The RAMSES code and related techniques

5. AMR and parallel computing

Romain Teyssier

University of Zurich

i r f u

cea

s a c la y

Outline

- The Modified Equation
- Adaptive Mesh Refinement
- Grid refinement strategy
- Hydrodynamics and AMR
- Parallel computing and AMR

The Euler equations in conservative form

A system of 3 conservation laws

$$\partial_t \rho + \nabla \cdot \mathbf{m} = 0$$

$$\partial_t \mathbf{m} + \nabla \cdot (\rho \mathbf{u} \times \mathbf{u}) + \partial_x P = 0$$

$$\partial_t E + \nabla \cdot \mathbf{u}(E + P) = 0$$

The vector of ***conservative variables*** $(\rho, \mathbf{m}, E)$

Godunov scheme for hyperbolic systems

The system of conservation laws

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} = 0$$

is discretized using the following integral form:

$$\frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t} + \frac{\mathbf{F}_{i+1/2}^{n+1/2} - \mathbf{F}_{i-1/2}^{n+1/2}}{\Delta x} = 0$$

The time average flux function is computed using the self-similar solution of the inter-cell Riemann problem:

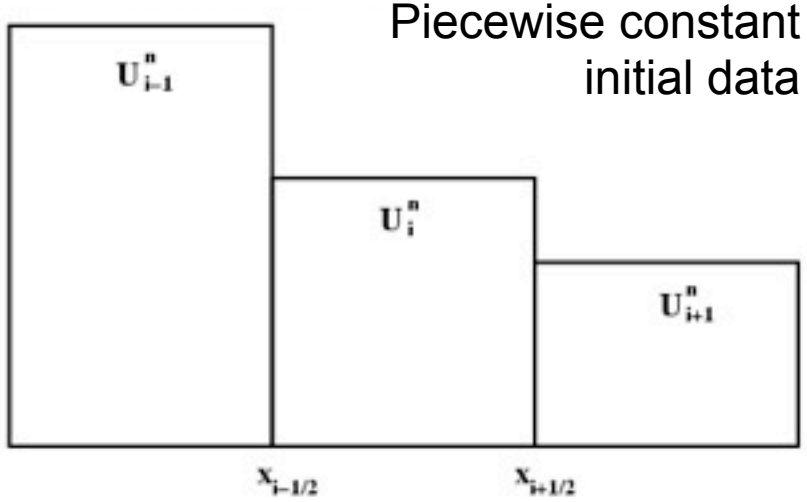
$$\mathbf{U}_{i+1/2}^*(x/t) = \mathcal{RP}\left[\mathbf{U}_i^n, \mathbf{U}_{i+1}^n\right]$$

$$\mathbf{F}_{i+1/2}^{n+1/2} = \mathbf{F}(\mathbf{U}_{i+1/2}^*(0))$$

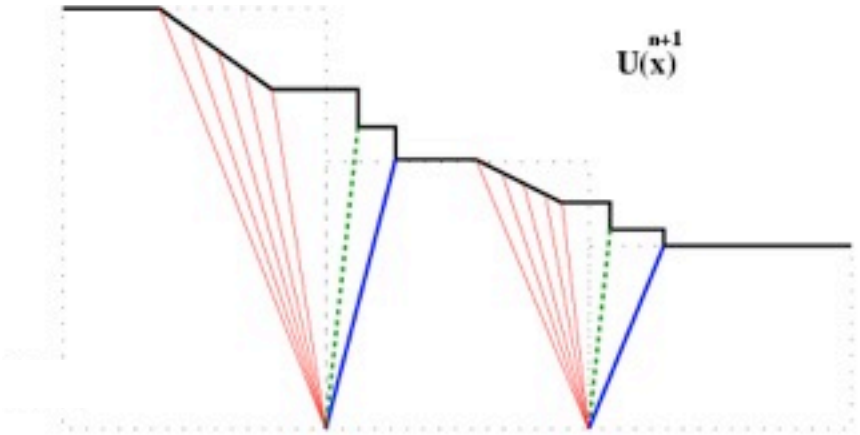
This defines the Godunov flux:

$$\mathbf{F}_{i+1/2}^{n+1/2} = \mathbf{F}^*(\mathbf{U}_i^n, \mathbf{U}_{i+1}^n)$$

Piecewise constant initial data

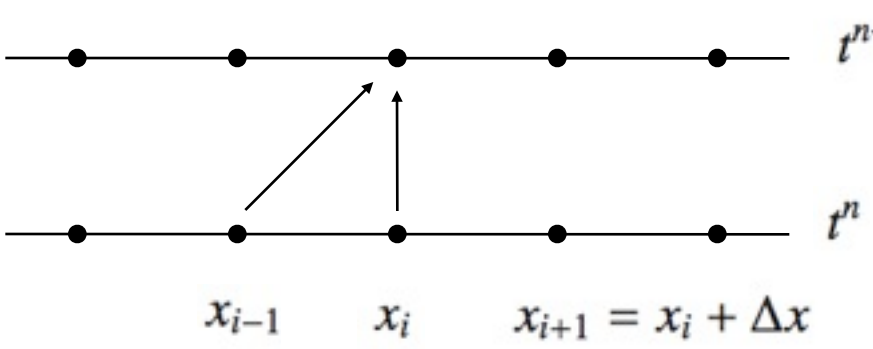


- **Godunov, S. K.** (1959), A Difference Scheme for Numerical Solution of Discontinuos Solution of Hydrodynamic Equations, *Math. Sbornik, 47, 271-306*, translated US Joint Publ. Res. Service, JPRS 7226, 1969.



Advection: 1 wave, Euler: 3 waves, MHD: 7 waves

Upwind scheme for the advection equation



a>0: use only upwind values, discard downwind variables

$$\partial_x u \simeq \frac{u_i^n - u_{i-1}^n}{\Delta x} \quad \Rightarrow \quad \frac{u_i^{n+1} - u_i^n}{\Delta t} + a\frac{u_i^n - u_{i-1}^n}{\Delta x} = 0$$

Taylor expansion up to second order:

$$\left(\frac{\partial u}{\partial t}\right) + a\left(\frac{\partial u}{\partial x}\right) = -\frac{\Delta t}{2}\left(\frac{\partial^2 u}{\partial t^2}\right) + a\frac{\Delta x}{2}\left(\frac{\partial^2 u}{\partial x^2}\right) + O(\Delta t^2, \Delta x^2)$$

Upwind scheme is stable if C<1, with $C = a\frac{\Delta t}{\Delta x}$

$$\left(\frac{\partial u}{\partial t}\right) + a\left(\frac{\partial u}{\partial x}\right) = a\frac{\Delta x}{2}(1 - C)\left(\frac{\partial^2 u}{\partial x^2}\right) + O(\Delta t^2, \Delta x^2)$$

Modified equation

2nd-order Godunov method: the MUSCL scheme

Compute second order predicted states using a Taylor expansion:

$$\begin{cases} \mathbf{W}_{i+1/2,L}^{n+1/2} = \mathbf{W}_i^n + \dfrac{\Delta t}{2}\left(\dfrac{\partial \mathbf{W}}{\partial t}\right)_i + \dfrac{\Delta x}{2}\left(\dfrac{\partial \mathbf{W}}{\partial x}\right)_i \\ \mathbf{W}_{i+1/2,R}^{n+1/2} = \mathbf{W}_{i+1}^n + \dfrac{\Delta t}{2}\left(\dfrac{\partial \mathbf{W}}{\partial t}\right)_{i+1} - \dfrac{\Delta x}{2}\left(\dfrac{\partial \mathbf{W}}{\partial x}\right)_{i+1} \end{cases}$$

$$\begin{cases} \mathbf{W}_{i+1/2,L}^{n+1/2} = \mathbf{W}_i^n + (\mathbf{I} - \mathbf{A}\dfrac{\Delta t}{\Delta x})\dfrac{\Delta x}{2}\left(\dfrac{\partial \mathbf{W}}{\partial x}\right)_i \\ \mathbf{W}_{i+1/2,R}^{n+1/2} = \mathbf{W}_{i+1}^n - (\mathbf{I} + \mathbf{A}\dfrac{\Delta t}{\Delta x})\dfrac{\Delta x}{2}\left(\dfrac{\partial \mathbf{W}}{\partial x}\right)_{i+1} \end{cases}$$

Update conservative variables using corrected Godunov fluxes

$$\mathbf{F}_{i+1/2}^{n+1/2} = \mathbf{F}^*(\mathbf{W}_{i+1/2,L}^{n+1/2}, \mathbf{W}_{i+1/2,R}^{n+1/2}) \qquad \frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t} + \frac{\mathbf{F}_{i+1/2}^{n+1/2} - \mathbf{F}_{i-1/2}^{n+1/2}}{\Delta x} = 0$$

Modified equation for the second order scheme

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}+a\frac{u_i^n-u_{i-1}^n}{\Delta x}+\frac{a}{2}(1-C)\left[\left(\frac{\partial u}{\partial x}\right)_i-\left(\frac{\partial u}{\partial x}\right)_{i-1}\right]=0$$

Taylor expansion in space and time up to third order:

$$u_i^{n+1}=u_i^n+\Delta t\left(\frac{\partial u}{\partial t}\right)+\frac{(\Delta t)^2}{2}\left(\frac{\partial^2 u}{\partial t^2}\right)+\frac{(\Delta t)^3}{6}\left(\frac{\partial^3 u}{\partial t^3}\right)$$

$$u_{i-1}^n=u_i^n-\Delta x\left(\frac{\partial u}{\partial x}\right)+\frac{(\Delta x)^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)-\frac{(\Delta x)^3}{6}\left(\frac{\partial^3 u}{\partial x^3}\right)$$

