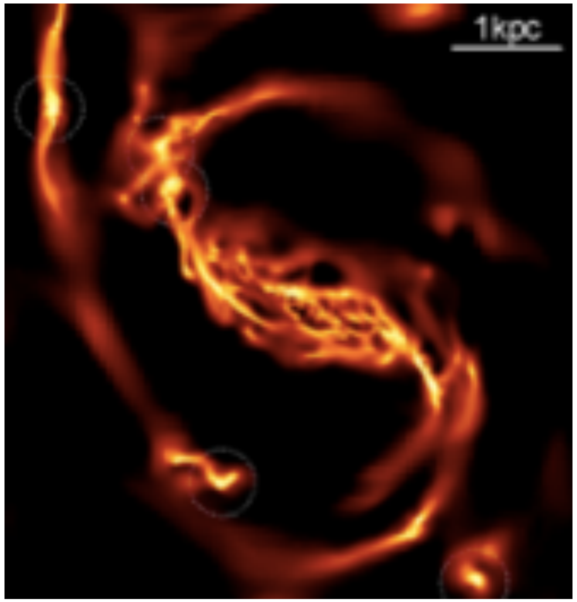
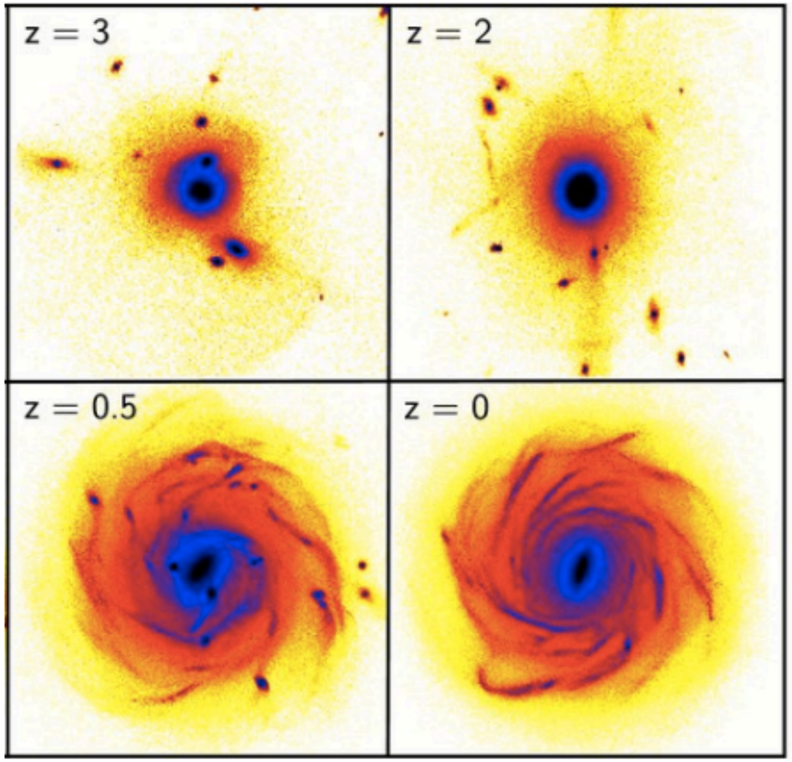
Astrophysical fluid dynamics:
N-body, gravity and hydrodynamics
Applications in cosmology, galaxy and star formation



