Computational Astrophysics 5
Higher-order and AMR schemes

Romain Teyssier

Oscar Agertz

University of Zurich

Outline

- The Godunov Method
- Second-order scheme with MUSCL
- Slope limiters and TVD schemes
- Characteristics tracing and 2D slopes.
- Adaptive Mesh Refinement

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:

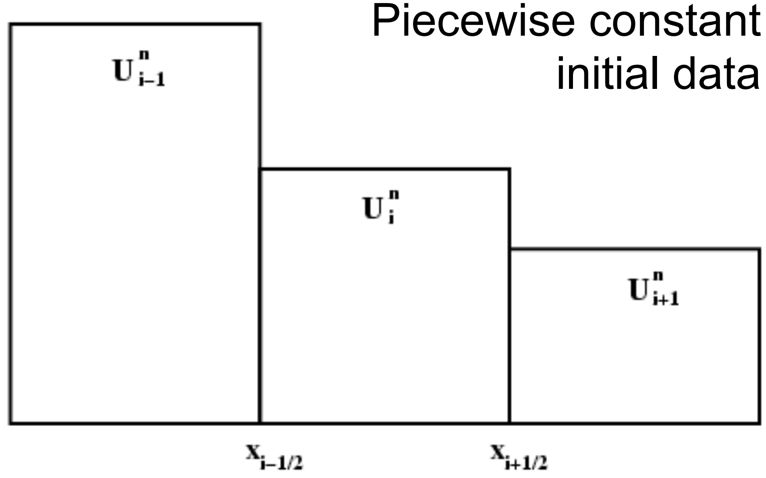
$$\mathbf{U}^*_{i+1/2}(x/t) = \mathcal{RP}\left[\mathbf{U}_i^n, \mathbf{U}_{i+1}^n\right]$$

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

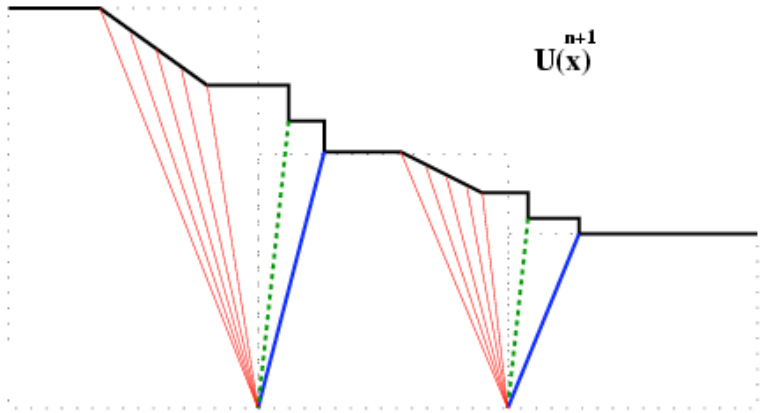
This defines the Godunov flux:

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

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

Higher Order Godunov schemes

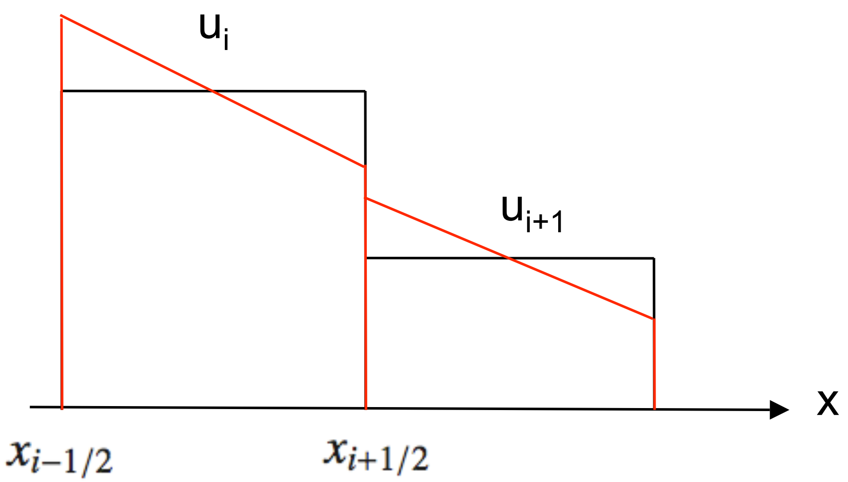
Godunov method is stable but very diffusive. It was abandoned for two decades, until…

Bram Van Leer

- **van Leer, B.** (1979), Towards the Ultimate Conservative Difference Scheme, V. A Second Order Sequel to Godunov's Method, *J. Com. Phys.*., *32*, 101–136.

Second Order Godunov scheme

Piecewise linear
approximation of the solution:



The linear profile introduces a length scale: the Riemann solution is not self-similar anymore:

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

The flux function is approximated using a *predictor-corrector* scheme:

$$\mathbf{F}^{n+1/2}_{i+1/2} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \mathbf{F}(x_{i+1/2}, t)\,\mathrm{d}t \quad \Longrightarrow \quad \mathbf{F}^{n+1/2}_{i+1/2} \simeq \mathbf{F}(\mathbf{U}^*_{i+1/2}(\frac{\Delta t}{2}))$$

The *corrected* Riemann solver has now *predicted* states as initial data:

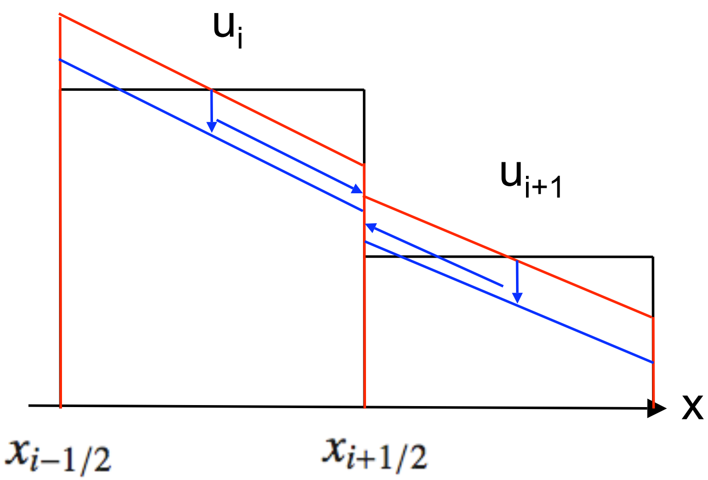
$$\mathbf{U}^*_{i+1/2}(x/t) = \mathcal{RP}\left[\mathbf{U}^{n+1/2}_{i+1/2,L}, \mathbf{U}^{n+1/2}_{i+1,R}\right]$$

Predictor Step for the advection equation

The predicted states are computed using a Taylor expansion in space and time:

$$u^{n+1/2}_{i+1/2,L} = u^n_i + \frac{\Delta t}{2}\left(\frac{\partial u}{\partial t}\right)_i + \frac{\Delta x}{2}\left(\frac{\partial u}{\partial x}\right)_i$$

$$u^{n+1/2}_{i+1/2,R} = u^n_{i+1} + \frac{\Delta t}{2}\left(\frac{\partial u}{\partial t}\right)_{i+1} - \frac{\Delta x}{2}\left(\frac{\partial u}{\partial x}\right)_{i+1}$$



Second order predicted states are the new initial conditions for the Riemann solver:

$$u^{n+1/2}_{i+1/2,L} = u^n_i + (1-C)\frac{\Delta x}{2}\left(\frac{\partial u}{\partial x}\right)_i \qquad u^{n+1/2}_{i+1/2,R} = u^n_{i+1} - (1+C)\frac{\Delta x}{2}\left(\frac{\partial u}{\partial x}\right)_{i+1}$$

The *corrected* flux function is the *upwind* predicted state:

$$f^{n+1/2}_{i+1/2} = a u^{n+1/2}_{i+1/2,L} \quad \text{if} \quad a > 0 \qquad f^{n+1/2}_{i+1/2} = a u^{n+1/2}_{i+1/2,R} \quad \text{if} \quad a < 0$$

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.

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

Summary: the MUSCL scheme for systems

Compute second order predicted states using a Taylor expansion:

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

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

Update conservative variables using corrected Godunov fluxes

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

Monotonicity preserving schemes

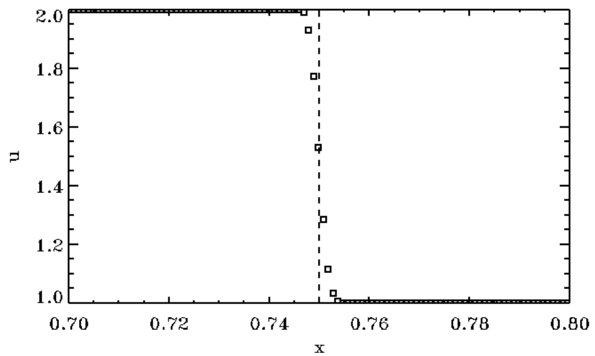
We use the central finite difference approximation for the slope:

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

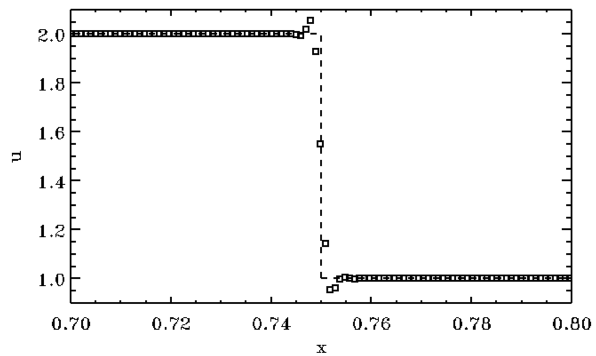
Second order linear scheme.