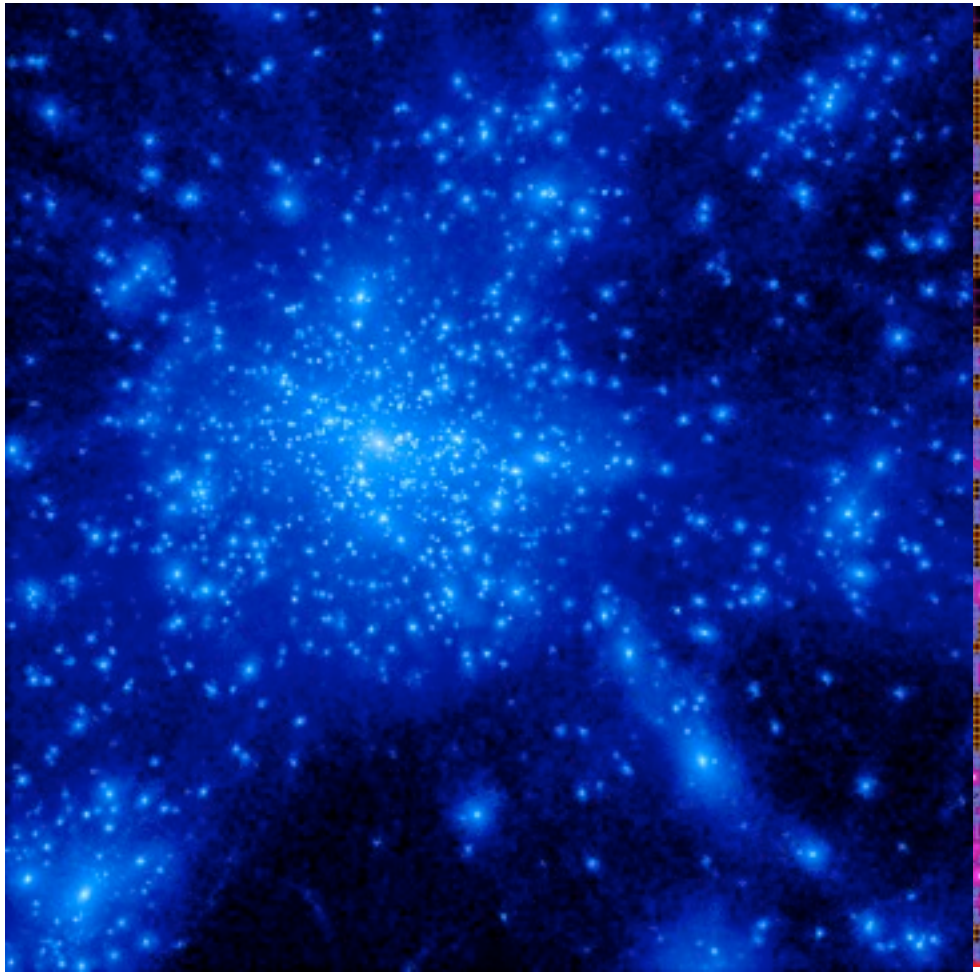
$$\left(\frac{\partial u}{\partial x}\right)_{i-1}=\left(\frac{\partial u}{\partial x}\right)_i-\Delta x\left(\frac{\partial^2 u}{\partial x^2}\right)+\frac{(\Delta x)^2}{2}\left(\frac{\partial^3 u}{\partial x^3}\right)$$

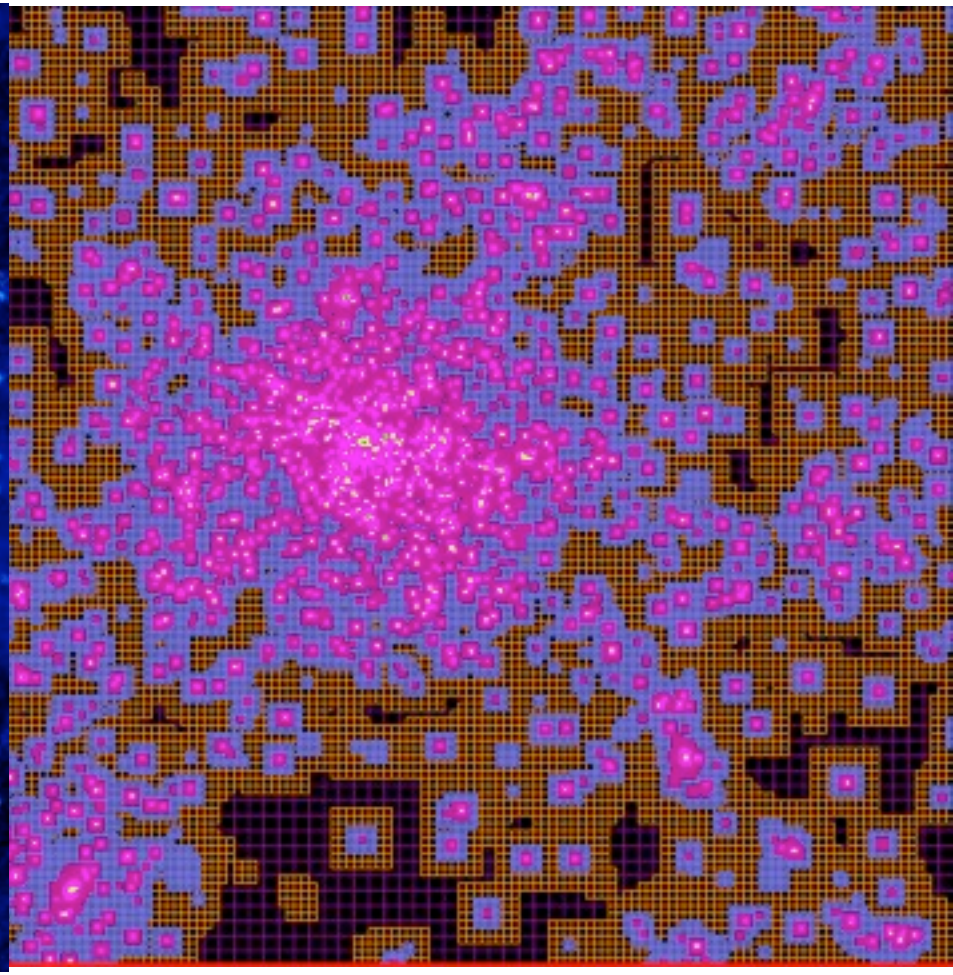
We obtain a *dispersive term* as leading-order error.

Von Neumann analysis says the scheme is stable for C<1.

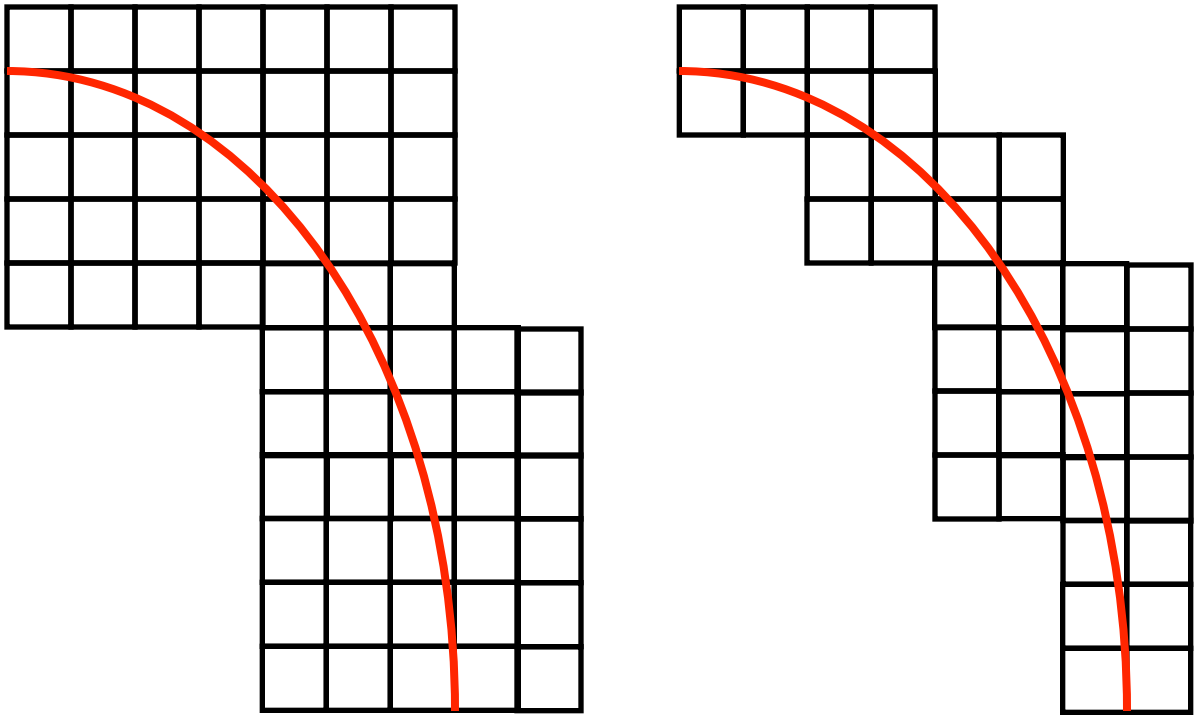
$$\left(\frac{\partial u}{\partial t}\right)+a\left(\frac{\partial u}{\partial x}\right)=a\frac{\Delta x^2}{6}(1-C)\left(\frac{1}{2}-C\right)\left(\frac{\partial^3 u}{\partial x^3}\right)+O(\Delta t^3,\Delta x^3)$$

Adaptive Mesh Refinement





Patch-based versus tree-based



AMR codes in astrophysics

ENZO: Greg Bryan, Michael Norman, Tom Abel

ART: Andrey Kravtsov, Anatoly Klypin

RAMSES: Romain Teyssier

NIRVANA: Udo Ziegler

AMRVAC: Gabor Thot and Rony Keppens

FLASH: The Flash group (PARAMESH lib)

ORION: Richard Klein, Chris McKee, Phil Colella

PLUTO: Andrea Mignone (CHOMBO lib, Phil Colella)

CHARM: Francesco Miniati (CHOMBO lib, Phil Colella)

ASTROBear: Adam Frank

and others !

RAMSES: a fully threaded graded octree

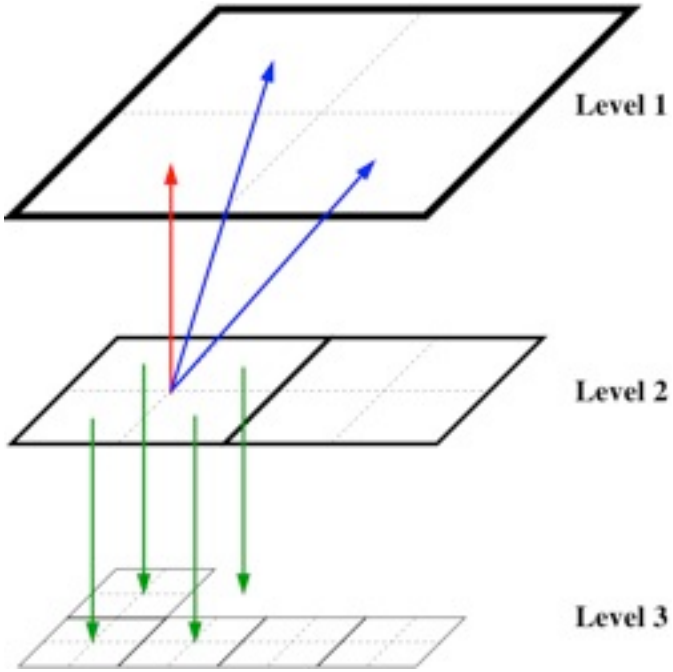
Fully Threaded Tree (Khokhlov 98).

