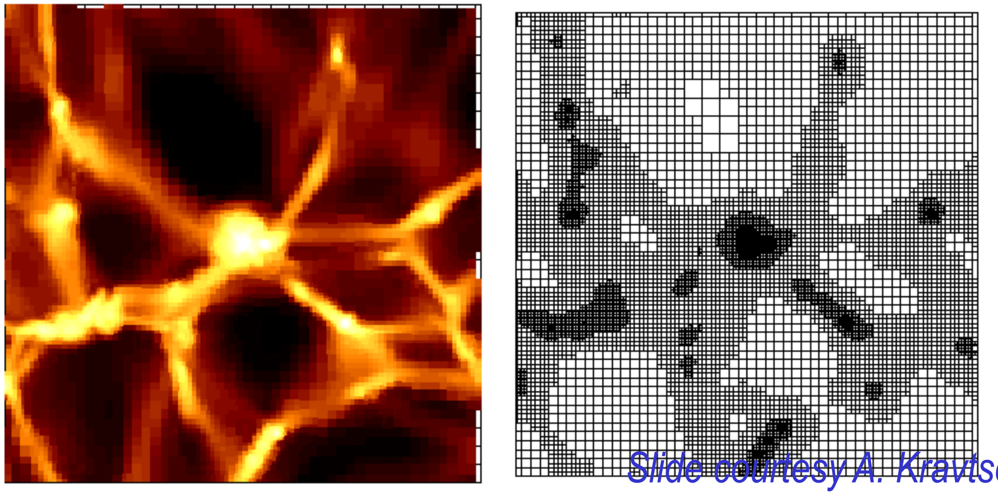




- The ART code *refines (and derefines) mesh cells individually*.

- We use a fully-threaded oct tree data structure (hence, the ART name) to support the refinement mesh hierarchy. The cost is only 2.5 storage words per cell. [Khokhlov 1998]

- This allows for flexible adaptive refinement structure that can be easily modified. The meshes can effectively match the complex geometry of filaments, sheets, and clumps in a cosmological simulation.



Slide courtesy A. Kravtsov

Modeling Physics

- Dark Matter:
  similar to plasma simulations
  implemented with particles (N-body)
- Gas Dynamics:
  need very high spatial resolution
  little use of engineering expertise
  implemented with Adaptive Refinement Tree
- Radiative Transfer:
  6D problem (= a very hard one)
  implemented only approximately
  (moment equations with a closure ansatz)

Modeling Physics

- Atomic Physics:
  basic physics is well understood
  implemented with a table lookup
- Gravity:
  just a Poisson equation
  moderately challenging to do fast with AMR
  implemented with a relaxation scheme
- Star Formation & Feedback:
  the Pandora Box of cosmological modeling
  implemented with various subgrid models
  (the best model is not yet established)

Gas Dynamics Solver

- 2^{nd} order Godunov solver with piecewise linear reconstruct [van Leer 1979; Collela & Glaz 1985; Khokhlov 1998]

- Higher order Godunov schemes are well motivated physically and are stable and have good resolution of discontinuities in the flow.

- They are enjoying wide popularity in engineering, physics, and astrophysics CFD.

- Other methods are being developed however, the most promising is WENO (e.g. Zhang & McFadyen 2006, and refs. therein).

- Improvement is desirable but not essential.

Slide courtesy A. Kravtsov

Particle Solver

- Particles are treated using standard particle-mesh methods: cloud-in -cell density assignment and force interpolation.

- 2nd order leapfrog time integration, interpolation and loss of 2nd order accuracy at the refinement mesh interfaces…

- Improvement is desirable but not essential.

Slide courtesy A. Kravtsov

Gravity Solver

- FFT solver on the uniform root grid covering the entire volume

- Relaxation solver for refined cells (Gauss-Seidel + SOR + Chebyshev accel)

- Potential on child refinement cells is inherited from the parent cells; potential from the previous step is used as initial guess for the next step.

