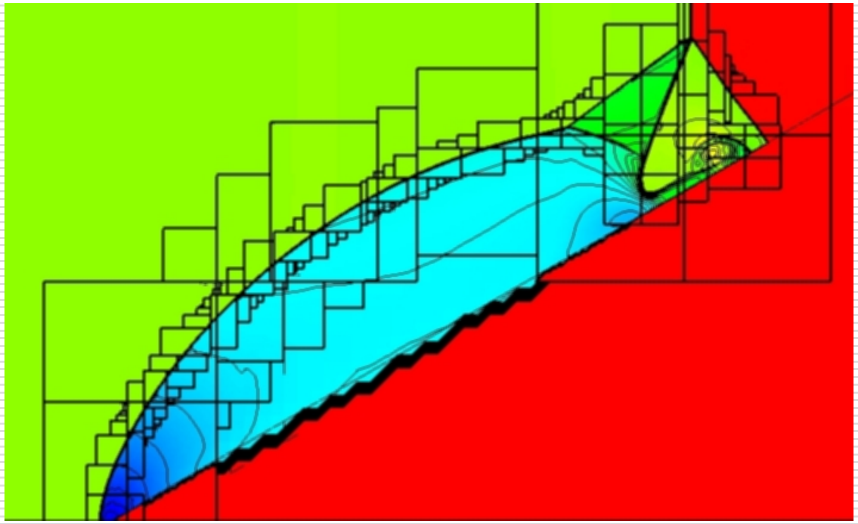
Adaptive Mesh Refinement

- Implementation
- Tests
- Applications



Why Adaptive Mesh?

- Multi dimensional solution of hyperbolic system of conservation laws often too time consuming

- Large range of spatial scale

- Often locally enhancement of resolution is sufficient

Simple Mesh: Euler equations

$$\frac{\partial}{\partial \tau}\begin{pmatrix}\rho \\ \rho V_x \\ \rho V_y \\ E_t\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho V_x \\ P + \rho V_x^2 \\ \rho V_x V_y \\ (E_t + P)V_x\end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix}\rho V_y \\ \rho V_y V_x \\ P + \rho V_y^2 \\ (E_t + P)V_y\end{pmatrix} = 0.$$

- Equation of state (here ideal gas)

$$P = (\gamma - 1)(E_t - \tfrac{1}{2}\rho[u^2 + v^2])$$

- External force (e.g. gravitation)

Simple Mesh: Euler equations

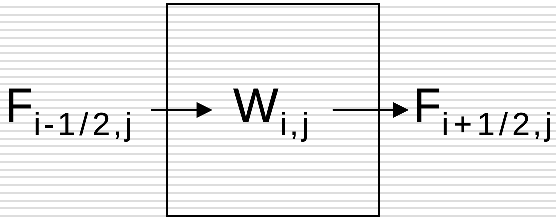
- **Conserved variables for every mesh point**
  - Mass density
  - Momentum density
  - Energy density

$$\mathbf{W} = \begin{pmatrix} \rho \\ \rho V_n \\ \rho V_t \\ E_t \end{pmatrix}$$

- **Simple approach**: finite differencing
  - Differential equations -> difference equations
  - Calculate on mesh
- **Problem**: not conserving, not monotonic conserving
- **Solution**: finite volume upwind discretization

Finite volume differencing on a simple mesh

- Fluxes



- Calculate: $$W_{i,j}(t+\Delta t) = W_{i,j}(t) + \frac{\Delta t}{\Delta x}(F_{i-1/2,j} - F_{i+1/2,j})$$

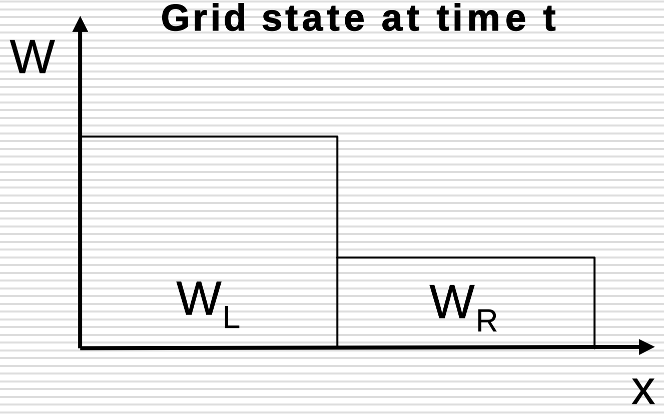
- **Problem**: find the right fluxes
  → Riemann solvers

Riemann solvers

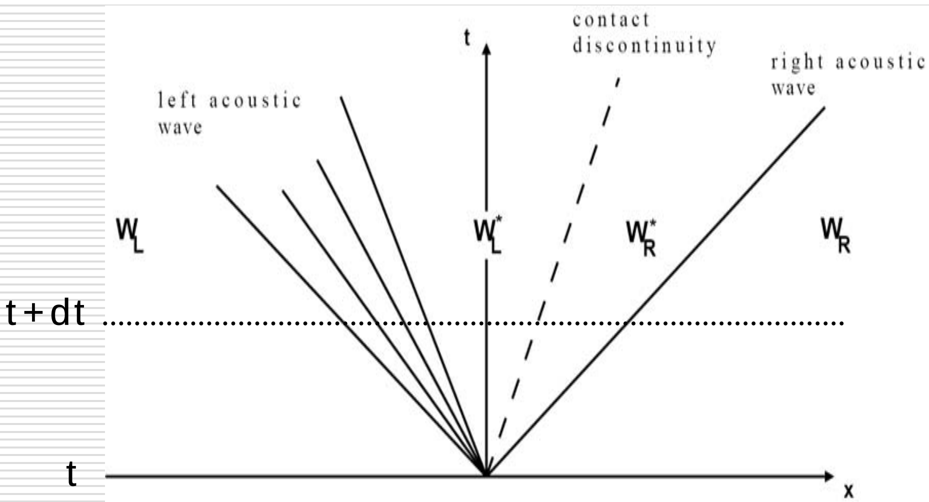
Riemann problem

- Piecewise constant function
- Intermediate states W^* at a given time
- Fluxes can be calculated exact