Cartesian mesh refined on a *cell by cell basis.*

`octs`: small grid of 8 cells

Pointers (arrays of index)

- 1 parent cell
- 6 neighboring parent cells
- 8 children octs
- 2 linked list indices



Cell-centered variables are updated level by level using linked lists.

Cost = 2 integer per cell.

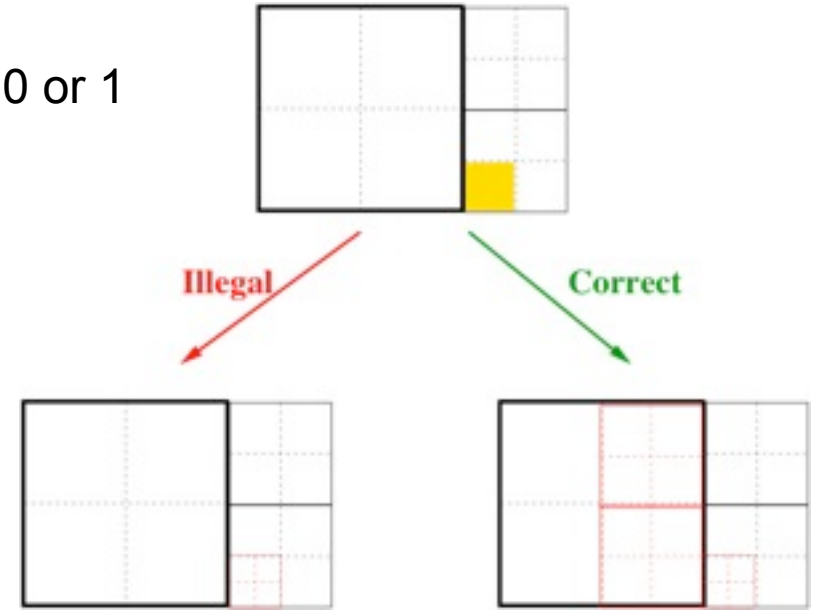
Optimize mesh adaptation to complex flow geometries, but CPU overhead compared to unigrid can be as large as 50%.

2 type of cell: - “leaf” or active cell

- “split” or inactive cell

Refinement rules for graded octrees

Compute the refinement map: flag = 0 or 1



Step 1: mesh consistency

if a split cell contains at least one split or marked cell, then mark the cell with flag = 1 and mark its 26 neighbors

Step 2: physical criteria

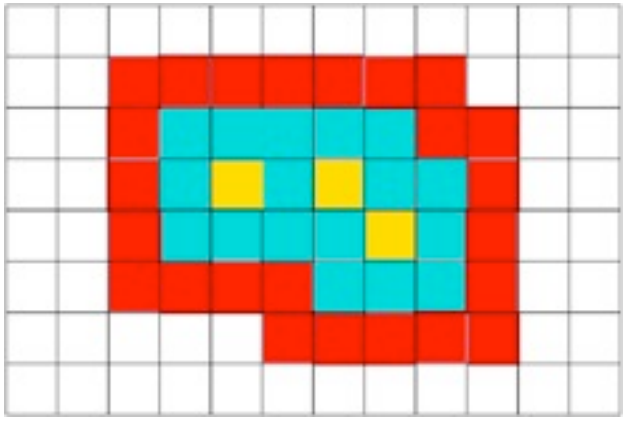
quasi-Lagrangian evolution, Jeans mass

geometrical constraints (zoom)

Truncation errors, density gradients…

Step 3: mesh smoothing

apply a dilatation operator (mathematical morphology) to regions marked for refinement → convex hull



Refinement strategies

- Refinement based on truncation errors estimate (using the Modified Equation).

- Refinement based on gradient of variables: $\Delta x/x < 1\%$

```
err_grad_d=0.01, err_grad_p=0.01
```

- Refinement based on source terms (Peclet number should be less than one): Jeans length or cooling length.

```
jeans_refine=4,4,8,8
```

- Refinement based on the mass (quasi-Lagrangian).

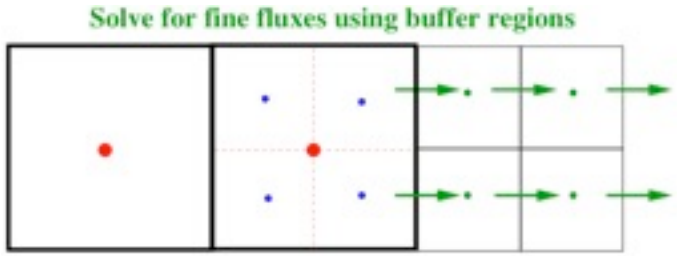
```
mass_sph=1e-4
```

- Refinement based on a color variable (geometrical criterion).

```
ivar_refine=6
```

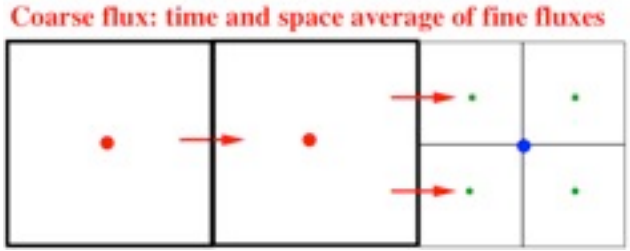
Godunov schemes and AMR

Berger & Oliger (84), Berger & Collela (89)



Prolongation (interpolation) to finer levels

- fill buffer cells (boundary conditions)
- create new cells (refinements)



Restriction (averaging) to coarser levels

- destroy old cells (de-refinements)

Flux correction at level boundary

$$(\mathbf{F}^{n+1/2,\ell}_{i+1/2,j}) = \frac{(\mathbf{F}^{n+1/2,\ell+1}_{i+1/2,j-1/4}) + (\mathbf{F}^{n+1/2,\ell+1}_{i+1/2,j+1/4})}{2}$$

Careful choice of interpolation variables (conservative or not ?)

Several interpolation strategies (with $R^T P = I$) :

- straight injection
- tri-linear, tri-parabolic reconstruction

```
interpol_var=0, 1
interpol_type=0, 1, 2, 3
```

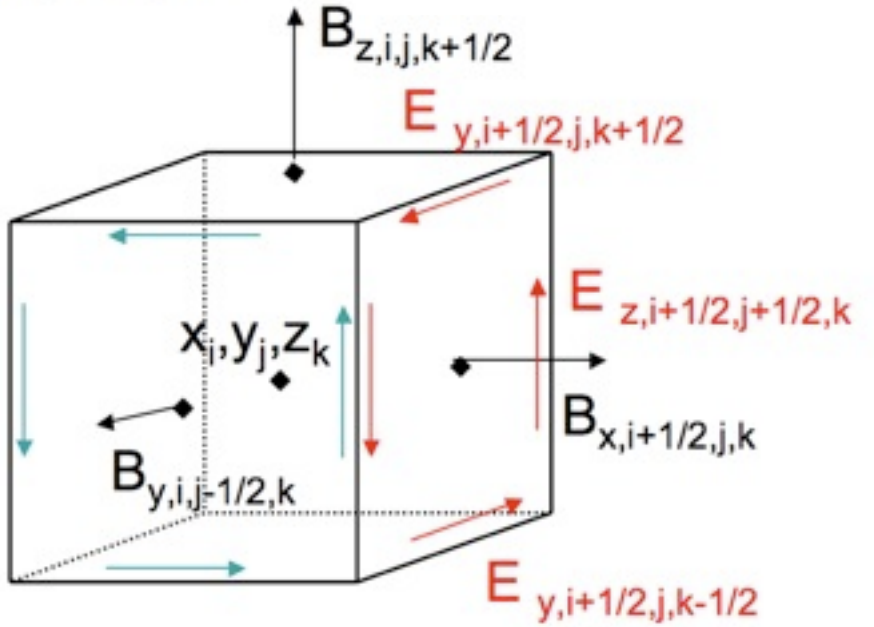
Godunov method with Constrained Transport

The induction equation in integral form suggests a surface-average form:

$$\partial_t \mathbf{B} + \nabla \times (\mathbf{B} \times \mathbf{u}) = 0 \qquad \text{(Stokes theorem)} \qquad \partial_t \int_S \mathbf{B} \cdot \mathrm{d}\mathbf{s} + \int_L (\mathbf{B} \times \mathbf{u}) \cdot \mathrm{d}\mathbf{l} = 0$$

The magnetic field is face-centred while Euler-type variables are cell-centred (staggered mesh approach).

$$(B_x)_{i+1/2,jk} = \frac{1}{S} \int_S B_x(y,z)\mathrm{d}y\mathrm{d}z \qquad S = [y_{i-1/2}, y_{i+1/2}] \times [z_{i-1/2}, z_{i+1/2}]$$



Similar to potential vector methods (Yee 1966; Dorfi 1986; Evans & Hawley 1988).