Frédéric Bournaud – CEA Saclay
frederic.bournaud@cea.fr

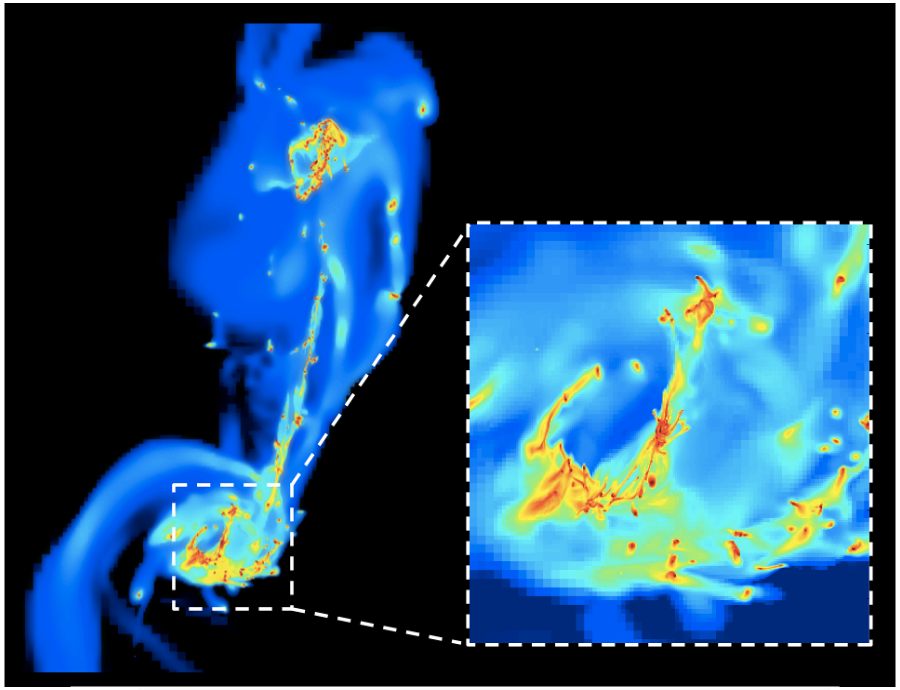
Some words on my research

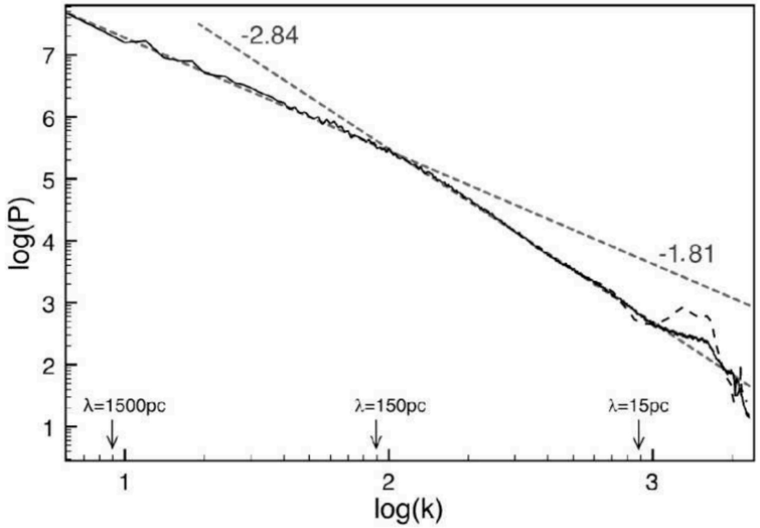
- Based in CEA Saclay, near Paris, France.
- Uses mainly the RAMSES code.
- Galactic dynamics and structure
- ISM hydrodynamics on galactic scales, star formation regulation/triggering
- Interaction of galaxies with their close environment



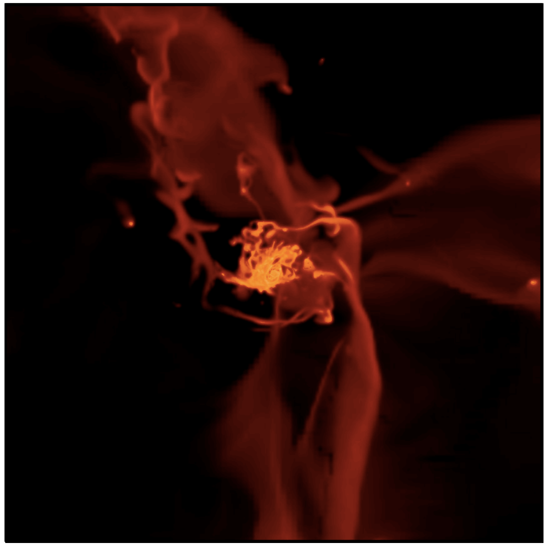
Some words on my research

- Based in CEA Saclay, near Paris, France.
- Uses mainly the RAMSES code.
- Galactic dynamics and structure
- ISM hydrodynamics on galactic scales, star formation regulation/triggering
- Interaction of galaxies with their close environment

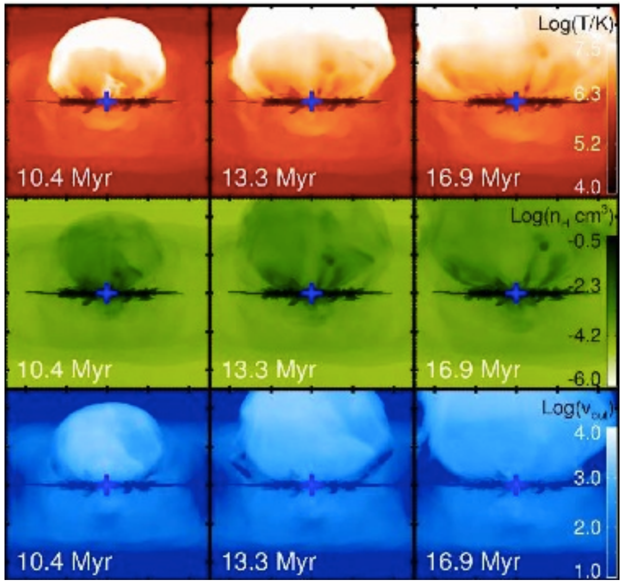




Some words on my research



- Based in CEA Saclay, near Paris, France.
- Uses mainly the RAMSES code.
- Galactic dynamics and structure
- ISM hydrodynamics on galactic scales, star formation regulation/triggering
- Interaction of galaxies with their close environment