Grid state at time t



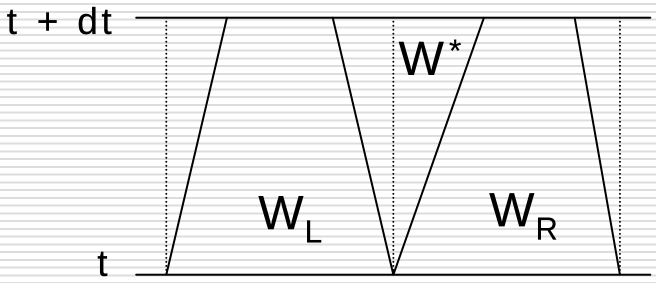
Time evolution



Riemann solvers

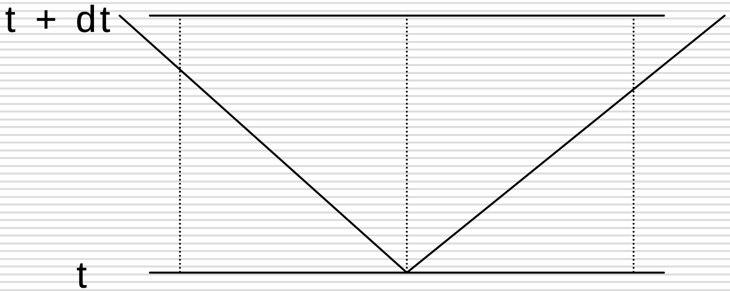
Courant condition

- Time step limit, to avoid that neighbouring Riemann problems affect each other

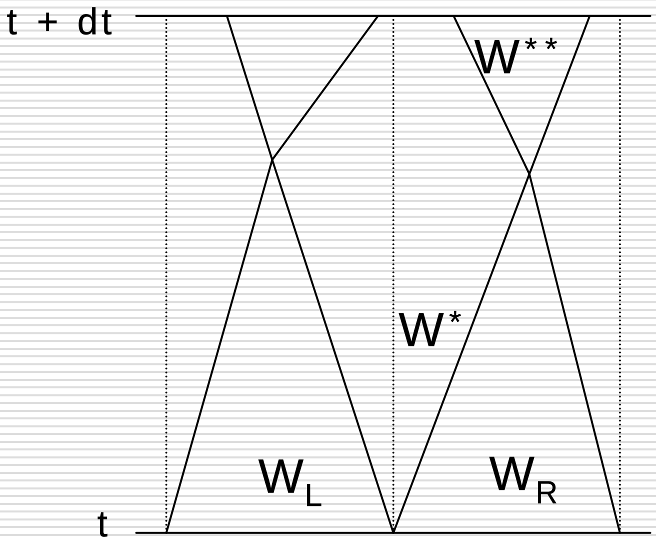


Courant number < 1/2

$$v \cdot \Delta t < \Delta x$$



Courant number > 1

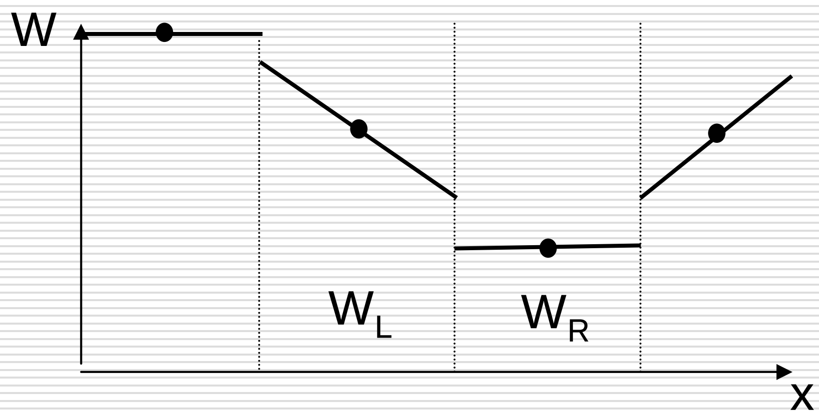


Courant number < 1

Summary of single mesh solution algorithm

For calculation: higher order schemes are used

- Interpolate a function between grid points

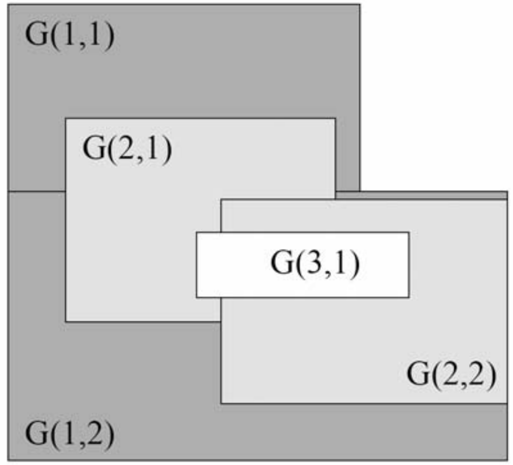


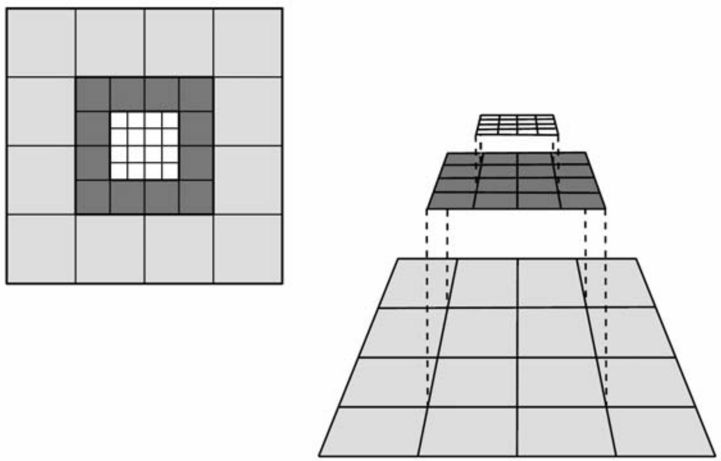
Summary

- Conservation
- Conserves shock shape and right speed
- No wiggles behind discontinuities
- Computational cost scales like
  N^3 in 2D ($N_x*N_y*N_t$)
  N^4 in 3D ($N_x*N_y*N_z*N_t$)
  → AMR might be of advantage

Adaptive Mesh

- Finer meshes overlie coarser
- Properly nested
- Refinement in space and time (e.g. factor 2)