CT: exact div B preserving scheme

Surface-averaged magnetic fields are updated conservatively:

$$B^{n+1}_{z,i,j,k-1/2} = B^{n}_{x,i,j,k-1/2} + \frac{\Delta t}{\Delta x}\left(E^{n+1/2}_{y,i+1/2,j,k-1/2} - E^{n+1/2}_{y,i-1/2,j,k-1/2}\right) - \frac{\Delta t}{\Delta y}\left(E^{n+1/2}_{x,i,j+1.2,k-1/2} - E^{n+1/2}_{x,i,j-1/2,k-1/2}\right)$$

$$B^{n+1}_{y,i,j-1/2,k} = B^{n}_{y,i,j-1/2,k} + \frac{\Delta t}{\Delta z}\left(E^{n+1/2}_{x,i,j-1/2,k+1/2} - E^{n+1/2}_{x,i,j-1/2,k-1/2}\right) - \frac{\Delta t}{\Delta x}\left(E^{n+1/2}_{z,i+1/2,j-1/2,k} - E^{n+1/2}_{z,i-1/2,j-1/2,k}\right)$$

$$B^{n+1}_{x,i-1/2,j,k} = B^{n}_{x,i-1/2,j,k} + \frac{\Delta t}{\Delta y}\left(E^{n+1/2}_{z,i-1/2,j+1/2,k} - E^{n+1/2}_{z,i-1/2,j-1/2,k}\right) - \frac{\Delta t}{\Delta z}\left(E^{n+1/2}_{y,i-1/2,j,k+1/2} - E^{n+1/2}_{y,i-1/2,j,k-1/2}\right)$$

using time-averaged electric fields defined at cell edge centres:

$$E^{n+1/2}_{x,i,j-1/2,k-1/2} = \frac{1}{\Delta t \Delta x}\int_{t^n}^{t^{n+1}}\int_{x_{i-1/2}}^{x_{i+1/2}} E_x(x, y_{j-1/2}, z_{k-1/2})\mathrm{d}t\mathrm{d}x$$

$$E^{n+1/2}_{y,i-1/2,j,k-1/2} = \frac{1}{\Delta t \Delta y}\int_{t^n}^{t^{n+1}}\int_{y_{j-1/2}}^{y_{j+1/2}} E_y(x_{i-1/2}, y, z_{k-1/2})\mathrm{d}t\mathrm{d}y$$

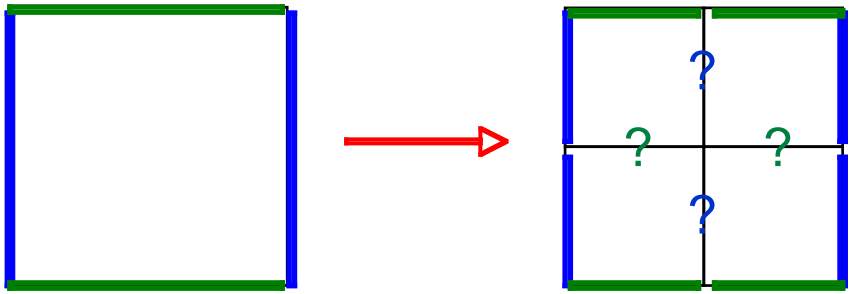
$$E^{n+1/2}_{z,i-1/2,j-1/2,k} = \frac{1}{\Delta t \Delta z}\int_{t^n}^{t^{n+1}}\int_{z_{k-1/2}}^{z_{k+1/2}} E_z(x_{i-1/2}, y_{j-1/2}, z)\mathrm{d}t\mathrm{d}z$$

The total flux (div B) across each cell bounding surface vanishes exactly.

Higher-order schemes and AMR

« Divergence-free preserving » restriction and prolongation operators

Balsara (2001) Toth & Roe (2002)



Flux conserving interpolation and averaging within cell faces using TVD slopes in 2 dimensions

EMF correction for conservative update at coarse-fine boundaries

$$\langle E_z \rangle_k^\ell \, \Delta z_\ell \Delta t_\ell^n = \langle E_z \rangle_{2k}^{\ell+1} \, \Delta z_{\ell+1} \Delta t_{\ell+1}^{2n} + \langle E_z \rangle_{2k+1}^{\ell+1} \, \Delta z_{\ell+1} \Delta t_{\ell+1}^{2n+1}$$

For a fully second-order MUSCL scheme for MHD with AMR, see Teyssier et al. 2006; Fromang et al. 2006.

More on buffer cells

Buffer cells provide boundary conditions for the underlying numerical scheme. The number of required buffer cells depends on the kernel of the chosen numerical method. *The kernel is the ensemble of cells on the grid on which the solution depends*.

- First Order Godunov: 1 cell in each direction

$$u_i^{n+1} = u_i^n(1-C) + u_{i-1}^n C$$

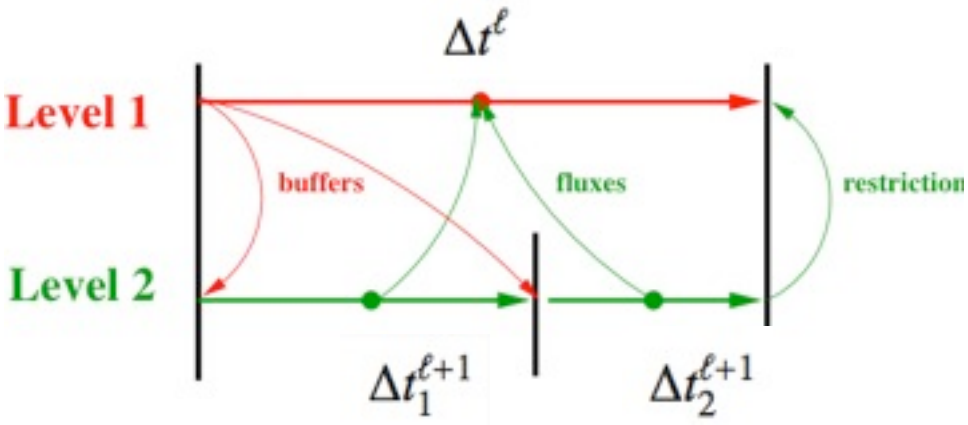
- Second order MUSCL: 2 cells in each direction

$$u_i^{n+1} = u_i^n(1-C) + u_{i-1}^n C - \frac{C}{2}(1-C)\left(\Delta u_i - \Delta u_{i-1}\right) = 0$$

- Runge-Kutta or PPM: 3 cells in each direction

Simple octree AMR requires 2 cells maximum. For higher-order schemes (WENO), we need to have a different data structure (patch-based AMR or augmented octree AMR).

Adaptive Time Stepping



Time integration: single time step or recursive sub-cycling

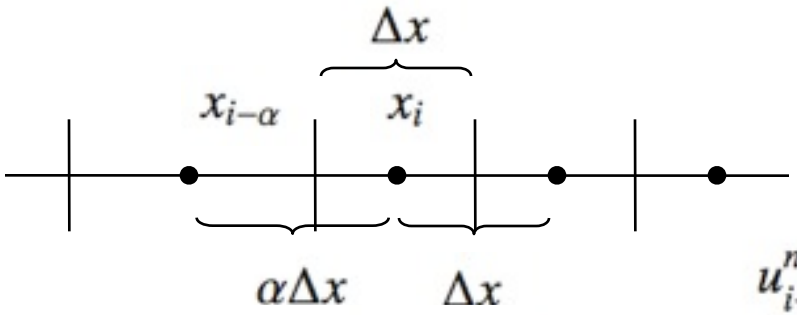
- froze coarse level during fine level solves (one order of accuracy down !)
- average fluxes in time at coarse fine boundaries

$$(\mathbf{F}_{i+1/2,j}^{n+1/2,\ell}) = \frac{1}{\Delta t_1^{\ell+1} + \Delta t_2^{\ell+1}} \left(\Delta t_1^{\ell+1} \frac{(\mathbf{F}_{i+1/2,j-1/4}^{n+1/4,\ell+1}) + (\mathbf{F}_{i+1/2,j+1/4}^{n+1/4,\ell+1})}{2} + \Delta t_2^{\ell+1} \frac{(\mathbf{F}_{i+1/2,j-1/4}^{n+3/4,\ell+1}) + (\mathbf{F}_{i+1/2,j+1/4}^{n+3/4,\ell+1})}{2} \right)$$

ART and RAMSES: update fine level first

ENZO: update coarse level first

The AMR catastrophe



Assume a and C>0.

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}+a\frac{u_{i+1/2}^{n+1/2}-u_{i-1/2}^{n+1/2}}{\Delta x}=0$$

$$u_{i+1/2}^{n+1/2}=u_i^n+(1-C)\frac{\Delta x}{2}\left(\frac{\partial u}{\partial x}\right)_i$$

$$u_{i-1/2}^{n+1/2}=u_{i-\alpha}^n+(2\alpha-1-C)\frac{\Delta x}{2}\left(\frac{\partial u}{\partial x}\right)_{i-1}$$

First order scheme: $$\left(\frac{\partial u}{\partial t}\right)+\alpha a\left(\frac{\partial u}{\partial x}\right)=a\frac{\Delta x}{2}(\alpha^2-C)\left(\frac{\partial^2 u}{\partial x^2}\right)+O(\Delta t^2,\Delta x^2)$$

Second order scheme: $$\left(\frac{\partial u}{\partial t}\right)+a\left(\frac{\partial u}{\partial x}\right)=a\frac{\Delta x}{2}(\alpha-C)(1-\alpha)\left(\frac{\partial^2 u}{\partial x^2}\right)+O(\Delta t^2,\Delta x^2)$$