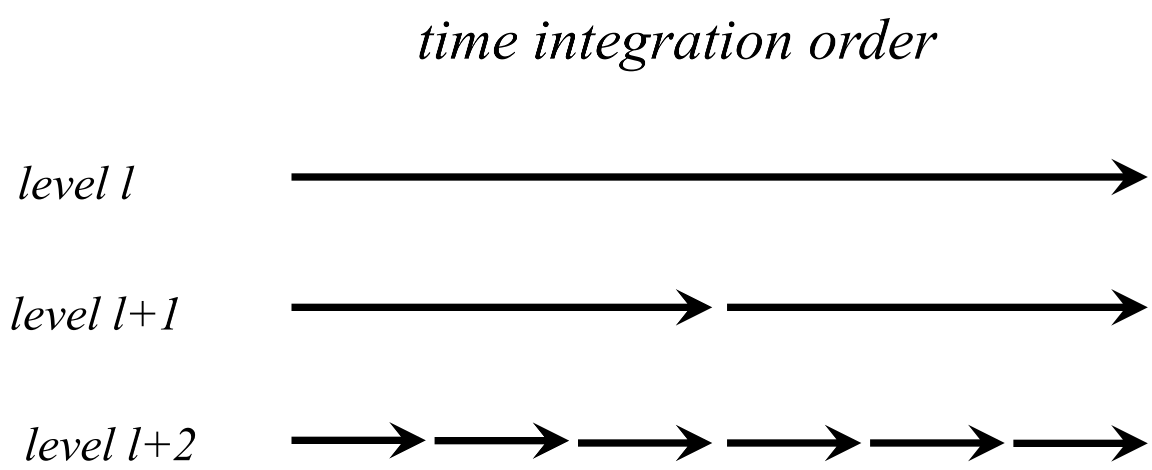
- Improvement is highly desirable.

Slide courtesy A. Kravtsov

Time-stepping

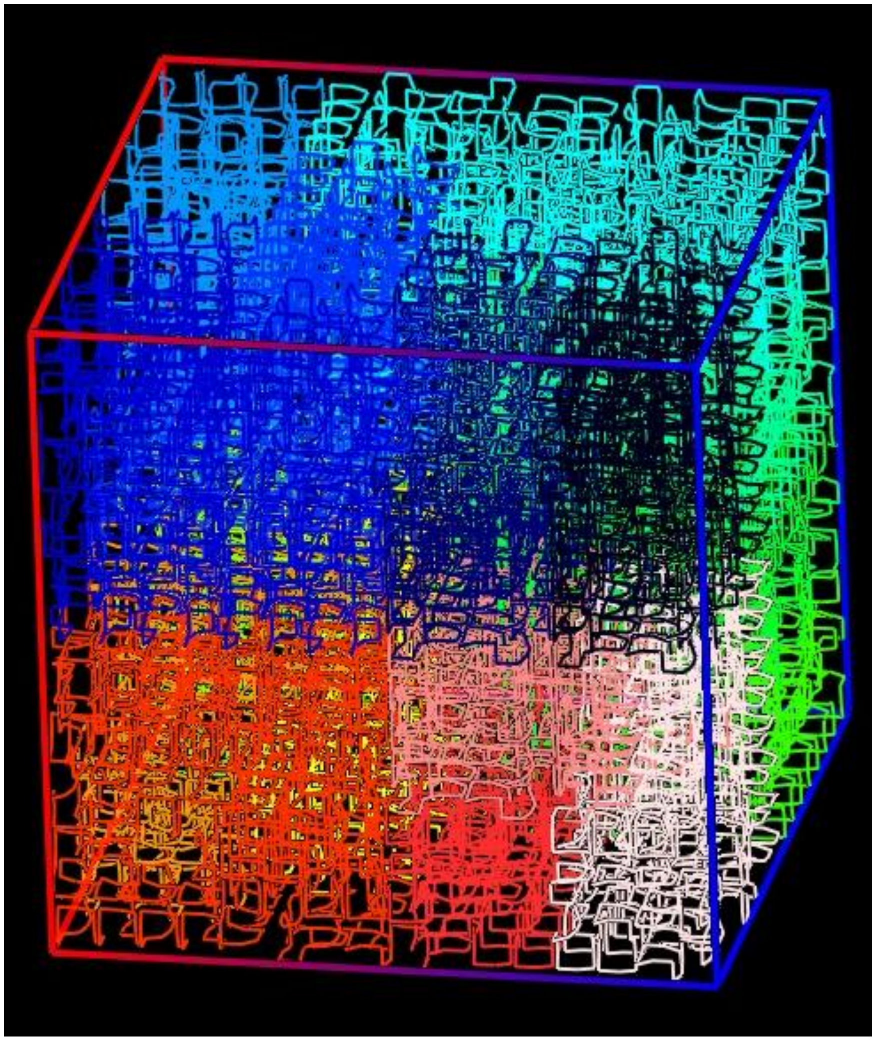
- each refinement level l is advanced with time step $dt_l = dt_{l-1} / N_l$,(for any N_l) so that the *CFL condition* is satisfied for cells on all levels. In Eulerian gasdynamics CFL condition is a must – it does not allow signal to travel faster than the speed of sound, preventing unphysical solutions, it also makes the integration stable.