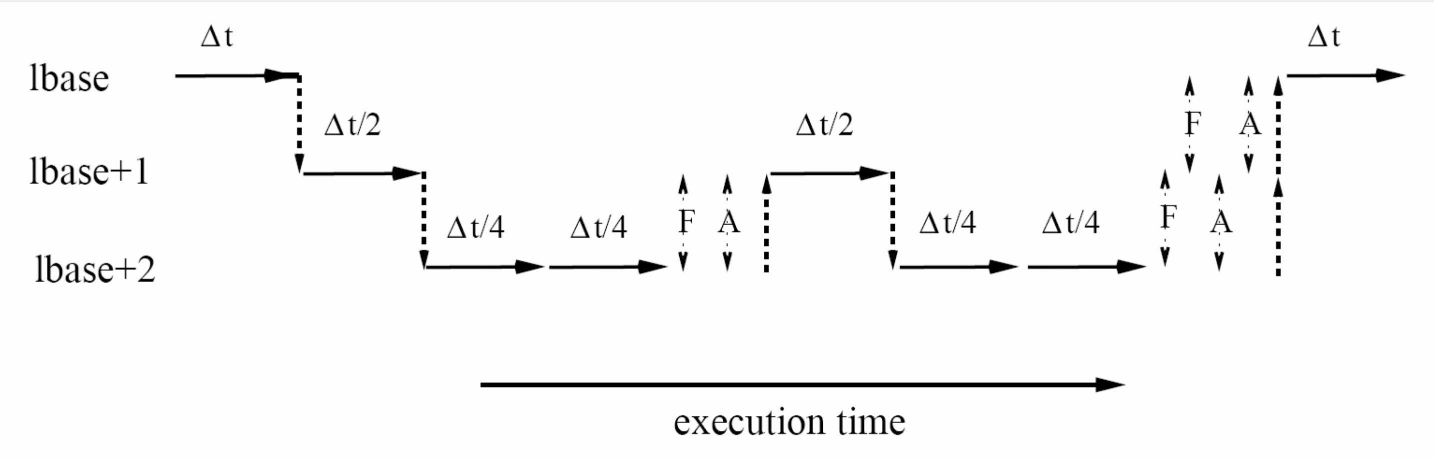




The calculation path

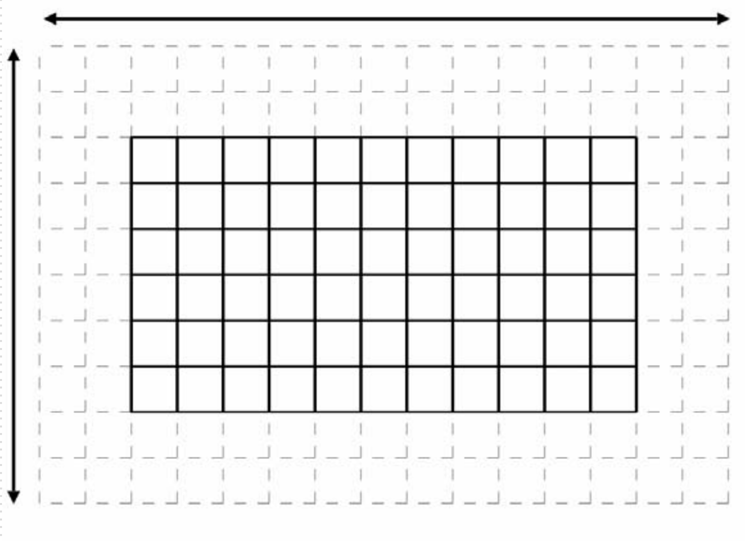
- From coarser to finer mesh
  - 1 step in coarser mesh = 2 steps in next finer mesh
  - Boundaries for finer mesh
  - Projection to the coarser mesh + fix fluxes

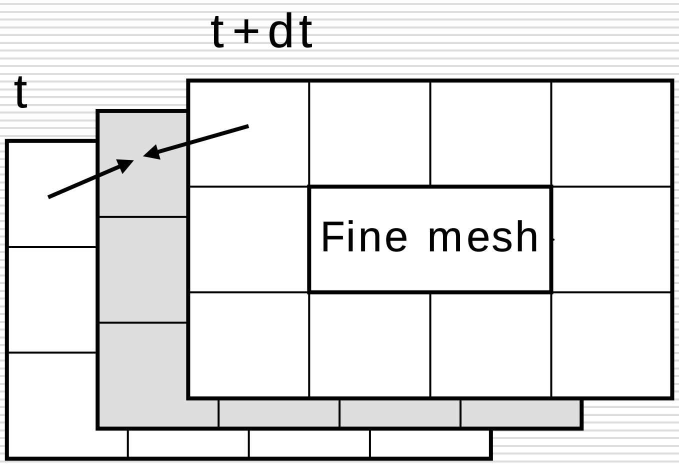
- Make it recursive!



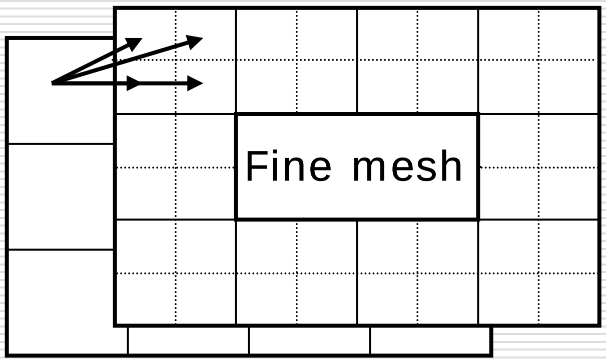
Building the boundaries for the finer mesh

- Boundaries necessary for finer meshes
- Calculate from coarser mesh
- Interpolate in space and in time
- Use same order interpolation, as in the PDE solver





Interpolation in time

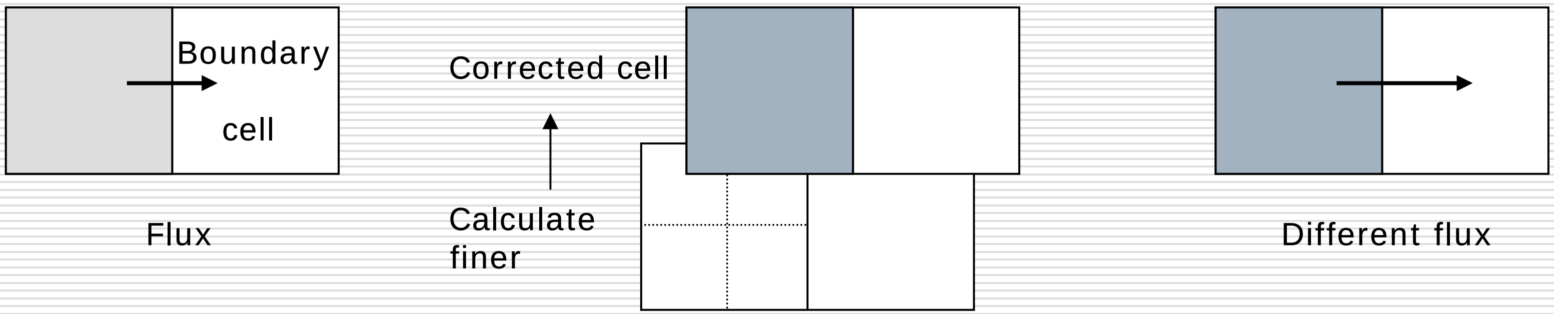


Interpolation in space

Update the cells of coarser mesh

- Build new coarse cell values

- Flux through boundary has changed because of finer calculations
- Need a fix-up for boundary cells
- Use sum over fine fluxes instead of coarse flux