In this case, the solution is oscillatory, and therefore non physical.

first order



second order



Oscillations are due to the *non monotonicity* of the numerical scheme.

A scheme is monotonicity preserving if:

- No new local extrema are created in the solution

- Local minimum (maximum) non decreasing (increasing) function of time.

Godunov theorem: only first order linear schemes are monotonicity preserving !

Slope limiters

Harten introduced the Total Variation of the numerical solution:

$$TV^n = \sum_i^n |u_{i+1} - u_i|$$

Harten's theorem: a Total Variation Diminishing (TVD) scheme is monotonicity preserving.

$$TV^{n+1} \le TV^n$$

Design non-linear TVD second order scheme using slope limiters:

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{\Delta u_i}{\Delta x} = \lim(u_{i-1}, u_i, u_{i+1})\left(\frac{u_{i+1} - u_{i-1}}{2}\right)$$

where the slope limiter is a non-linear function satisfying:

$$0 \le \lim(u_{i-1}, u_i, u_{i+1}) \le 1$$

- Harten, Ami (1983), "High resolution schemes for hyperbolic conservation laws", *J. Comput. Phys* **49**: 357-393, doi:10.1006/jcph.1997.5713

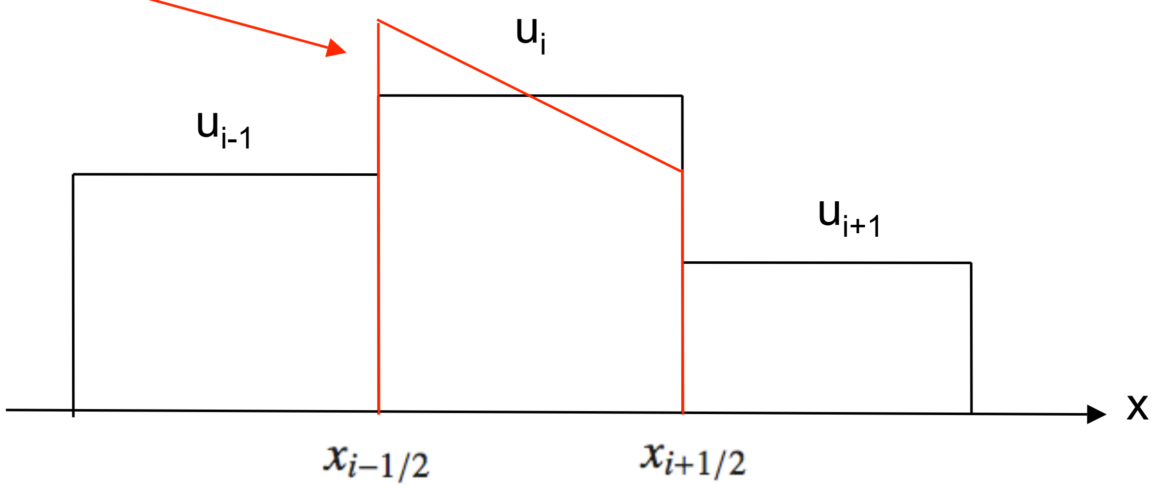
No local extrema

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{\Delta u_i}{\Delta x} \quad = \lim(u_{i-1}, u_i, u_{i+1})\left(\frac{u_{i+1} - u_{i-1}}{2}\right)$$

We define 3 local slopes: left, right and central slopes

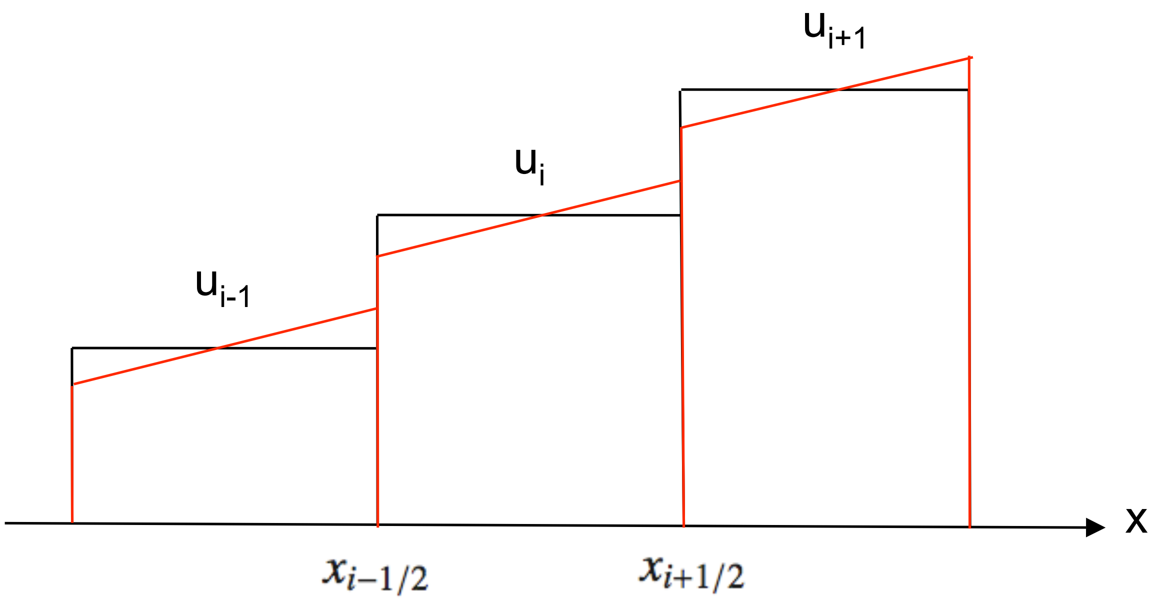
$\Delta u_L = u_i - u_{i-1} \quad \Delta u_R = u_{i+1} - u_i$ and $\Delta u_C = \dfrac{u_{i+1} - u_{i-1}}{2}$

New maximum !



For all slope limiters: $\Delta u_i = 0 \quad \text{if} \quad \Delta u_L \Delta u_R < 0$

The *minmod* slope



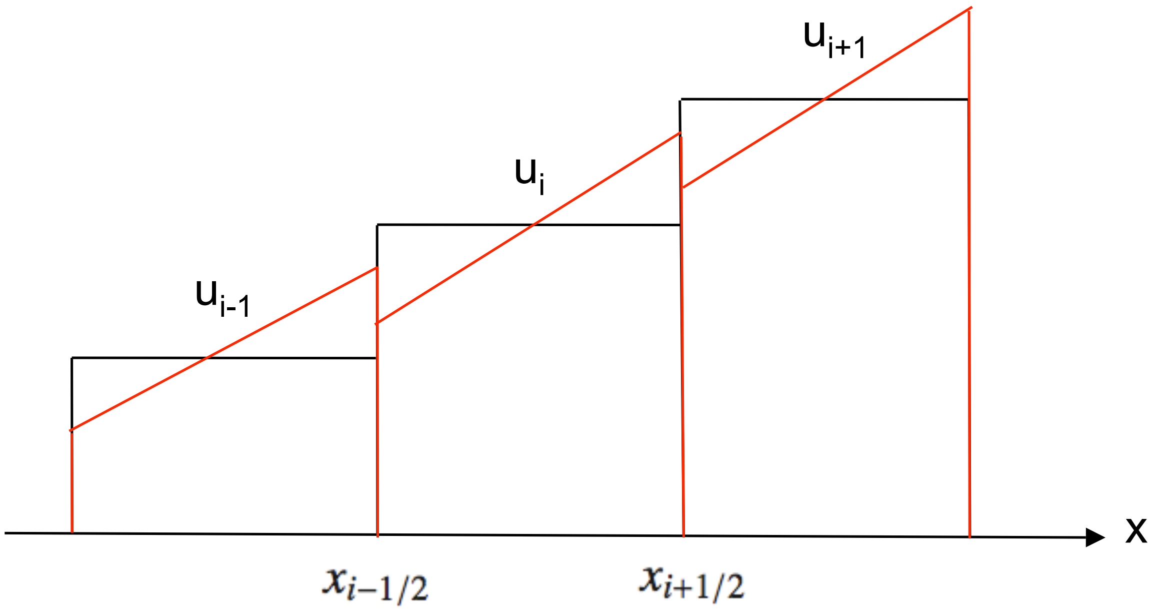
Linear reconstruction is monotone at time t^n

$$u^n_{i+1/2,L} = u^n_i + \frac{\Delta u_i}{2} \qquad u^n_{i-1/2,R} = u^n_i - \frac{\Delta u_i}{2}$$