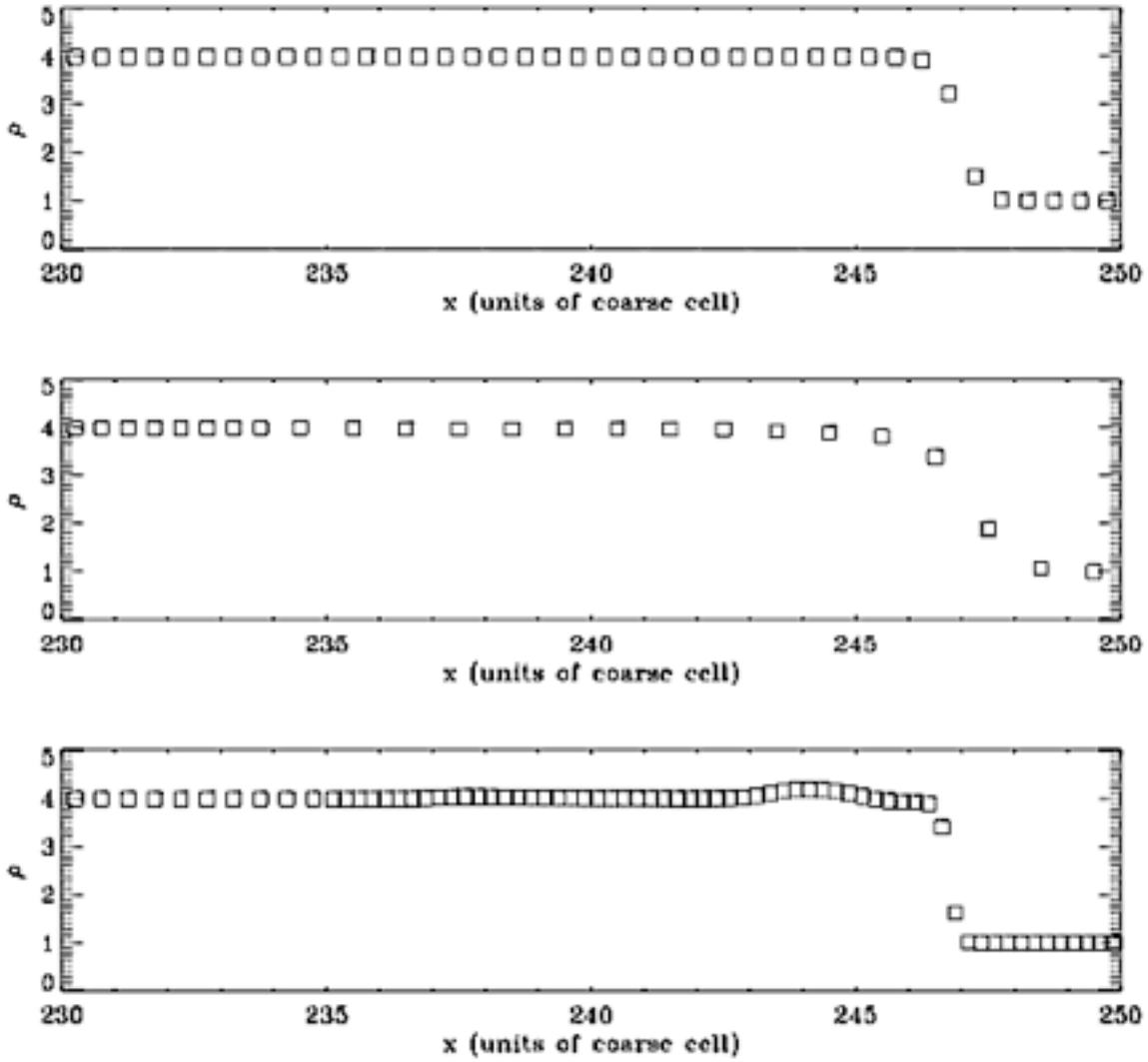
At level boundary, we loose one order of accuracy in the modified equation.

First order scheme: the AMR extension is *not consistent* at level boundary.

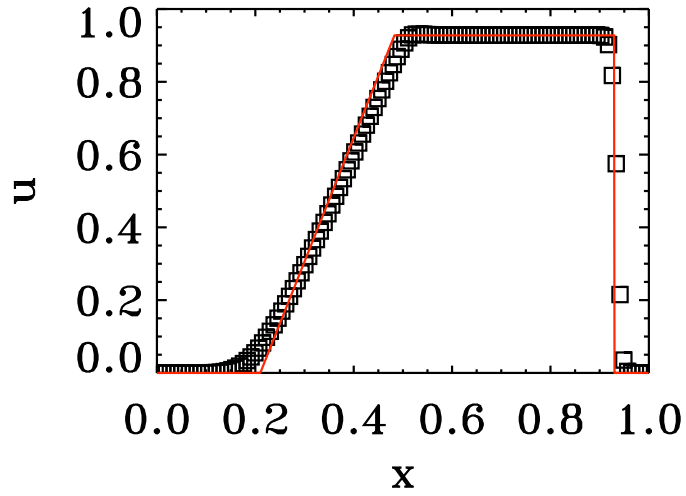
Second order scheme: for α=1.5, AMR is *unstable* at level boundary.

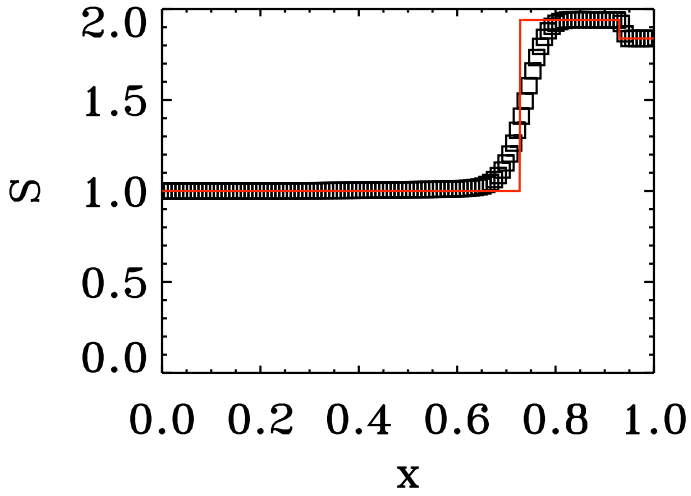
Solutions: 1- refine gradients, 2- enforce first order, 3- add artificial diffusion