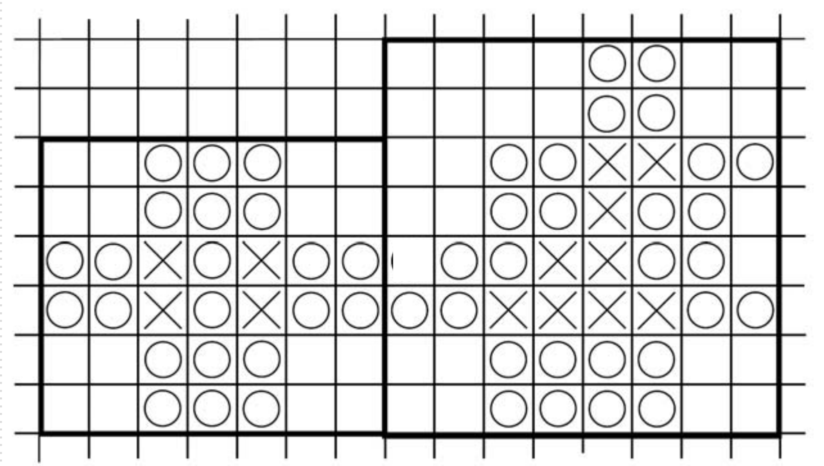


Automatic grid adaption

- Update mesh after fixed number of steps on a given level
- Mash is recreated (no “moving” grid)
- Grid generation recursively:
  from finest to coarsest grid

Steps

- Error estimation
- Flagging for refinement
- Build a buffer zone
- Grouping/clustering
- Transfer old solutions
- Or interpolate coarser mesh
  (careful! Not automatically conservative!)

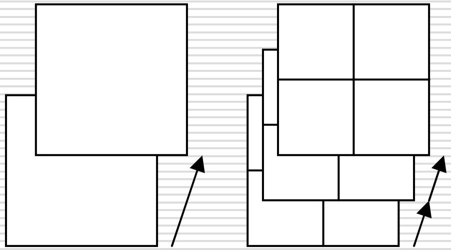


Automatic grid adaption

Flagging

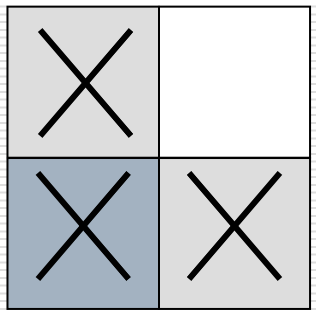
estimate potential error

- Compare finer solution with coarser solution
- Difference too high: refine!



simpler: upper limit of value change from cell to cell

- e.g. 10%
- Faster, also good results



Flagged for refinement

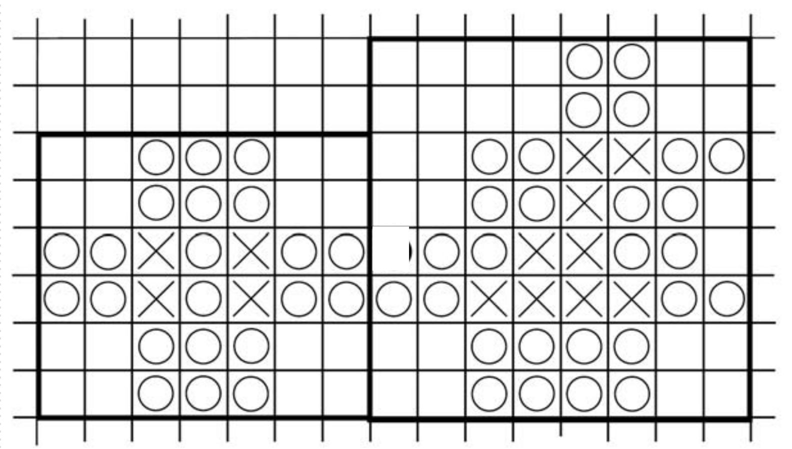
Other refinement criteria possible

Automatic grid adaption

Flagging

- Flag all cells from level l+2 in level l
  because the finest variations would not have been recognised
  (maintain proper nesting)