Minmod slope limiting is never truly second order !

$$u^n_{i+1/2,L} \leq u^n_{i+1/2,R} \qquad \Delta u_i = \min(\Delta u_L, \Delta u_R)$$

The *moncen* slope



Extreme values must be bounded by the *initial average* states.

$$u^n_{i-1/2,R} = u^n_i - \frac{\Delta u_i}{2} \qquad\qquad u^n_{i+1/2,L} = u^n_i + \frac{\Delta u_i}{2}$$

$$u^n_{i-1} \le u^n_{i-1/2,R} \le u^n_i \qquad\qquad u^n_i \le u^n_{i+1/2,L} \le u^n_{i+1}$$

$$\Delta u_i = \min(2\Delta u_L, \Delta u_C, 2\Delta u_R)$$

The *superbee* slope

Predicted states must be bounded by the initial average states.

$$u_{i+1/2,L}^{n+1/2} = u_i^n + (1 - C)\frac{\Delta u_i}{2}$$

$$u_{i+1/2,R}^{n+1/2} = u_{i+1}^n - (1 + C)\frac{\Delta u_{i+1}}{2}$$

TVD constraint is preserved by the Riemann solver.

$$u_i^n \leq u_{i+1/2,L}^{n+1/2} \leq u_{i+1}^n$$

$$u_{i-1}^n \leq u_{i-1/2,R}^{n+1/2} \leq u_i^n$$

The Courant factor now enters the slope definition.

$$\Delta u_i = \min(\frac{2}{1+C}\Delta u_L, \frac{2}{1-C}\Delta u_R)$$

The ultrabee slope

Use the final state to compute the slope limiter.

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

Upwind Total Variation constraint.

$$u_{i-1}^n \leq u_i^{n+1} \leq u_i^n$$

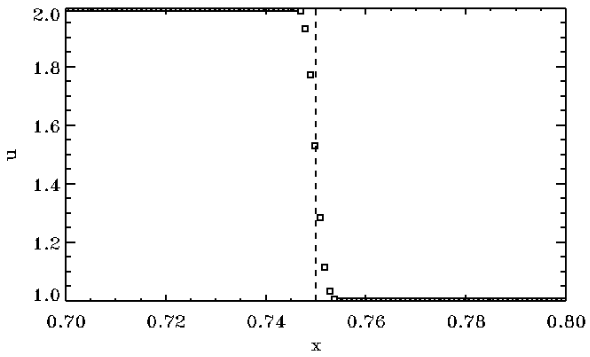
Strict Total Variation preserving limiter.

$$\text{if} \quad C > 0 \qquad \Delta u_i = \min\left(\frac{2}{C}\Delta u_L, \frac{2}{1-C}\Delta u_R\right)$$

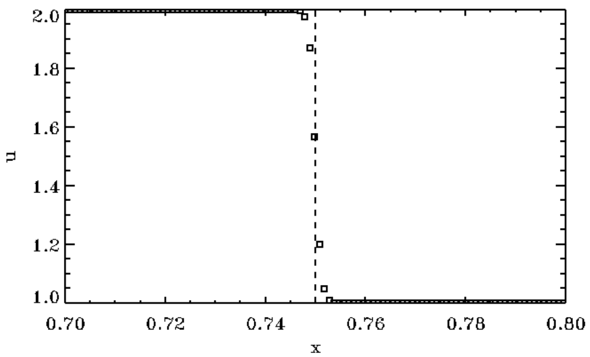
$$\text{if} \quad C < 0 \qquad \Delta u_i = \min\left(\frac{2}{1+C}\Delta u_L, \frac{2}{-C}\Delta u_R\right)$$

Summary: slope limiters

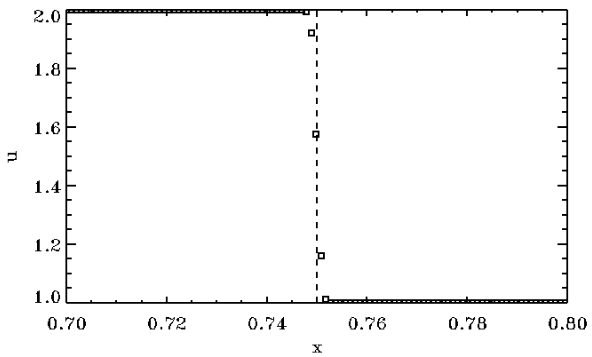
first order



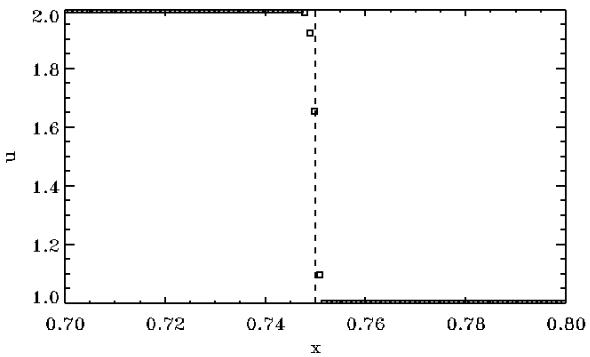
minmod



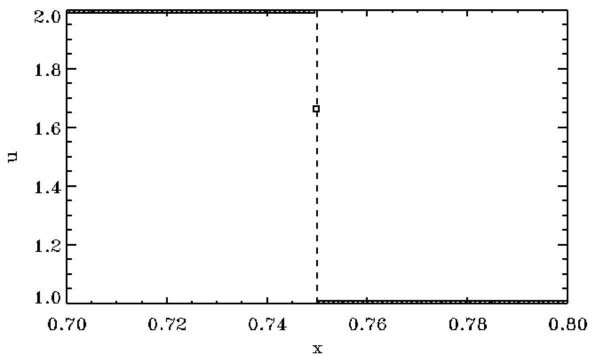
moncen



superbee



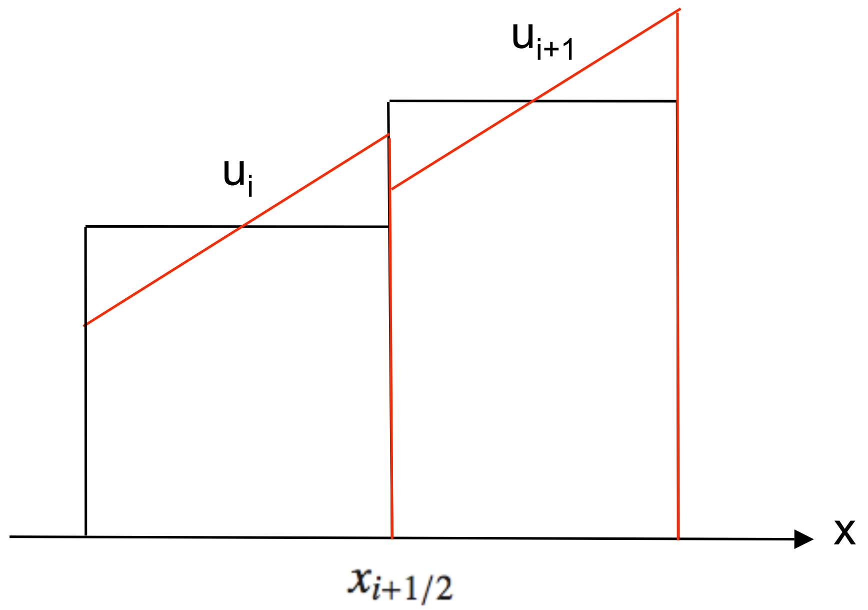
ultrabee



Summary: slope limiters

The previous analysis is valid only for the advection equation.

Non-linear systems: the wave speeds depend on the initial states (L and R).

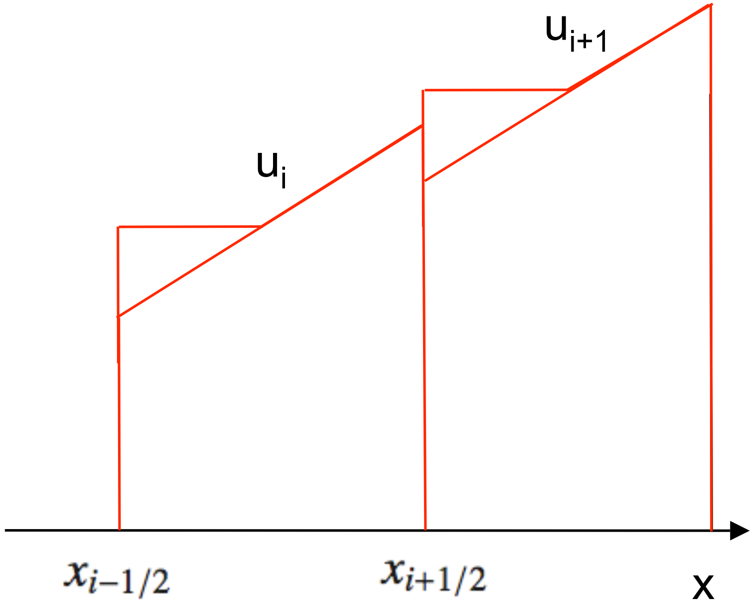


MinMod is the only monotone slope limiter before the Riemann solver !