Shock wave propagating through level boundary

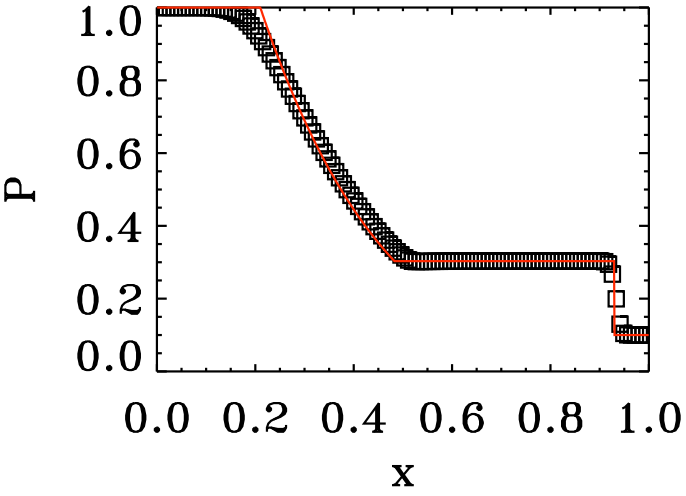


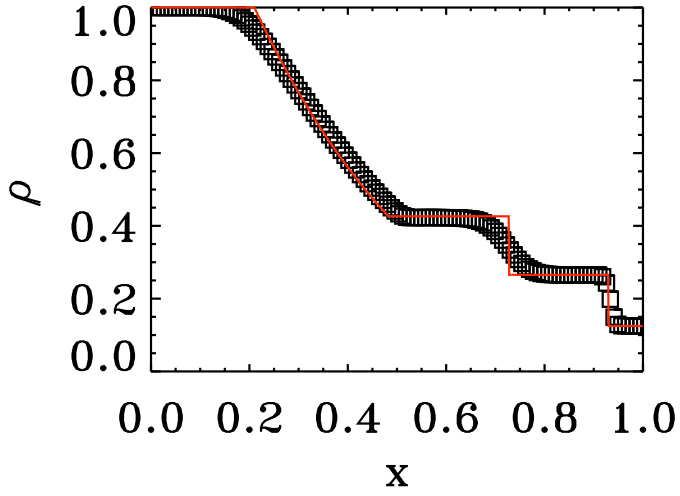
Sod test with HLLC first order



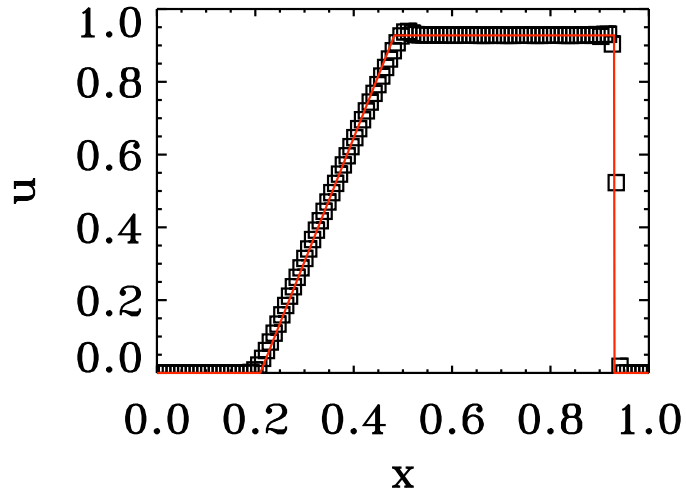


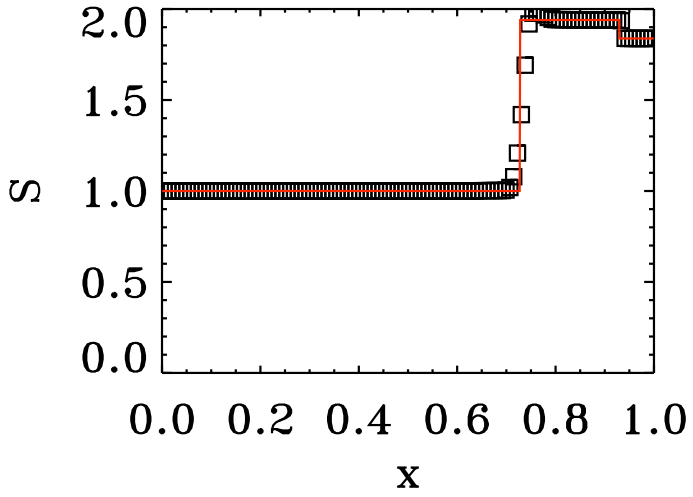
128 cells



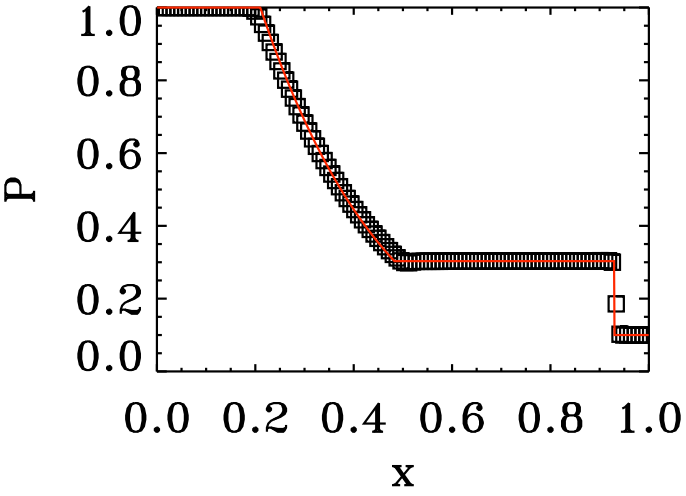


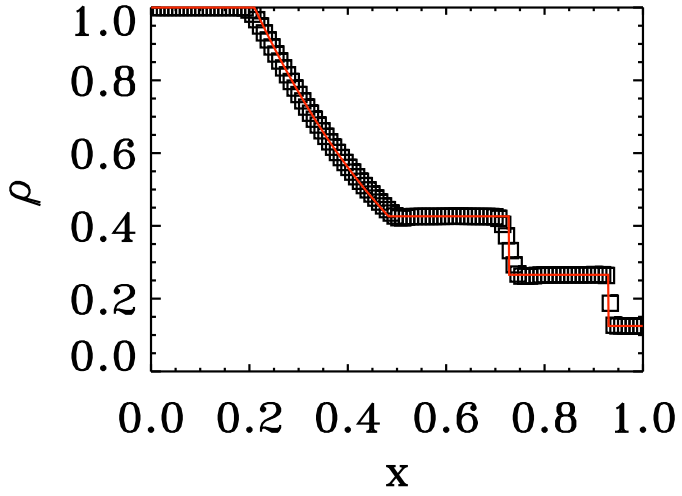
Sod test with HLLC and MonCen



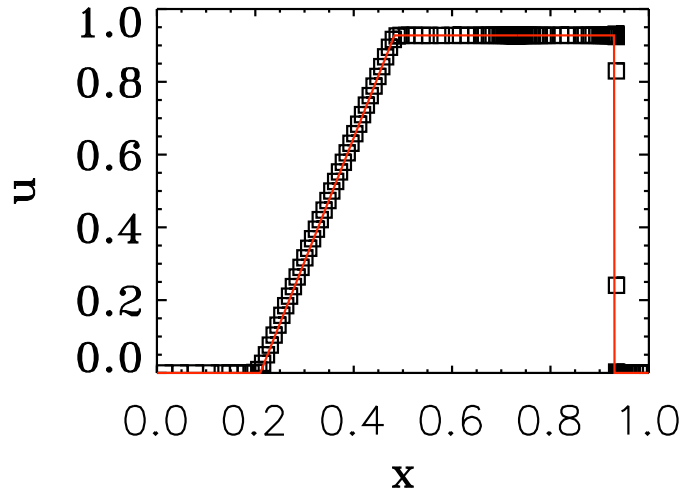


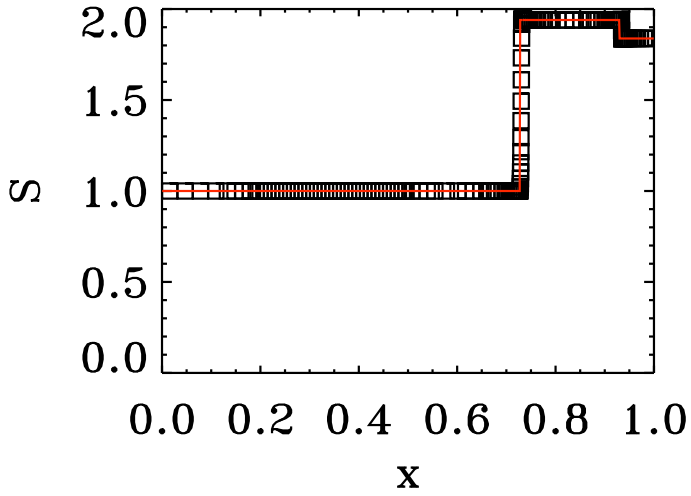
128 cells



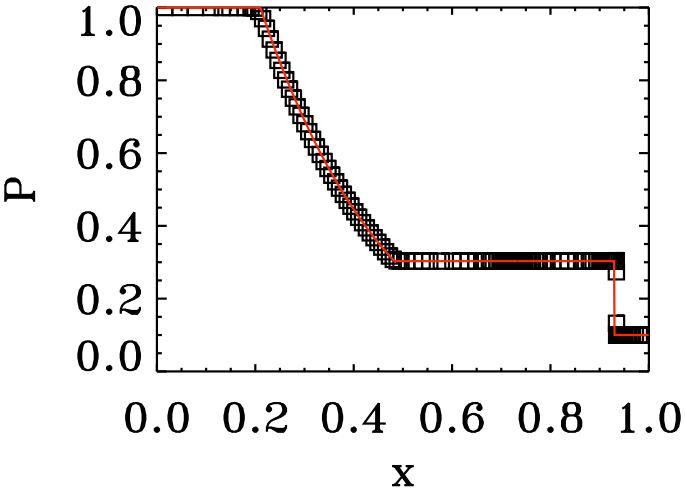


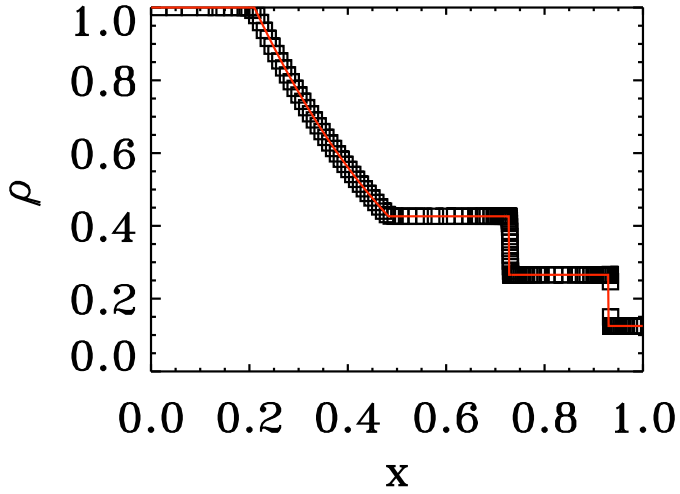
Sod test with HLLC and AMR



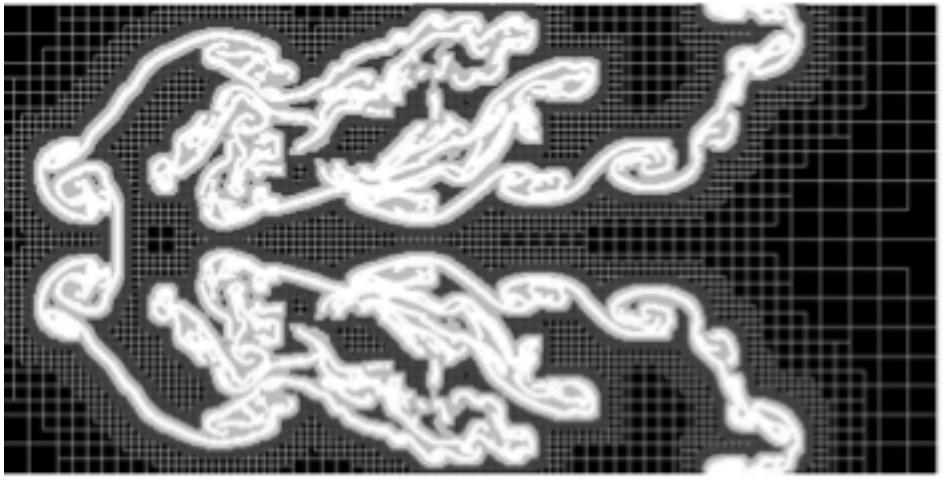


153 cells



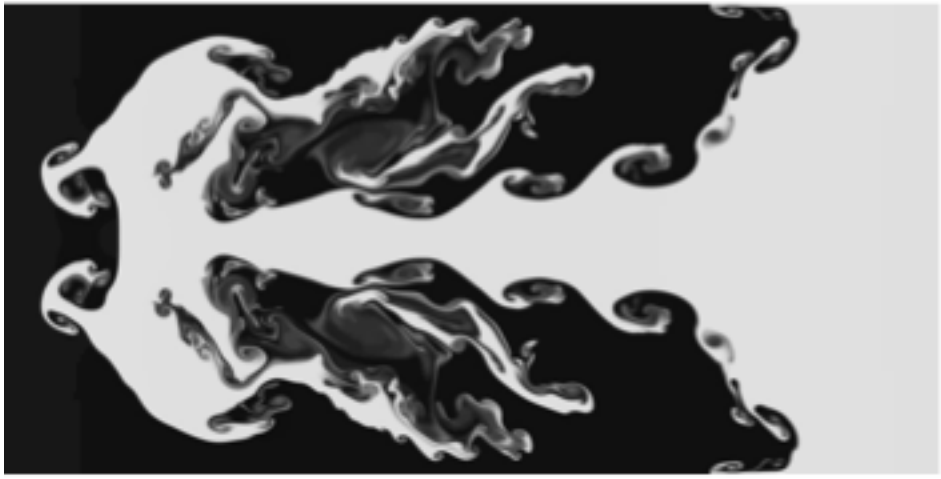


Complex geometry with AMR



Maximum numerical dissipation occurs at the 2 fluids interface.

The optimal refinement strategy is based on density gradients.



The number of required cells is directly related to the *fractal exponent* n of the 2D surface.

$$N_{cell} \propto (\Delta x)^{-n}$$

Cosmology with AMR

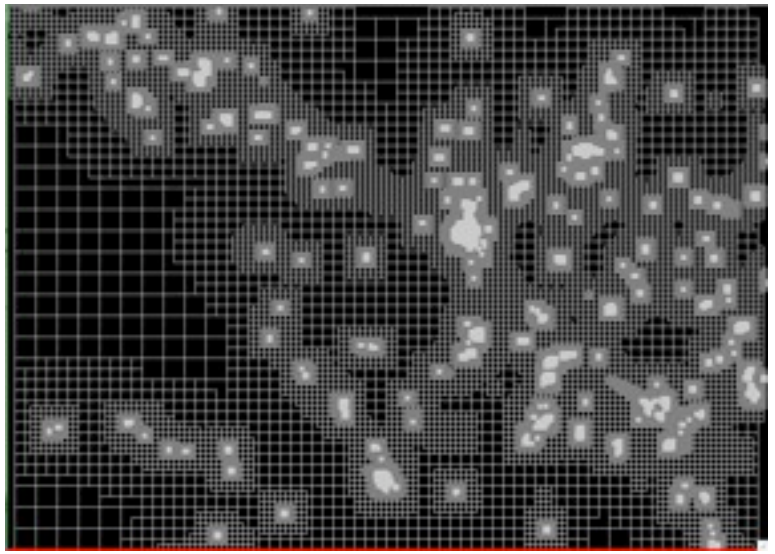
Particle-Mesh on AMR grids:
Cloud size equal to the local mesh spacing