- Flag a buffer zone to keep
  the fine structures in the fine mesh

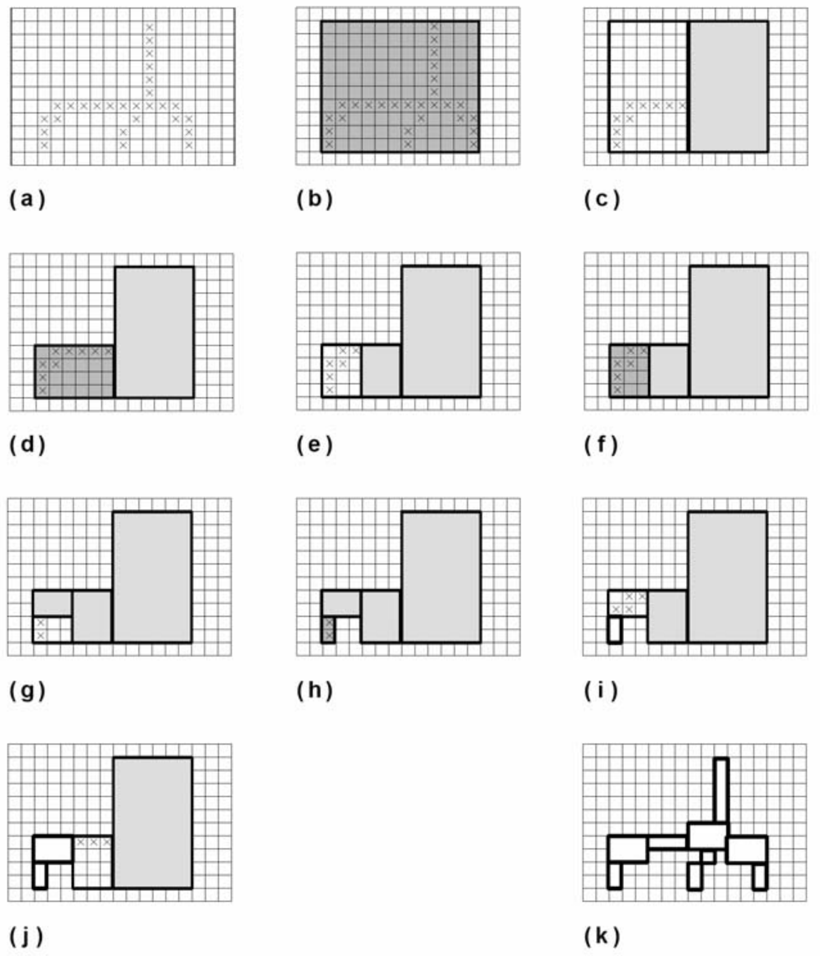


Automatic grid adaption

Creation of grids

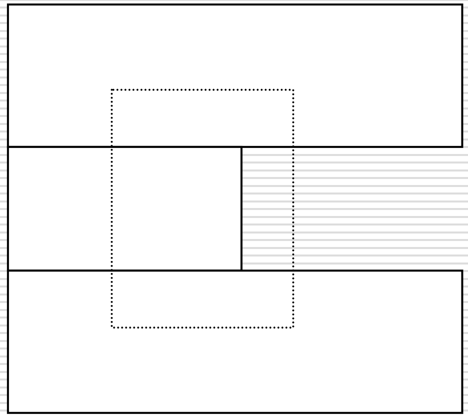
- Build basic grid
- Flagging ratio <60%: bisect
- Go on recursive

- Merge where necessary, to avoid bad cell/border ratio



Automatic grid adaption

- Lock for proper nesting



Not properly nested!

Initialise new grids

- Transfer of solutions where possible
- Building an average where only coarser meshes exist

That it is!

Code tests (J J Quirk, Ph.D. Thesis)

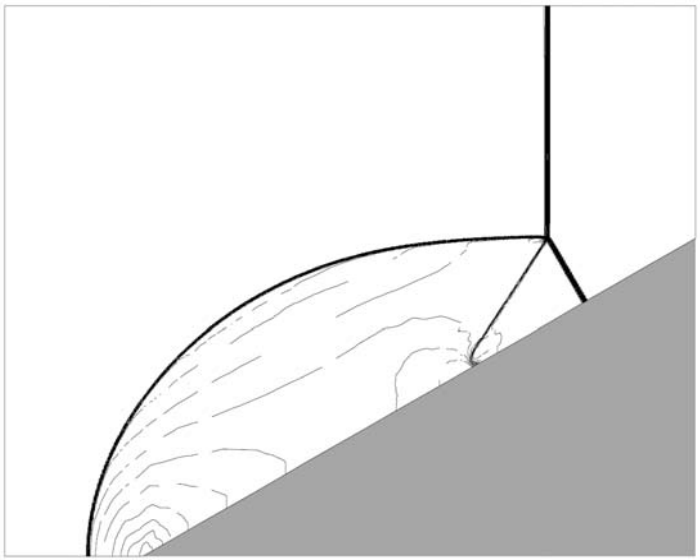
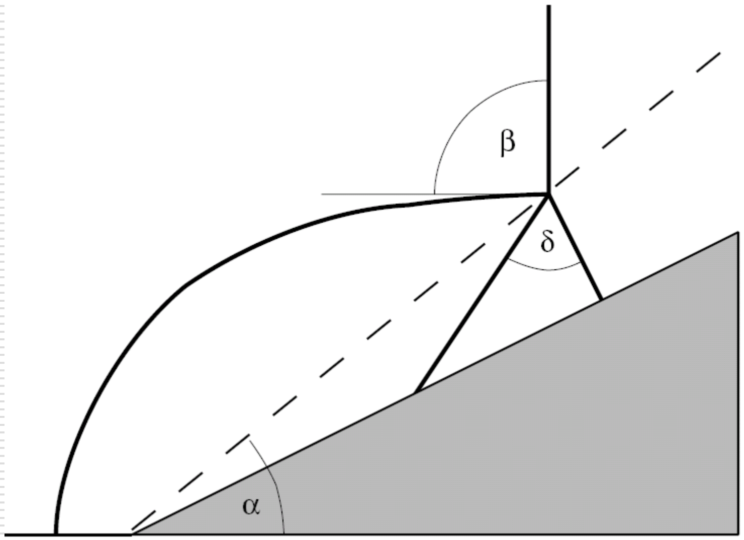


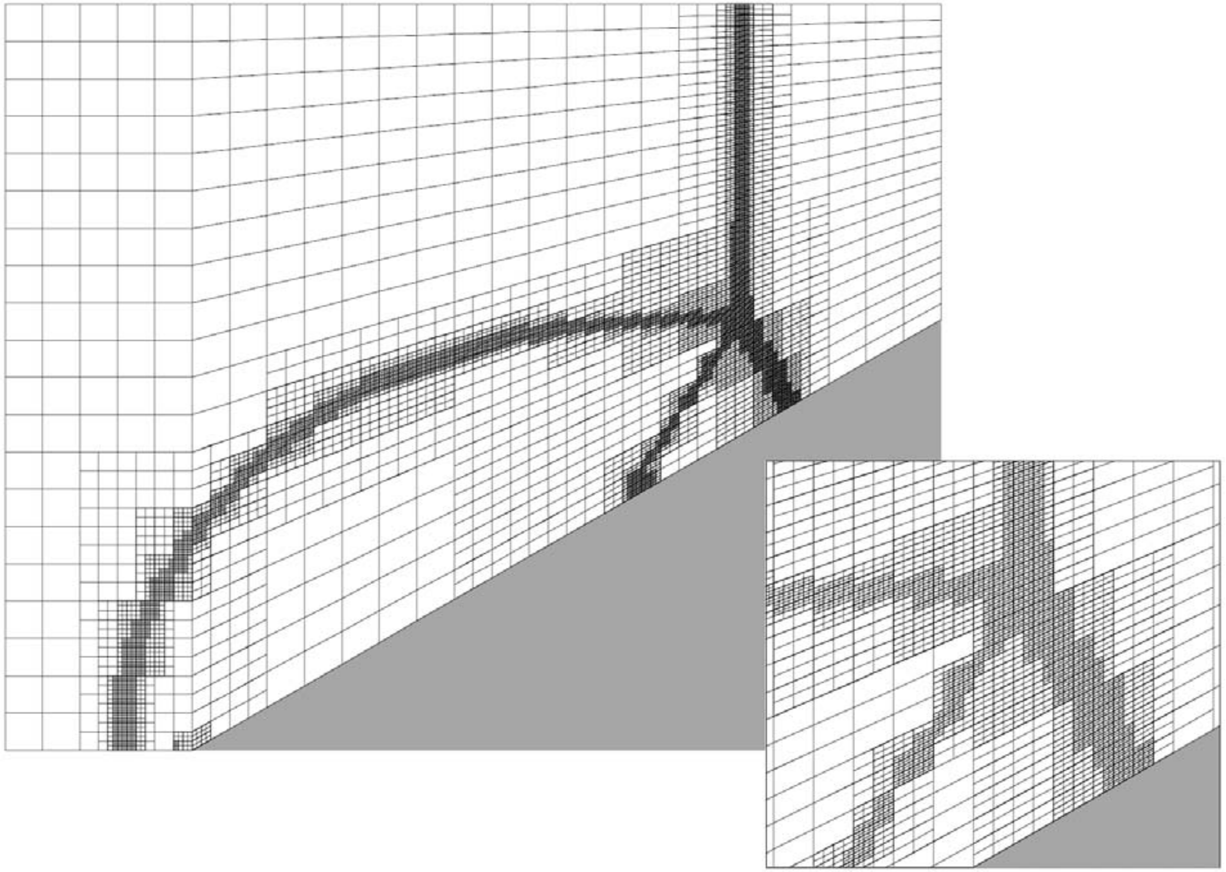
Figure 4.24: Single Mach reflection: $M_s = 2.12$, $\theta = 30^o$, $\gamma = 1.4$.