Superbee and Ultrabee must not be used for non-linear systems !

MonCen can be used, but with care: the characteristics tracing method.

Non-linear systems: characteristics tracing.



Non-linear Riemann problems: waves speeds depend on the input states.

TVD schemes are not necessary monotone.

Modify the predictor step according to the local Riemann solution: Piecewise Linear Method (PLM) and Piecewise Parabolic Method (PPM).

If $(C_k)_i > 0$ $\quad (\alpha_k)^{n+1/2}_{i+1/2,L} = (\alpha_k)^n_i + (1 - (C_k)_i)\dfrac{(\Delta\alpha_k)_i}{2}$

else $\quad (\alpha_k)^{n+1/2}_{i+1/2,L} = (\alpha_k)^n_i$

$$C_- = (u - a)\frac{\Delta t}{\Delta x}$$

$$C_0 = u\frac{\Delta t}{\Delta x}$$

$$C_+ = (u + a)\frac{\Delta t}{\Delta x}$$

If $(C_k)_{i+1} < 0$ $\quad (\alpha_k)^{n+1/2}_{i+1/2,R} = (\alpha_k)^n_{i+1} - (1 + (C_k)_{i+1})\dfrac{(\Delta\alpha_k)_i}{2}$

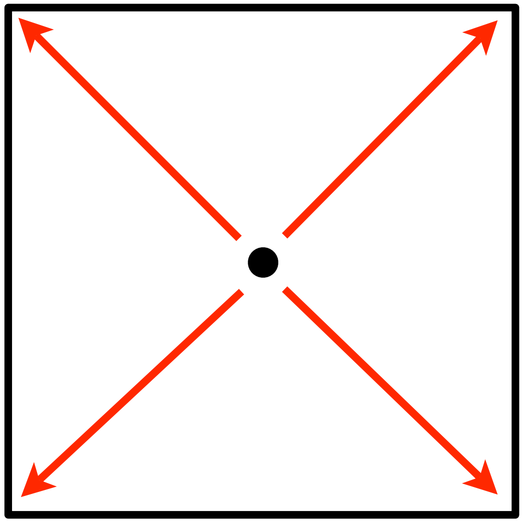
else $\quad (\alpha_k)^{n+1/2}_{i+1/2,R} = (\alpha_k)^n_{i+1}$

- Colella, P. and Woodward, P., "The Piecewise parabolic Method (PPM) for Gasdynamical Simulations", J. Comput. Phys., **54**, 174-201 (1984).

2D slope limiter for unsplit schemes

$$u^{n+1/2}_{i,j+1/2} = u^n_{i,j} - C_x\Delta_x u_{i,j} + (1 - C_y)\Delta_y u_{i,j}$$

$$u^{n+1/2}_{i+1/2,j} = u^n_{i,j} + (1 - C_x)\Delta_x u_{i,j} - C_y\Delta_y u_{i,j}$$



If 1D slope limiters are used, 2D schemes may become oscillatory.

Predicted states involve 2D neighboring cells.

2D *moncen* slope: corner values must be bounded by the 8 neighboring initial values.

Surech, Ambady, "Positivity Preserving Schemes in Multidimensions", SIAM J. Sci. Comput., **22**, 1184-1198 (2000).

Beyond second order Godunov schemes ?

Smooth regions of the flow

More efficient to go to higher order.

Spectral methods can show *exponential convergence*.

More flexible approaches: use *ultra-high-order* shock-capturing schemes: 4th order scheme, ENO, WENO, discontinuous Galerkin and discontinuous element methods

Discontinuity in the flow

More efficient to refine the mesh, since higher order schemes drop to first order.

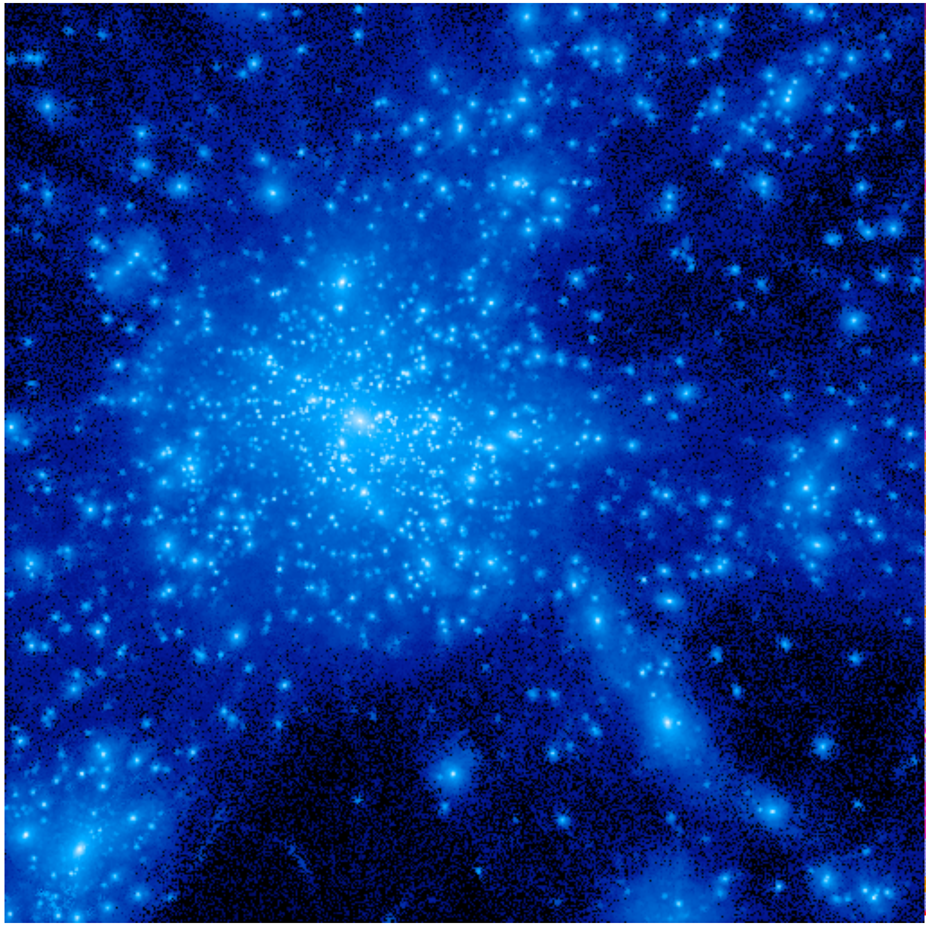
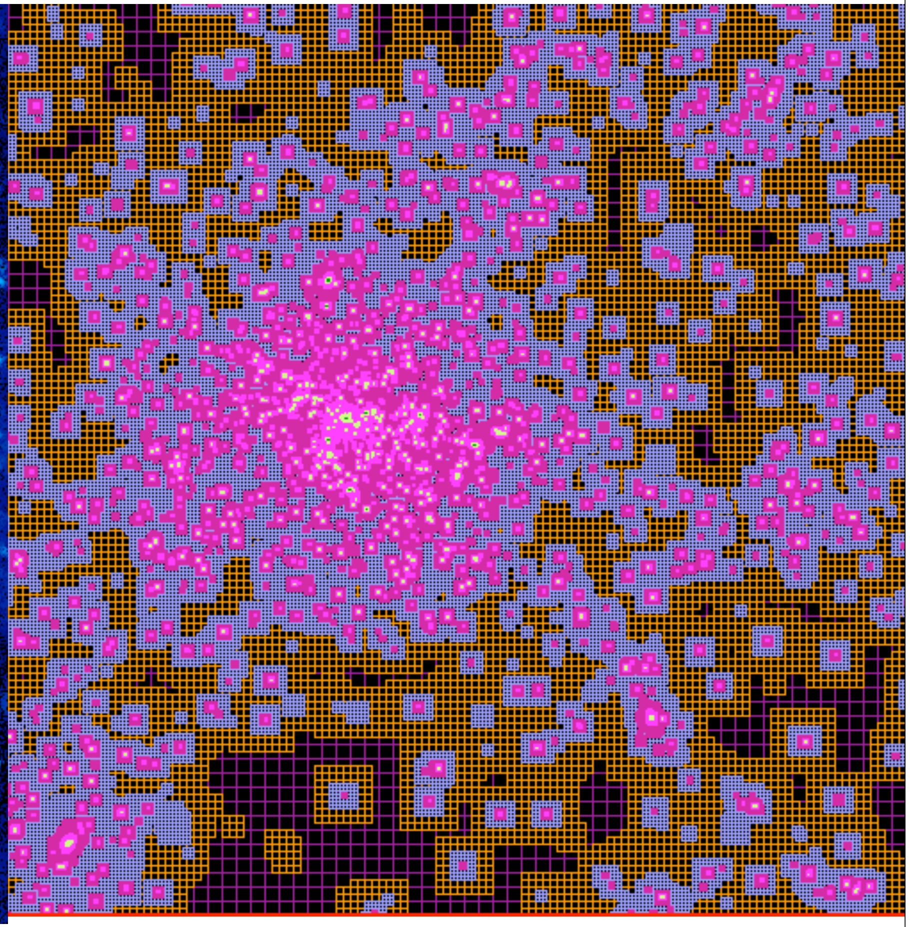
Adaptive Mesh Refinement is the most appealing approach.