$$\Delta t_0 = cfl \times \frac{\Delta x_0}{\max_i(\max_j(c_s^i + |U_{i,j}|))},$$



Slide courtesy A. Kravtsov

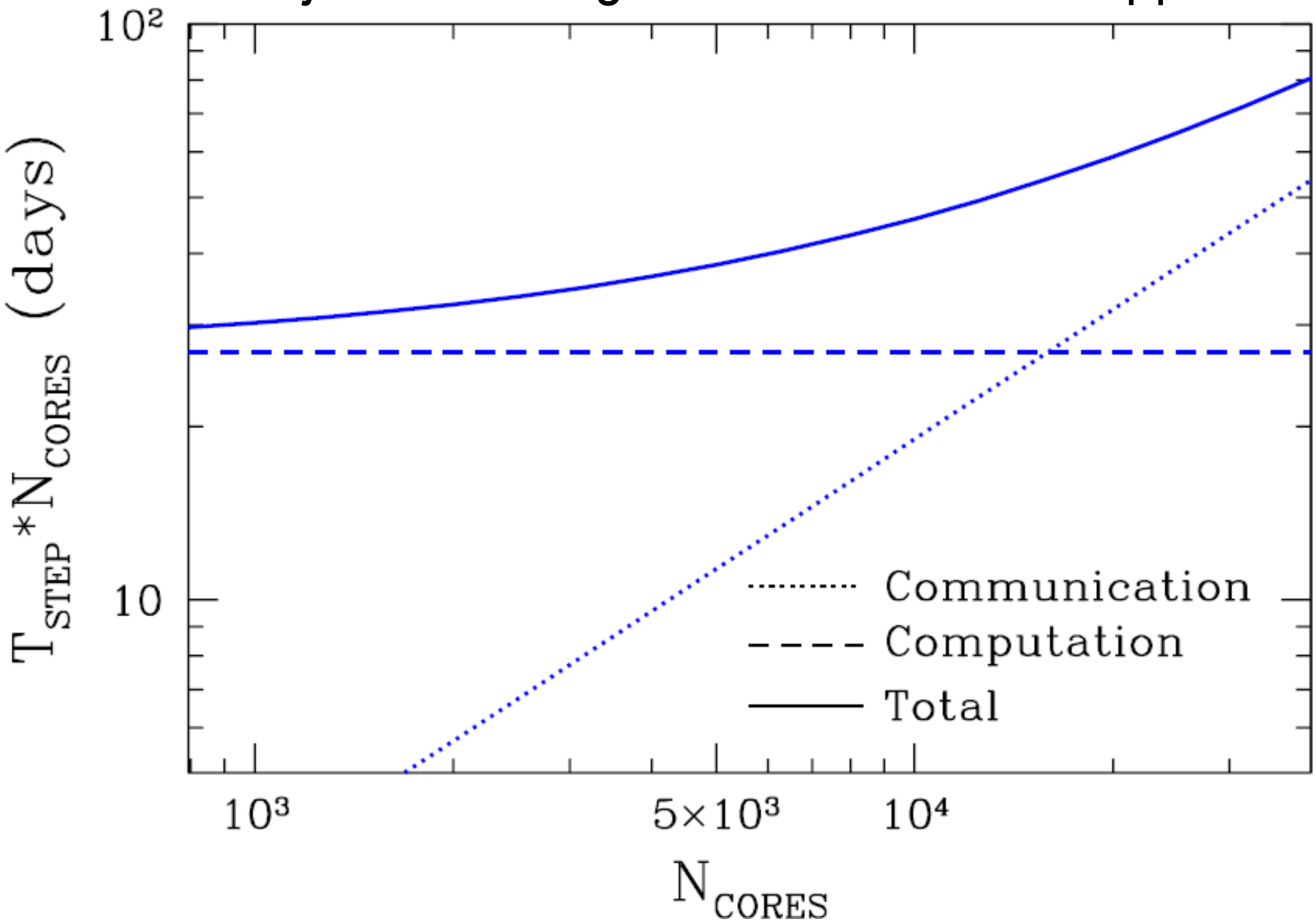
Parallelization In ART



- Hybrid mode parallelization
- OpenMP over a modest number of cores (6-8)
- MPI between tasks
- Load balance is achieved by a 3D domain decomposition using a Space Filling Curve
- SFC is restricted to the root grid only (intentional design choice; not flexible enough)

Parallel Scaling

Projected scaling for 1024^3 runs on Hopper

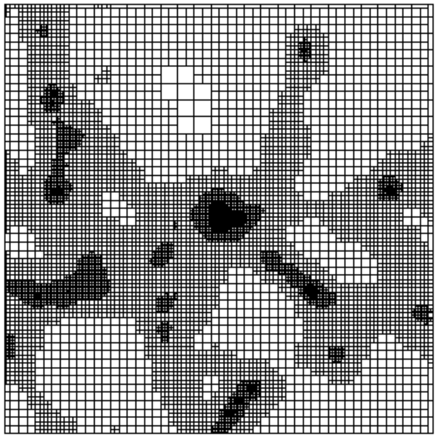


Known Problems

- OpenMP scaling within one node needs improvement:
  - currently scales to 6-8 cores
  - would like to go to 12-16 (12 would be ideal for Hopper, 16 for Mira)
  - cache access needs optimization, some part of the problem may go away when a new gravity solver is implemented

Known Problems

- Domain decomposition algorithm is not flexible enough:
  - need non-SFC based methods:
    - clustering based algorithms, (in progress)
    - graph-partitioning based algorithms (exploring)



  - in the future, use a 4D domain decomposition (use the refinement level as the 4th dimension) – will require a major algorithmic development and code refactoring