Angle	Experiment	Computation
α	38.5	38.4
β	92.0	91.6
δ	63.5	63.7

Code tests

Final grid structure



CPU-time split-up

Grid integration takes most of time.

Justifies AMR (for this problem)

Procedure	**% of total run time**
Integrate Grid	93.5
Evaluate gradient for flagging	1.6
Adapt	2.5
Project Solution	0.9
Apply Fixup	0.1

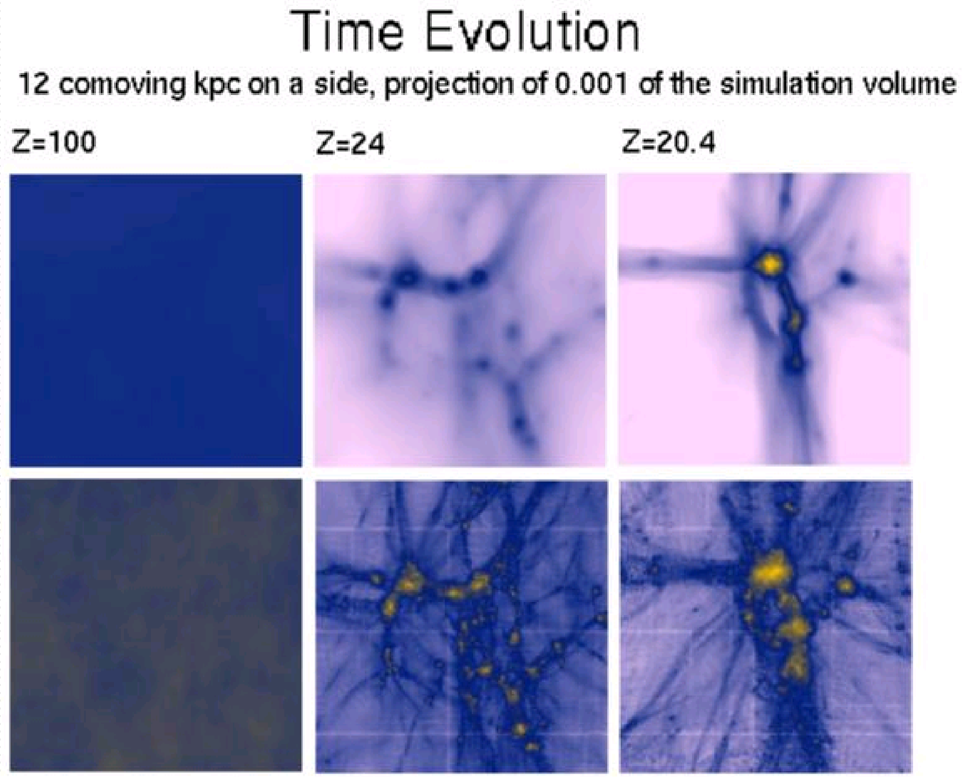
Adding different physical Processes

- Radiative cooling (in optically thin case)
- Nuclear burning
- Physical viscosity
- Radiation transport
- Photoionization
- Gravitational forces

Application in early star formation (T. Abel et al.)

Problem setup:

- Dark Matter with SPH
- Gas with AMR
- Chemical and radiative processes included
- Start at Z=100 with 120kpc box (comoving)