What about the future ?

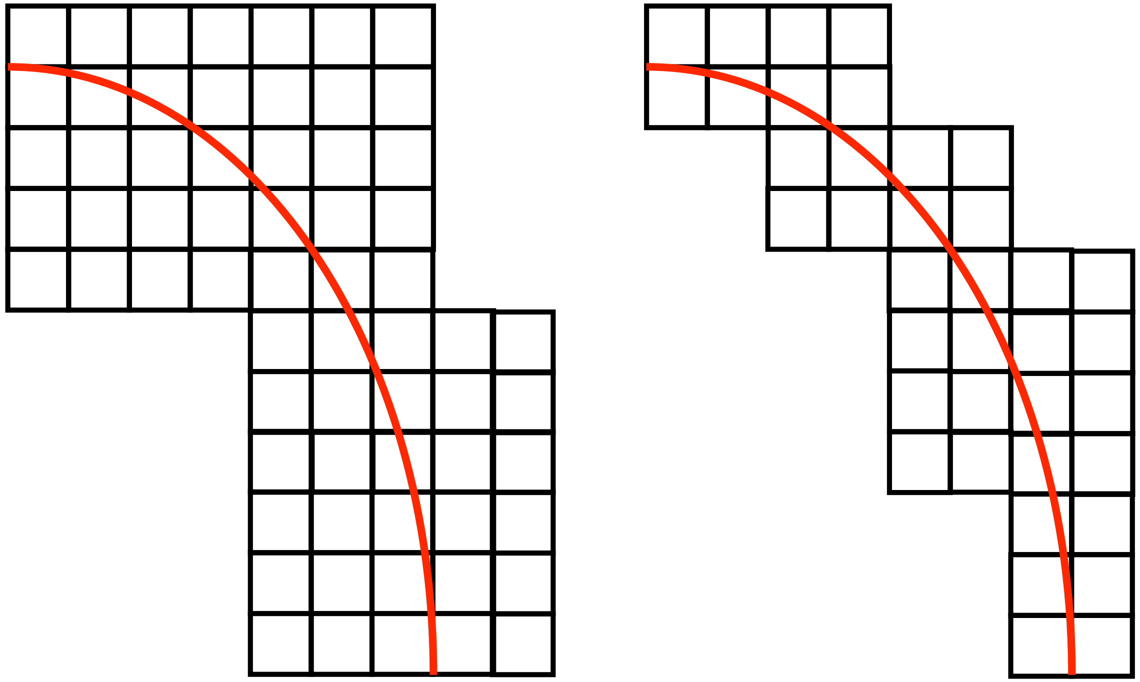
Combine the 2 approaches.

Usually referred to as “*h-p adaptivity*”.

Adaptive Mesh Refinement

Patch-based versus tree-based



A few AMR codes in astrophysics

ENZO: Greg Bryan, Michael Norman…

ART: Andrey Kravtsov, Anatoly Klypin

RAMSES: Romain Teyssier

NIRVANA: Udo Ziegler

AMRVAC: Gabor Thot and Rony Keppens

FLASH: The Flash group (PARAMESH lib)

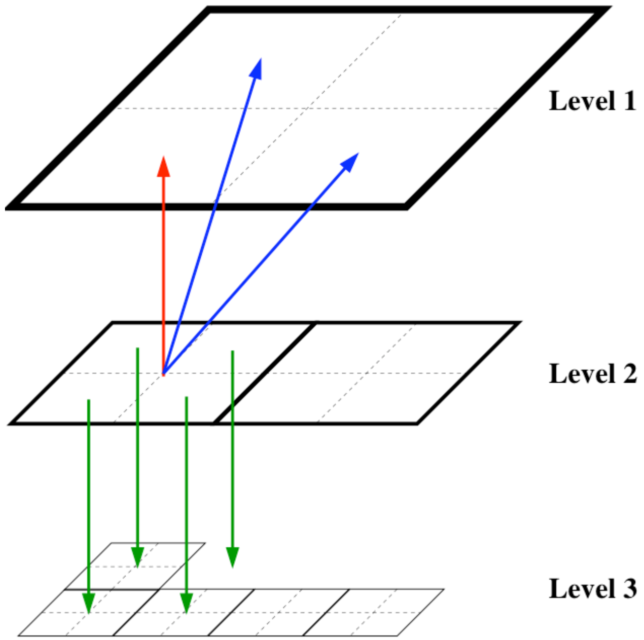
ORION: Richard Klein, Chris McKee, Phil Colella

PLUTO: Andrea Mignone (CHOMBO lib, Phil Colella)

CHARM: Francesco Miniati (CHOMBO lib, Phil Colella)

ASTROBear: Adam Frank…

Graded Octree structure



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.

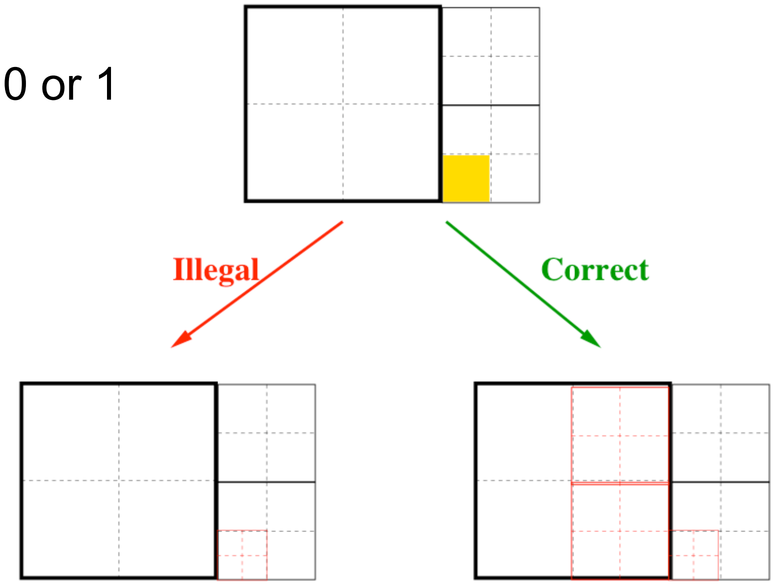
Optimize mesh adaptation to complex flow geometries, but CPU overhead compared to unigrid can be as large as 50%.

2 type of cell: - "leaf" or active cell

- "split" or inactive cell

Refinement rules for graded octree

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

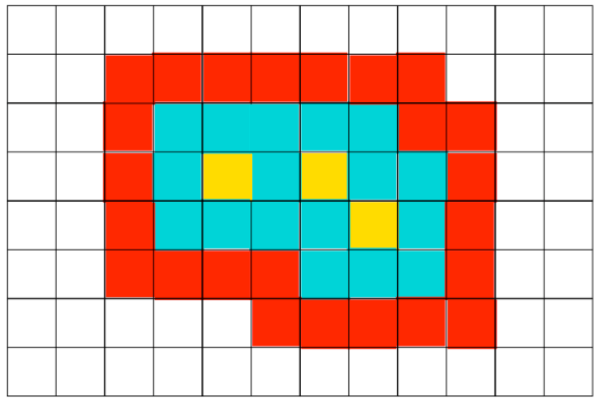
quasi-Lagrangian evolution, Jeans mass

geometrical constraints (zoom)

Truncation errors, density gradients…

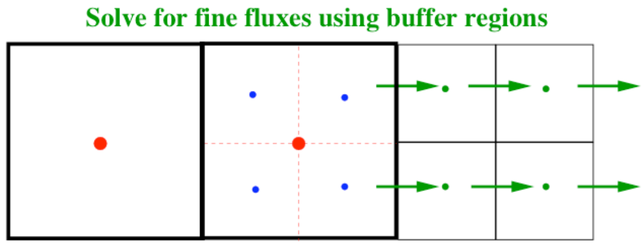
Step 3: mesh smoothing

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



Godunov schemes and AMR

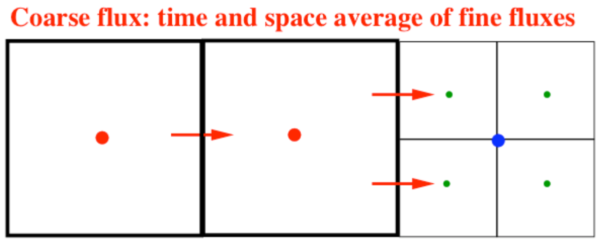
Berger & Oliger (84), Berger & Collela (89)



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

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

Careful choice of interpolation variables (conservative or not ?)