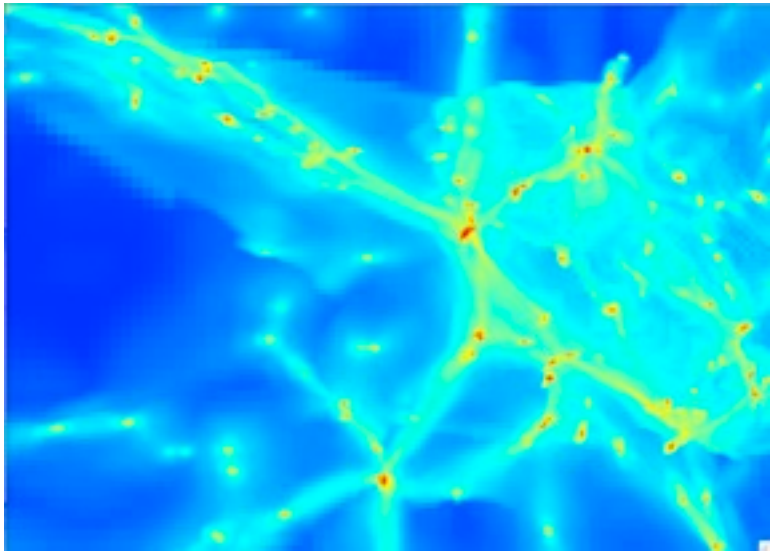
Poisson solver on the AMR grid
Multigrid or Conjugate Gradient
Interpolation to get Dirichlet boundary conditions (one way interface)

Quasi-Lagrangian mesh evolution:
roughly constant number of particles per cell

$$n = \frac{\rho_{DM}}{m_{DM}} + \frac{\rho_{gas}}{m_{gas}} + \frac{\rho_*}{m_*}$$

Trigger new refinement when n > 10-40 particles. The fractal dimension is close to 1.5 at large scale (filaments) and is less than 1 at small scales (clumps).





RAMSES: a parallel graded octree AMR

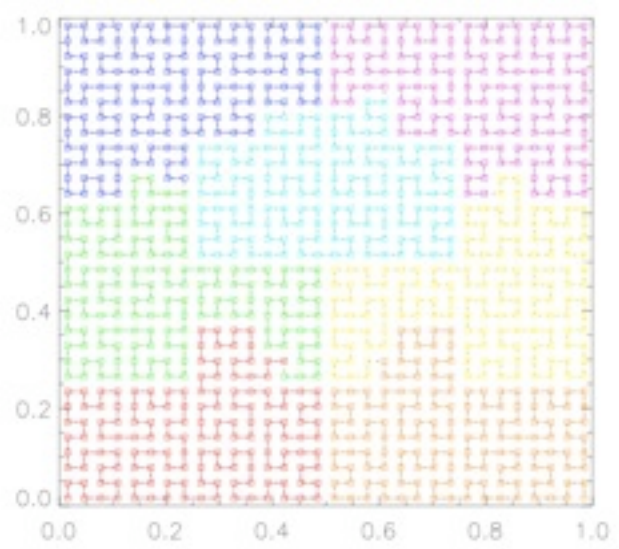
- Tree-based AMR (octree structure) : the cartesian mesh is recursively refined *on a cell by cell basis.*
- Full connectivity : each "oct" have direct access to neighboring parent cells and to children "octs". (memory overhead : 2 integers per cell).

→ Optimize the mesh adaptivity to complex geometries, but CPU overhead can be as large as 50%.

Code is freely available http://irfu.cea.fr/Projets/Site_ramses

N body module :	Particle-Mesh method on AMR grids (similar to the ART code). Poisson equation solved using Conjugate Gradient and Multigrid.
Hydro module :	*Unsplit* second order Godunov method : Riemann solver with piecewise linear reconstruction (option : MUSCL or PLMDE).
Time integration :	Single time step or W cycle (fine levels subcycling)
Other	Cooling & UV heating, Zoom simulation technology MPI based parallel implementation → *Space Filling Curves*

Domain decomposition for parallel computing



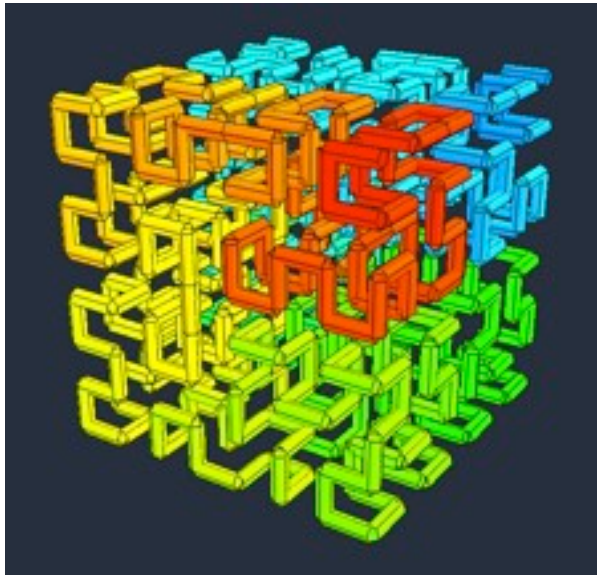
Parallel computing using the MPI library with a domain decomposition based on the *Peano-Hilbert curve*.

Algorithm inspired by gravity solvers (tree codes).
Use locally essential tree.

Tested and operational up to 20 000 cores.
Scaling depends on problem size and complexity.

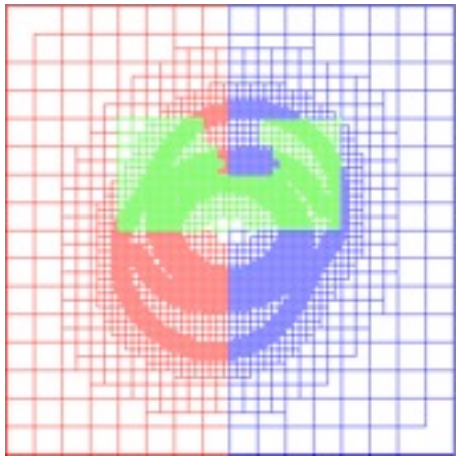
Salmon, J.K. and Warren, M.S., "Parallel out-of-core methods for N-body simulation", In Eighth SIAM Conference on Parallel Processing for Scientific Computing, SIAM,1997.

Peter MacNeice, Kevin M. Olsonb, Clark Mobarryc, Rosalinda de Fainchteind and Charles Packer, « PARAMESH: A parallel adaptive mesh refinement community toolkit, », 2000, Computer Physics Communications, 126, 3.



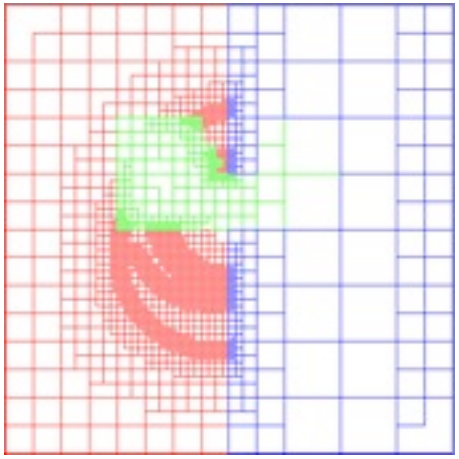
Locally essential trees

Salmon 90,
Warren 92,
Dubinski 96

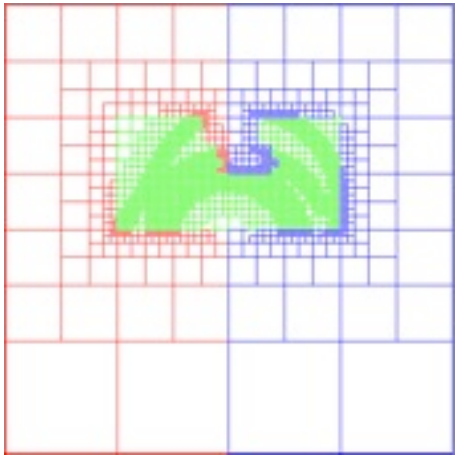


Domain decomposition over 3 processors

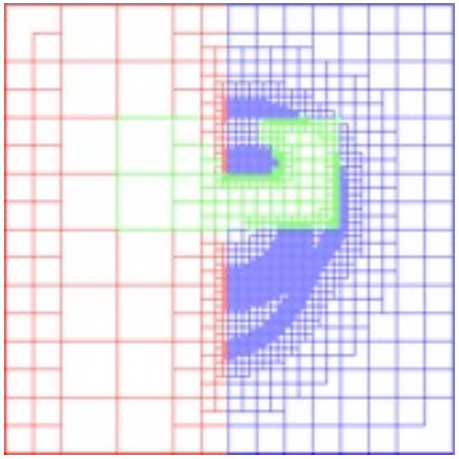
Each processor octree is surrounded by ghost cells (local copy of distant processor octrees) so that the resulting local octree contains all the necessary information.