Application in early star formation

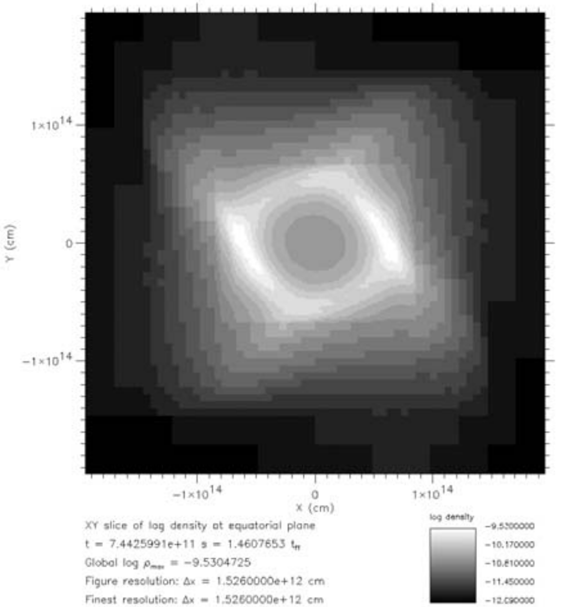
Appropriate refinement criterion

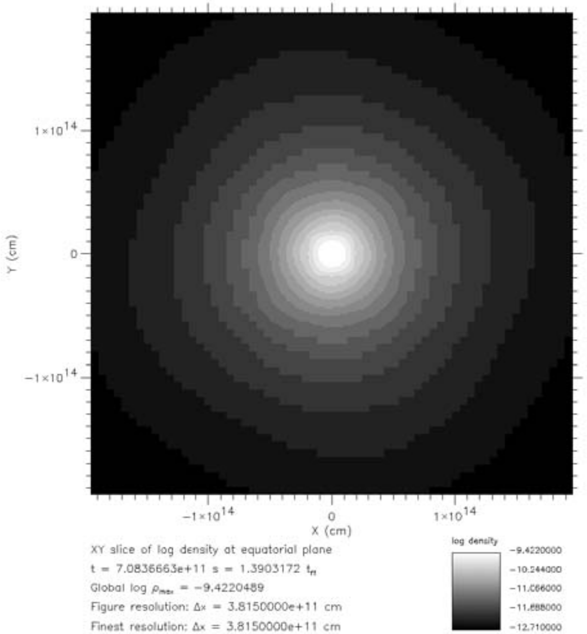
- gravitational collapse

  → Need resolved Jeans length scale $\lambda_J = \left(\frac{\pi c_s^2}{G\rho} \right)^{1/2}$
- Else: artificial fragmentation

Collapsing gas cloud

- Left: J = 0.5
- Right: J = 0.125





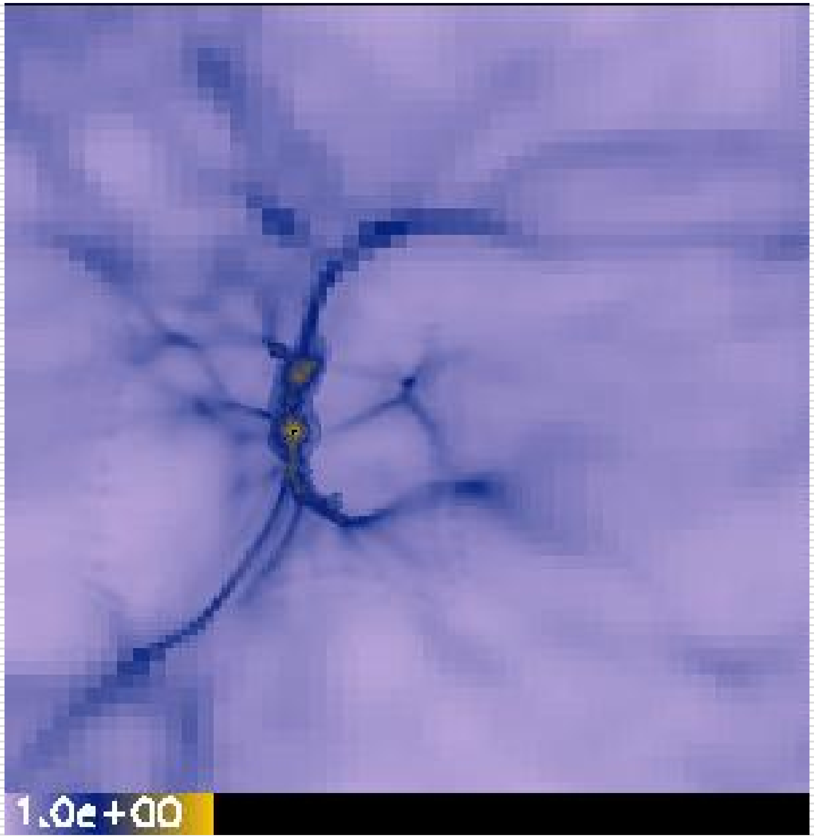
(Truelove et al.)

Application in early star formation

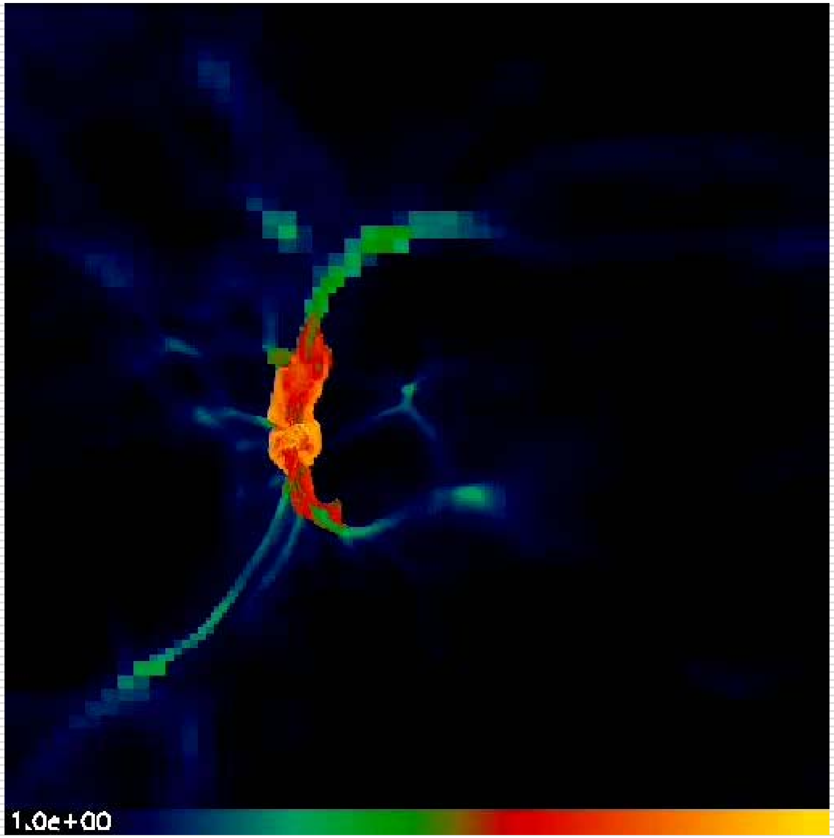
Results

- Calculation stopped when first "star" is built,
    - Time steps become very small
    - Implemented physics unreliable
- >5500 grids, 27 refinement levels, 260^3 grid cells
- Pregalactic halo with a total mass of $7 \cdot 10^5 M_\odot$
- Protostar in the center