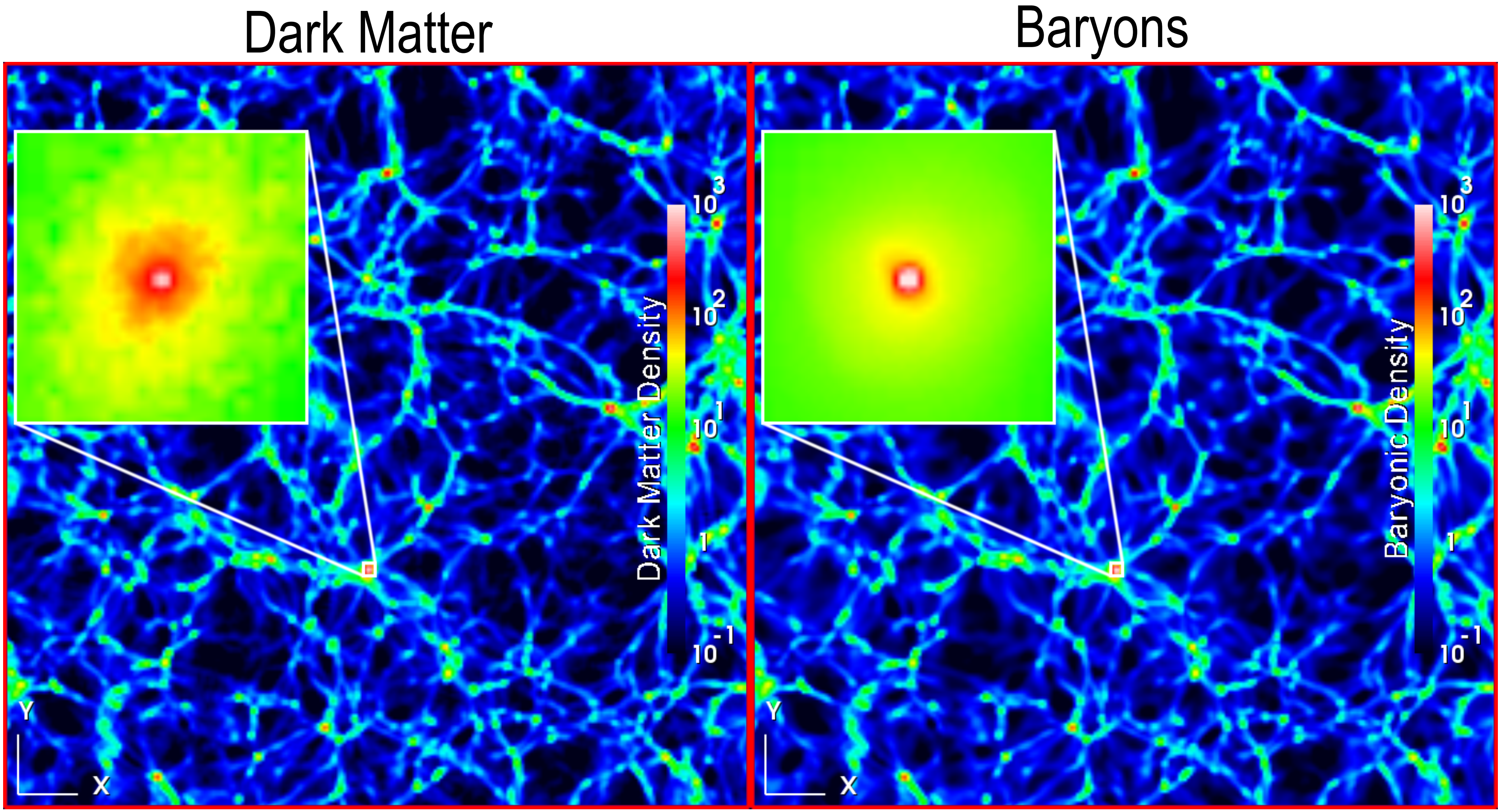
Known Problems

- Need a more accurate gravity solver, a-la Multi-Grid:
  - Relaxation-based solver is not converging fast enough
  - Measuring the accuracy of the solution is ambiguous
  - Refinement boundaries cause artificial orbit scatterings
  - Relaxation algorithms exhibit poor cache performance on modern architectures, especially for non-uniform meshes

Science Application: Baryonic Effects on Matter Clustering

- Weak lensing measurements from future surveys (DES, LSST) will provide a measurements of the ***total matter*** power spectrum.
- N-body simulations can predict the ***dark matter*** power spectrum rather precisely.
- Computing ***baryonic*** power spectrum precisely is not yet an option (unknown physics).
- Baryons will affect the total matter clustering at ~10% level, they need to be corrected out.

Find 6 Differences



We are after subtle difference in clustering of dark matter vs baryons

Simulation Plan

	Box (CHIMP)	N	Mass resol	Spatial resol	
Jing et al 2006	100	512	7e8	?	
Rudd, Zentner, Kravtsov 2008		60	256	1e9	2 kpc
Guilett, Teyssier, Colombi 2010	50	1024	1e7	1 kpc	
van Daalen et al 2011	100	512	7e8	2 kpc	
Our plan	200	1024+	7e8	<1 kpc	

These numbers are not a matter of choice, they should be set by the requirement for the simulations to numerical converge.

The End

Title