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

- straight injection

- tri-linear, tri-parabolic reconstruction

Godunov schemes and AMR

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

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

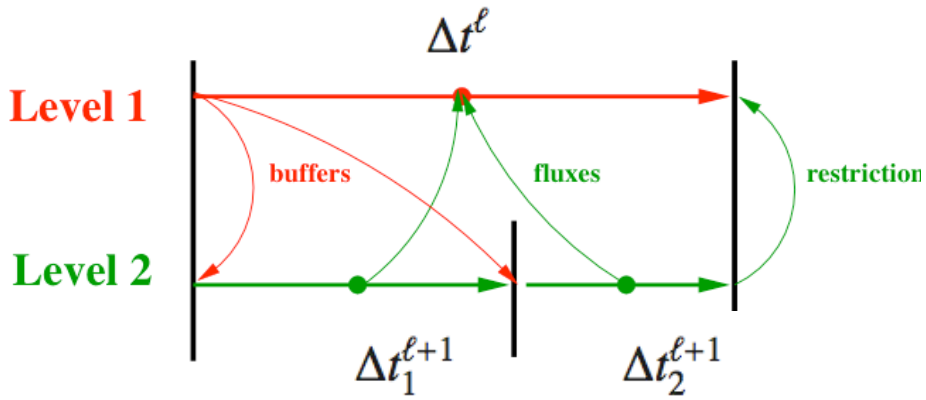
- Second order MUSCL: 2 cells in each direction

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

- 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



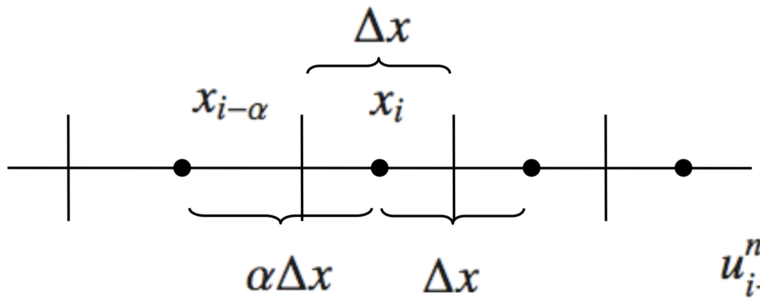
Time integration: single time step or recursive sub-cycling

- froze coarse level during fine level solves (one order of accuracy down !)

- average fluxes in time at coarse fine boundaries

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

The AMR catastrophe



Assume a and C>0.

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

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

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

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

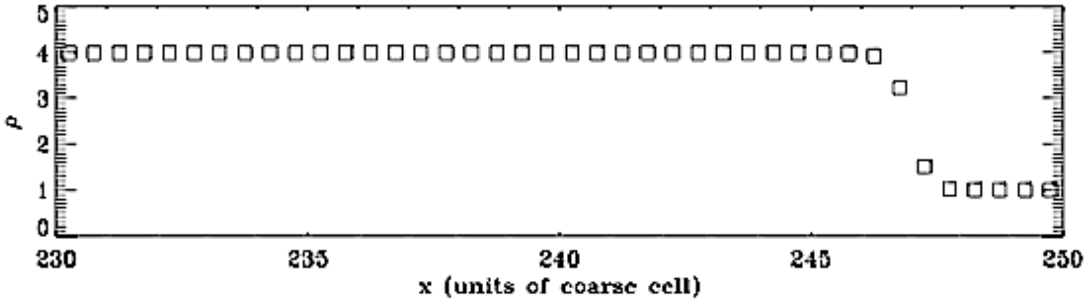
At level boundary, we loose one order of accuracy in the modified equation.

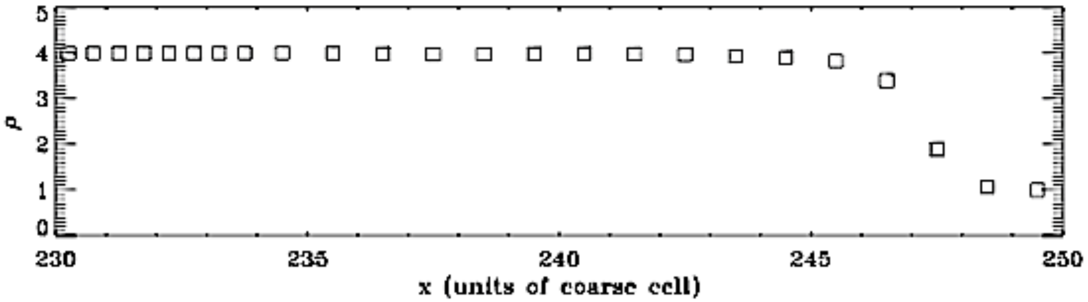
First order scheme: the AMR extension is *not consistent* at level boundary.

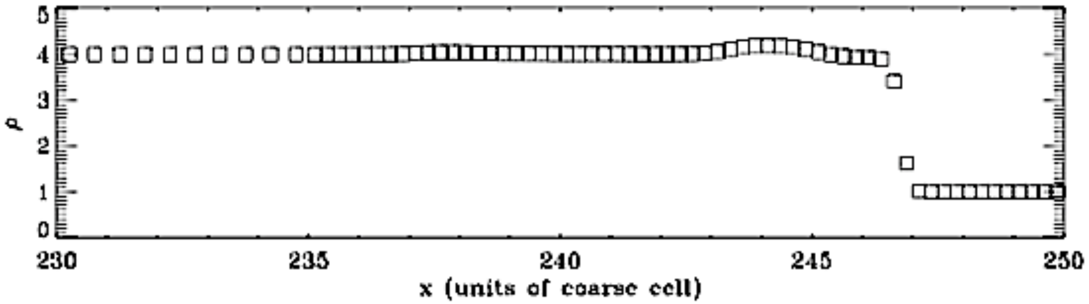
Second order scheme: for α=1.5, AMR is *unstable* at level boundary.

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

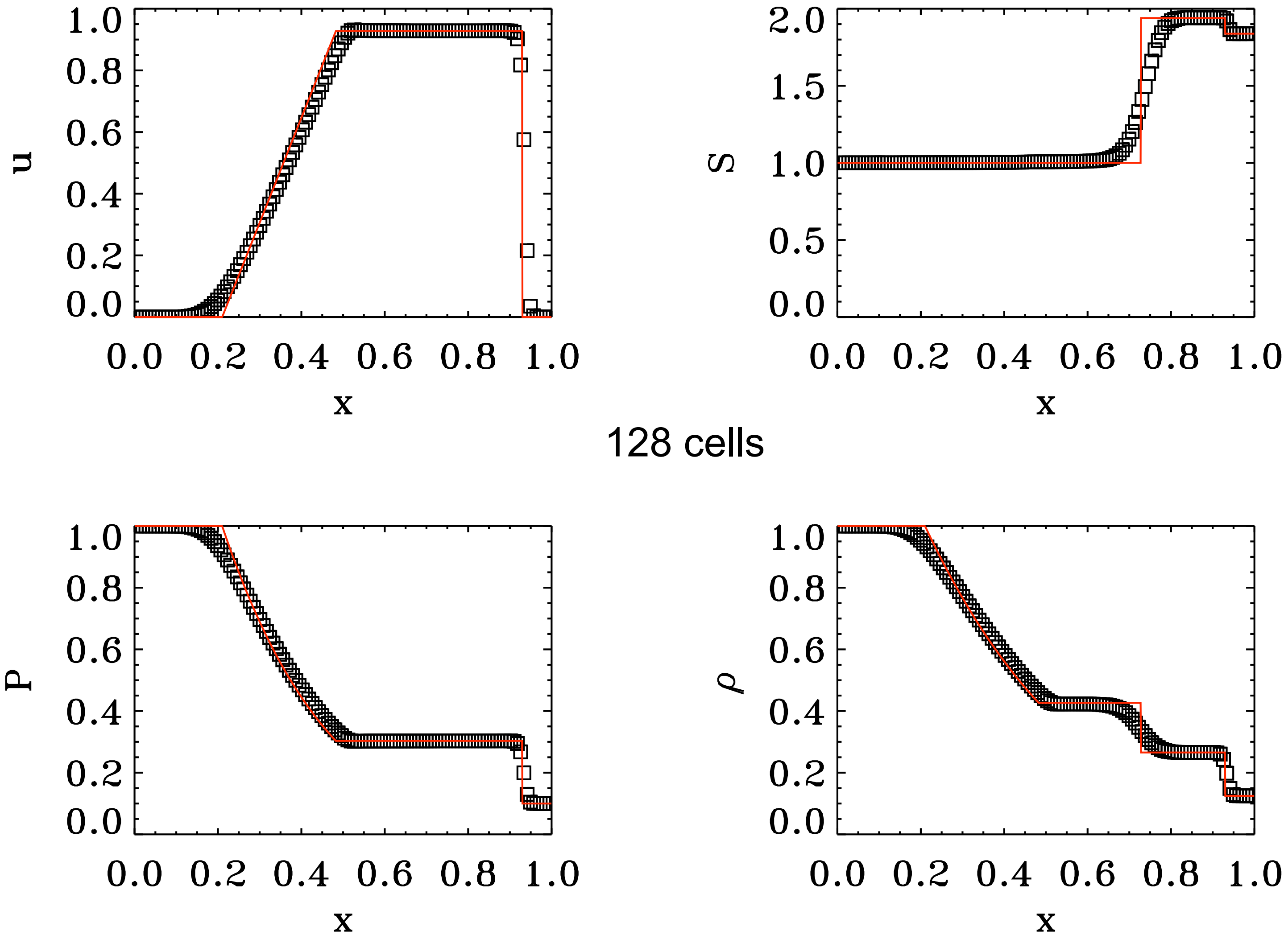
Shock wave propagating through level boundary