Application in early star formation



Gas Density



Temperature

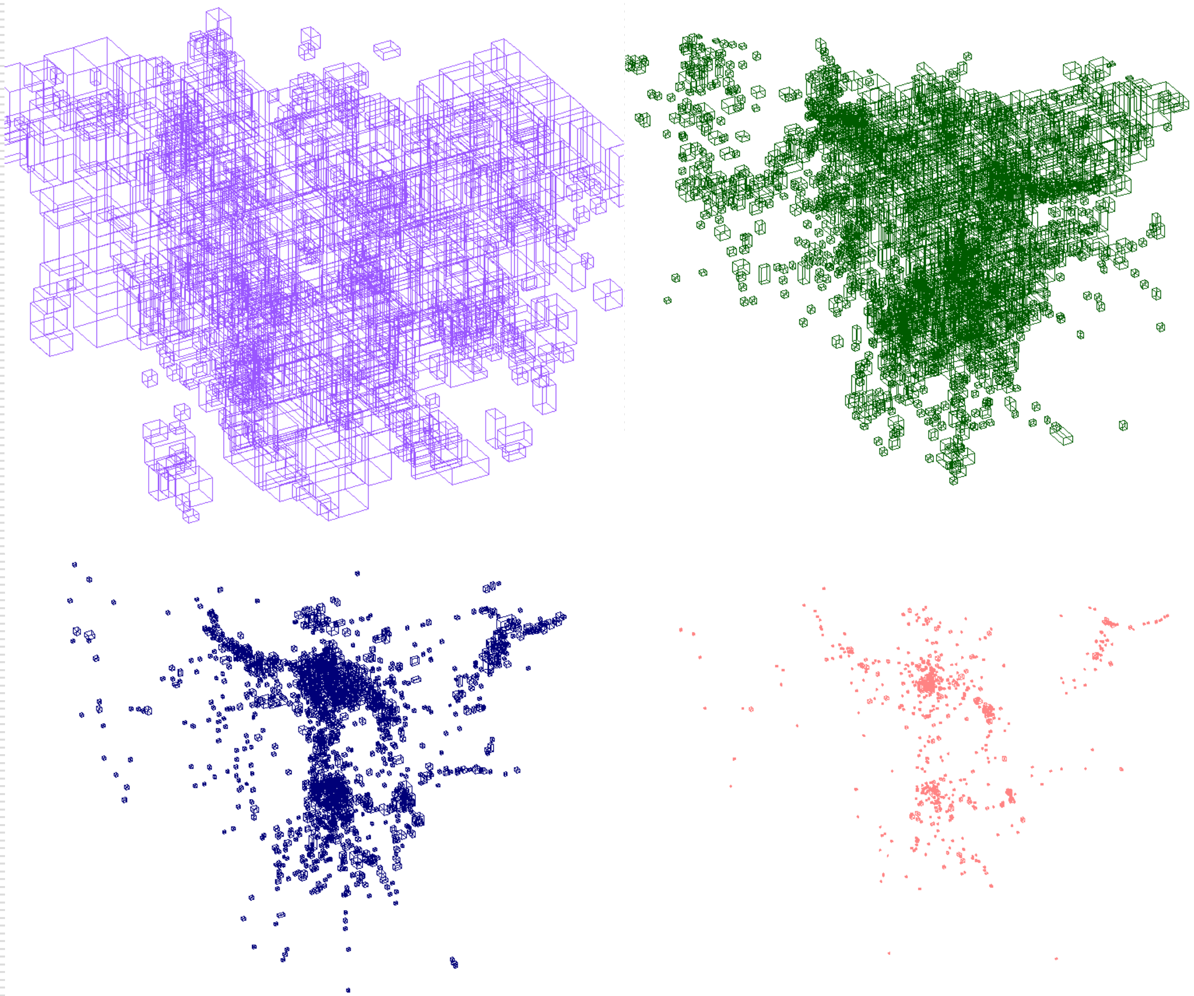
Videos: www.slac.stanford.edu/~tabel/GB

Application in early star formation

Grids

Level 3

Level 4



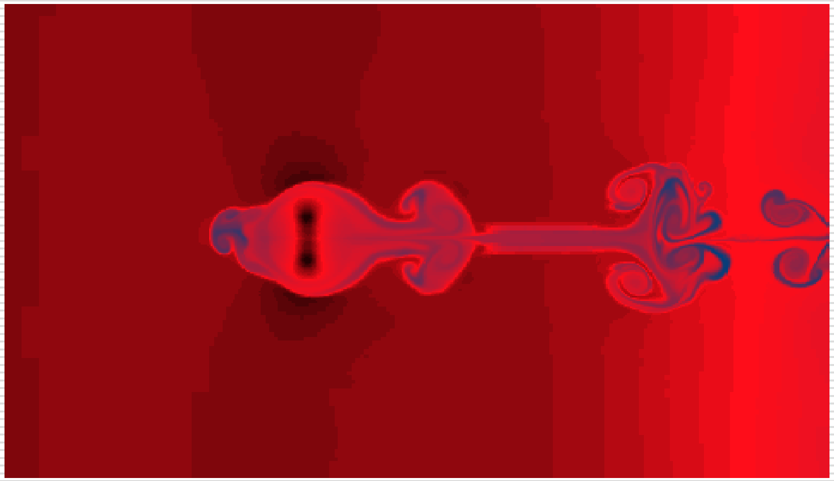
Level 5
1924 grids

Level 6
1423 grids

Jan Homann, Adaptive Mesh Refinement, 2005

Other applications

- Supernova explosions
- Supernova remnants
- Protostellar disks
- …



Interaction of Shock Waves with Inhomogeneous Media

(A. Poludnenko et al., Rochester)

References

- **J J Quirk**, 1991, Ph.D. Thesis, An Adaptive Grid Algorithm for computational Shock Hydrodynamics
- **T.Plewa, E. Müller**, 2005, arXiv:astro-ph, AMRA: An adaptive mesh refinement hydrodynamic code for astrophysics
- **T. Abel, G. Bryan, M. Norman**, 2000, The Astrophysical Journal 540:39, The formation and fragmentation of primordial molecular clouds
- **T. Abel, G. Bryan, M. Norman**, 2002, Science 295:93, The formation of the first star in the universe
- **K. Truelove**, 1997, The Astrophysical Journal 489:179, The Jeans Condition: a new constraint on spatial resolution in simulations of isothermal self-gravitational hydrodynamics
- **R. Klein et al.**, 1998, arXiv:astro-ph, Gravitational collapse and fragmentation in molecular clouds with adaptive mesh refinement hydrodynamics
- **W. Hillebrandt, E. Müller**, 2005, Lecture Notes, Hydrodynamik: Grundlagen und numerische Verfahren
- **M. J. Berger, P. Colella**, 1989, Journal of Computational Physics 82:64
- **R. J. LeVeque, D.Milhalas, E.A. Dorfi, E. Müller**, 1997, Computational Methods for Astrophysical Fluid Flow