Locally essential tree in processor #1



Locally essential tree in processor #2



Locally essential tree in processor #3

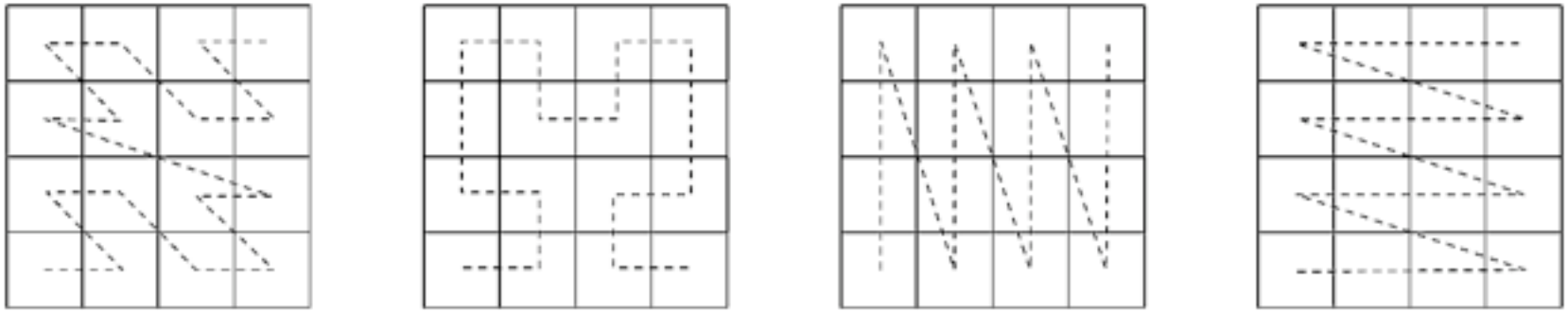
Dynamic partitioning in RAMSES

Several cell ordering methods:

1- column, row or plane major

2- Hilbert or Morton

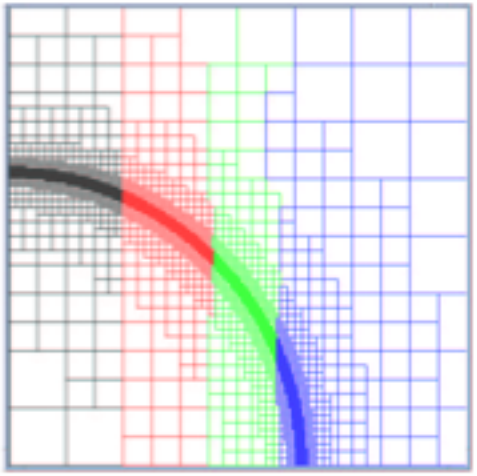
3- User defined (angular, column+Hilbert…)



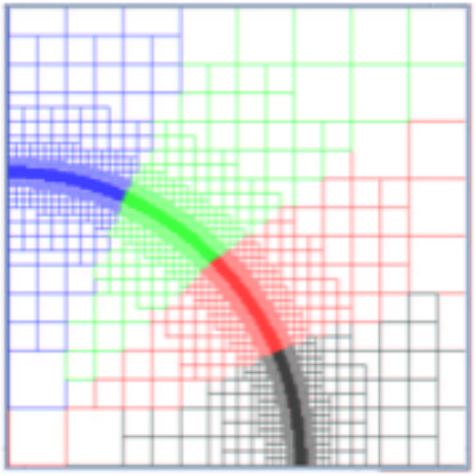
`nremap=10`

Dynamic partioning is performed every N steps by sorting each cell along chosen ordering and redistributing the mesh across processors. Usually, a good choice is N=10 (10% overhead).

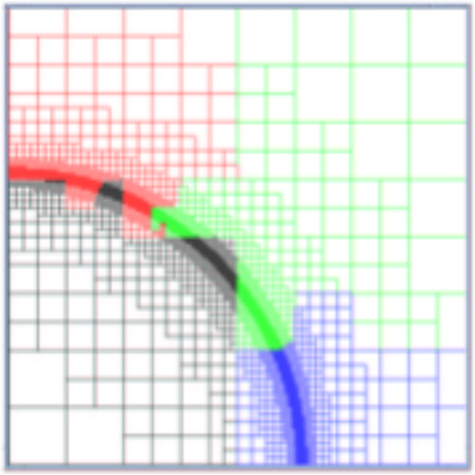
Is there an optimal load balancing strategy ?



Column major

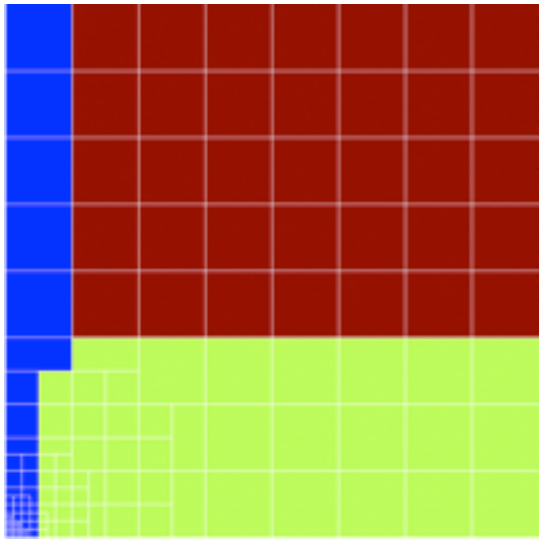


Angular

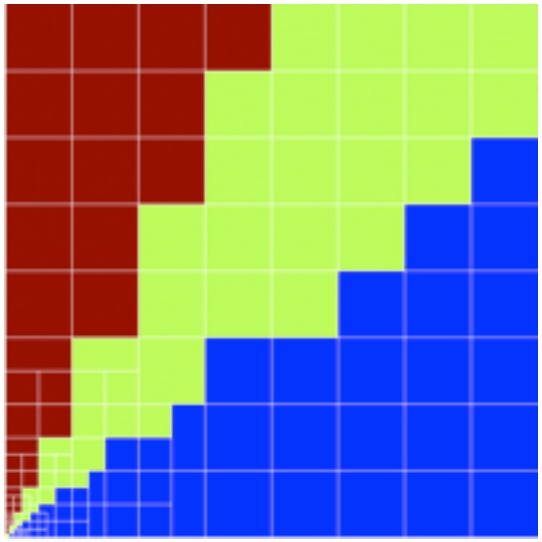


Hilbert

Is there an optimal load balancing strategy ?

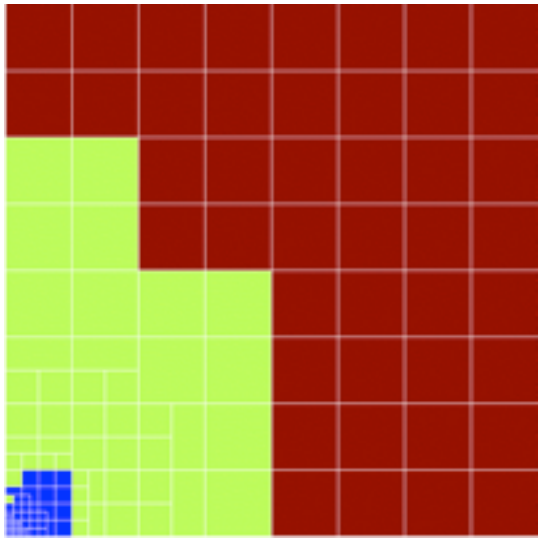


Recursive bisection



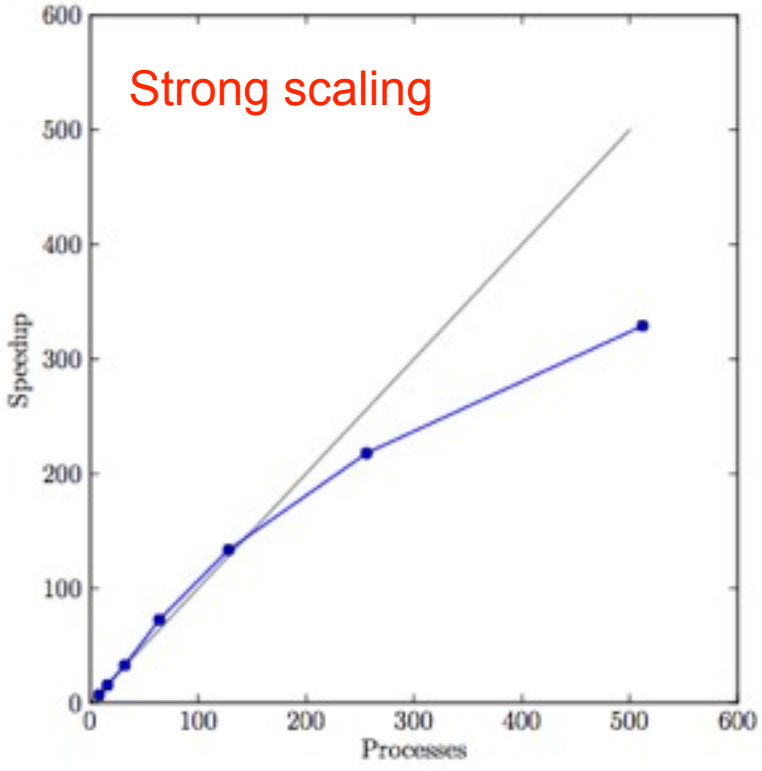
Angular ordering

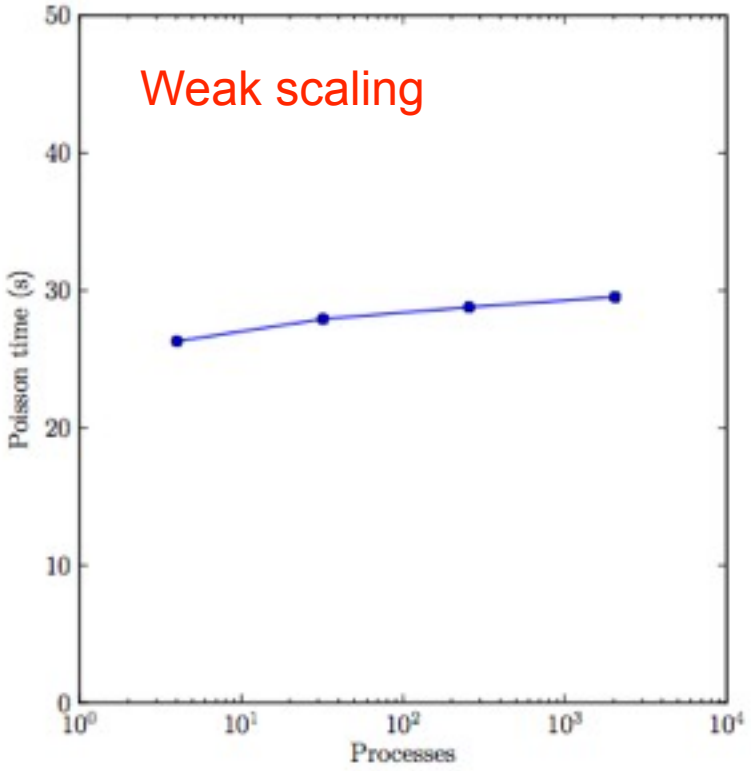
Hilbert ordering



Scaling properties depends on problem size

For a 1 million particle zoom-in simulation





Current bottleneck: adaptive time stepping.

Domain decomposition performed globally, computations performed on a level by level basis

Conclusion

- With AMR, one can minimize numerical diffusion and obtain very accurate solutions.

- AMR is efficient only if the fractal dimension of the grid is significantly smaller than 3.

- Refinement strategy is the key.

- At coarse-fine boundaries, problems arise (stationary waves).

- Parallel computing with dynamically adapted domain decomposition

- Current limitations in strong scaling are due to adaptive time stepping

- Possible solutions:

  - OpenMP-MPI hybrid programming

  - Domain decomposition per level and per species