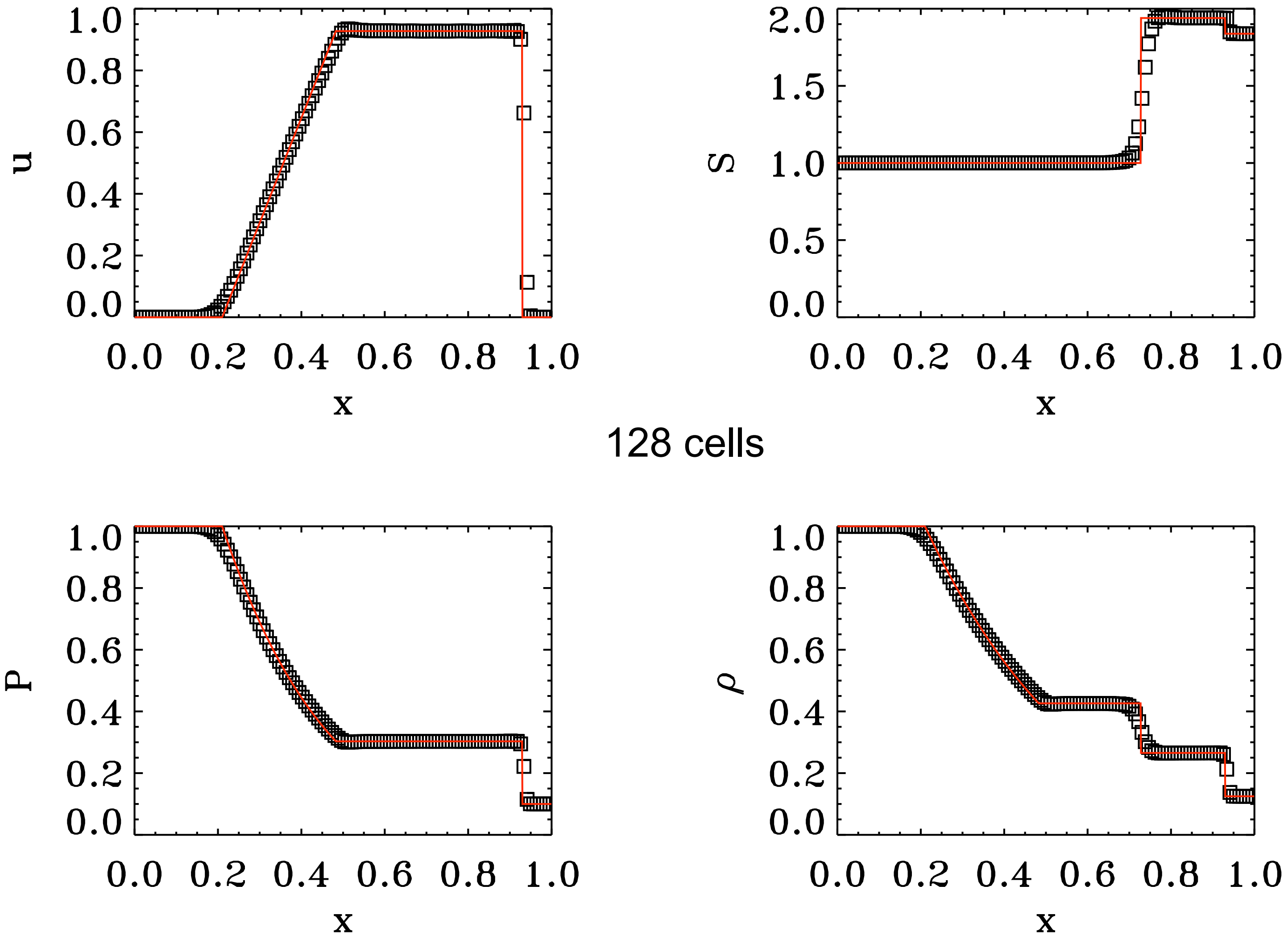


Sod test with HLLC first order

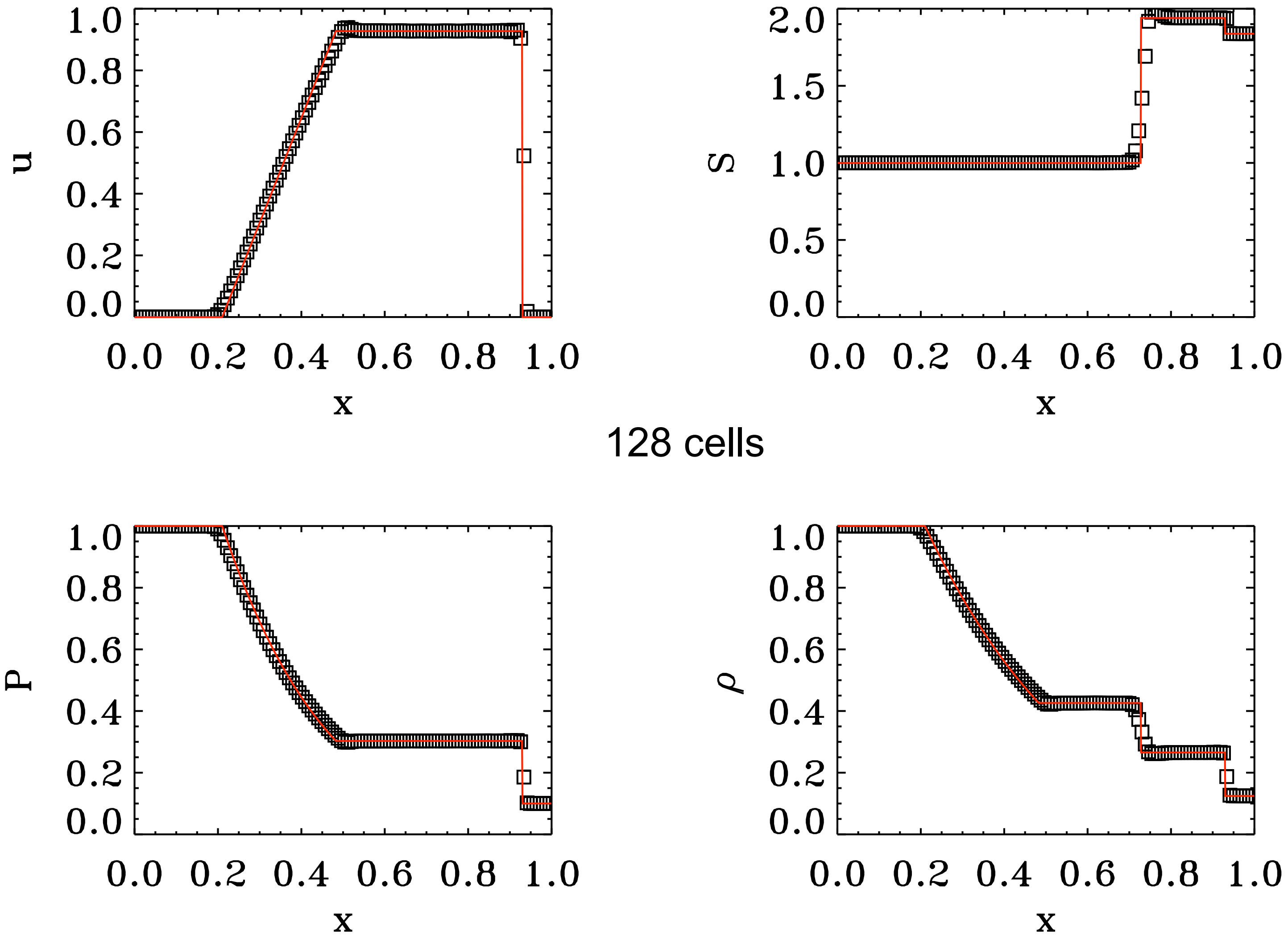


128 cells

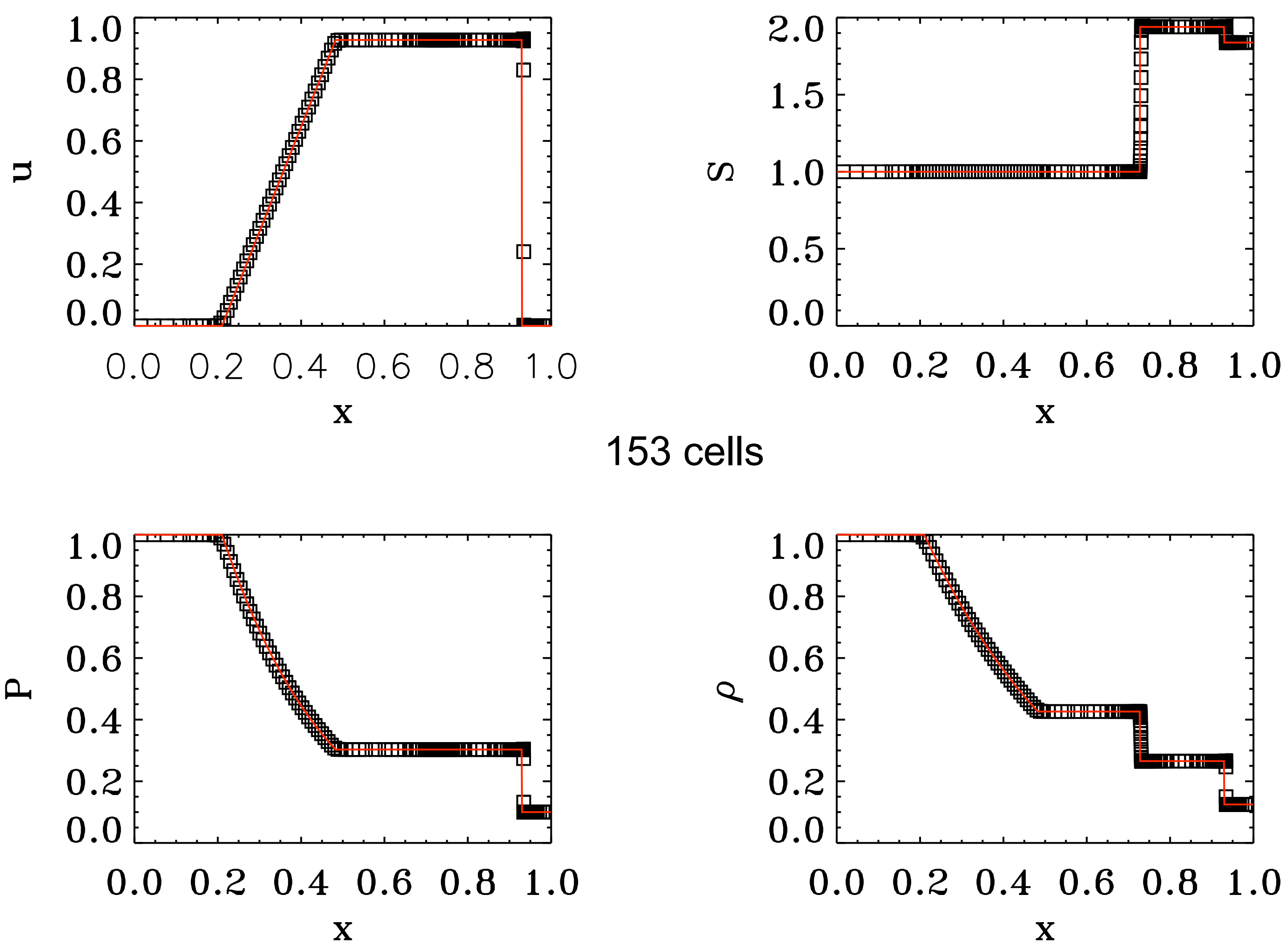
Sod test with HLLC and MinMod



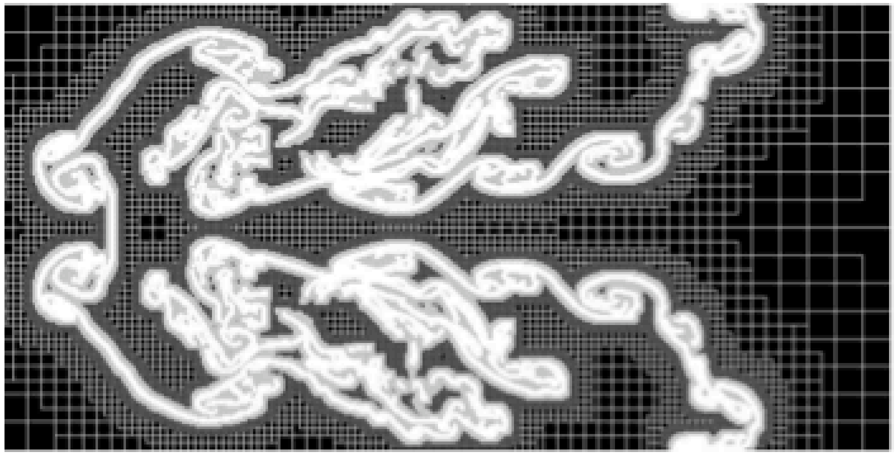
Sod test with HLLC and MonCen



Sod test with HLLC and AMR

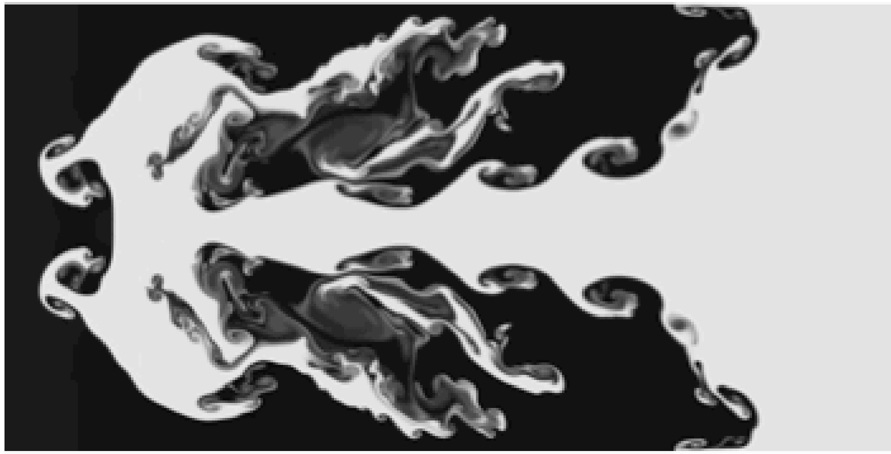


Complex geometry with AMR



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

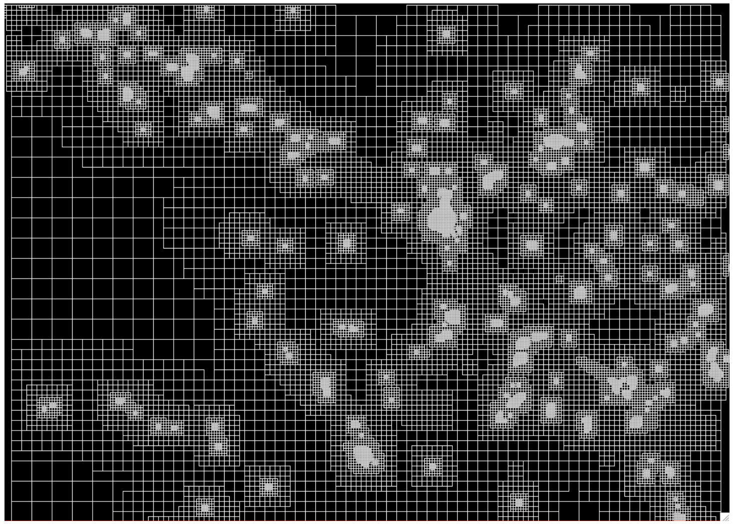
Particle-Mesh on AMR grids:
Cloud size equal to the local mesh spacing

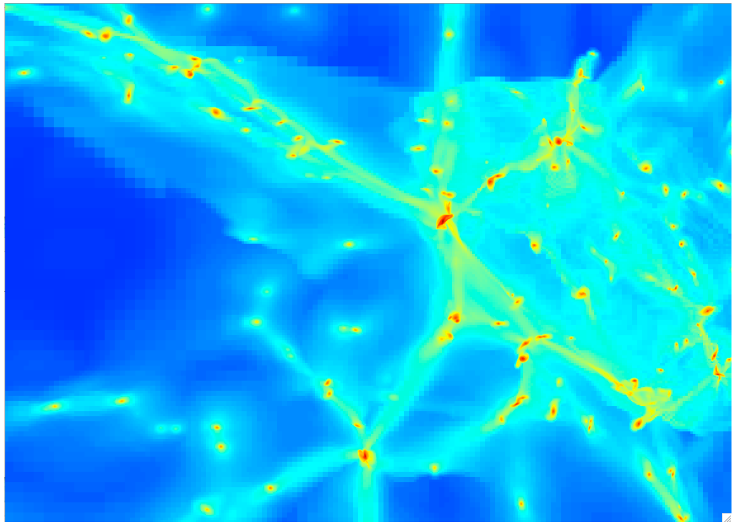
Poisson solver on the AMR grid
Multigrid or Conjugate Gradient
Interpolation to get Dirichlet boundary conditions (one way interface)

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*

Conclusion

- Second-order in space and time with predictor-corrector: MUSCL
- Dispersive error term in the Modified Equation
- MinMod and MonCen slope limiters (+ characteristics tracing ?)
- 2D slope limiting for unsplit 2D schemes
- Patch-based versus Tree-based AMR
- AMR looses one order of accuracy at level boundary
- Refinement strategy and *h/p* adaptivity ?

Next lecture: Hands on RAMSES