The lectures

- Self-gravitating systems :
  - N-body modelling
  - Poison solvers and schemes for self-gravity
- Hydrodynamics :
  - Particle-based and grid-based methods
  - AMR and unstructured mesh
- Gravity-hydro simulations in practice :
  - Some criteria for « reliable » simulations
  - Sub-grid modeling of star formation, black holes, and feedback
- Recent highlights and outstanding challenges

The lectures

- **Self-gravitating systems :**
  - Vlassov-Poisson systems
  - N-body modelling and particle-mesh
  - Force softening
  - PM, P3M, TPM methods
  - Poisson solvers
  - Self-force correction
  - Time intragrators
  - multi-resolution and AMR

- Hydrodynamics
- Gravity-hydro simulations in
- Recent highlights and outstanding challenges

Vlassov-Poisson systems

Collisionless Boltzmann equation :

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial}{\partial t} f(\mathbf{x}, \mathbf{p}, t) + \frac{\mathbf{p}}{m} \frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}, \mathbf{p}, t) - m \nabla_{\mathbf{x}} . \Phi(\mathbf{x}) \frac{\partial}{\partial \mathbf{p}} f(\mathbf{x}, \mathbf{p}, t) = 0$$

Poisson equation for Φ :

$$\Delta \Phi(\mathbf{x}) = 4\pi G m \int f(\mathbf{x}, \mathbf{p}, t) \mathrm{d}^3 \mathbf{p}$$

(modulo average density for infinite or periodic system)

Solving gravitational dynamics for any system requires only to
solve this set of two equations… but the CPU and/or memory cost is a real problem

Pure fluid aproach : memory issues

- Discretizing the entire phase-space on a 6D grid

- N^6 resolution elements, each containing at least 1 real for the phase-space density

- Even for just (256^3)^2 this is >1GB of memory.

- Velocity discretization error propagates (integrated) on real space errors

- The CPU time is not necessariy prohibitive but the memory cost becomes prohibitive. Cannot fit a 1024^6 on a few thousands of cores !

Pure N-body aproach : CPU issues

- Storing a millions of particles (3D each at each time *t*) in a modern computer is not an issue

- Memory cost OK to store the real elements at any given instant

- Problem is the force calculation :

$$\Delta\Phi(\mathbf{x}) = \frac{4\pi G m}{a} \int f(\mathbf{x}, \mathbf{p}, t) \mathrm{d}^3\mathbf{p}$$

The Poison integral becomes the direct summation of the pair interaction forces, each involves the calculation of a distance x^2+y^2+z^2   ~N^2

Here the CPU cost rapidly becomes prohibitive + hard to parallelize this problem (long-range P-P interactions).

Particle-mesh methods

- Need to discretize some dimensions to get a reasonable CPU cost
- But not all for memory.

$\Rightarrow$ Typically, discretize the real space on a 3D grid
+ use particles with real velocities (at the numerical accuracy limit)

$\Rightarrow$ No prohibitive calculation of particle-particle forces
$\Rightarrow$ No huge 6D grid, only a 3D grid + millions of particles.
$\Rightarrow$ No velocity-to-position error propagation (at first order)

Almost all Vlassov-Poisson methods are based on Particle-Mesh schemes

2-body interactions versus long-range forces

Still a (manageable) memory issue :

A galaxy contains 10^{11} stars.
Cannot do Nbody with 10^{11} bodies (particles).
(except maybe pure gravity at super-high resolution, no real science case…)

No way to model all stars of a galaxy cluster, all dark matter particles in a dark matter halo, all gas clouds and cores in a galaxy, etc…

General consequence :
In astrophysical simulations, Nbody is done with N << real number of elements

Exceptions : star clusters, planetary systems.

2-body interactions versus long-range forces

In astrophysical simulations, Nbody is done with N << real number of elements

In a real galaxy, a star's movement is dominated by the global galaxy mass,
somewhat influenced by galactic structures (spiral arms etc),
not influenced by the closest neighbours

In a model with 10^6 particles, the closest neighbour's force is not so
negligible anymore => change in the **long-range/short-range** force ratio !

Two-body relaxation timescale

2-body interactions versus long-range forces

In astrophysical simulations, Nbody is done with N << real number of elements

In a model with 10^6 particles, the closest neighbour's force is not so negligible anymore => change in the **long-range/short-range** force ratio !

Two-body relaxation timescale

Star cluster

Globuler cluster

$$T_{2body} \sim T_{cross}/10 \;\; N/\ln(N)$$

Galaxy

2-body interactions versus long-range forces

In astrophysical simulations, Nbody is done with N << real number of elements

In a model with 10^6 particles, the closest neighbour's force is not so negligible anymore => change in the **long-range/short-range** force ratio !

Two-body relaxation timescale

Star cluster 10-100 Myr

Globuler cluster 1-5Gyr

$$T_{2body} \sim T_{cross}/10 \; N/\ln(N)$$

Galaxy >10Gyr

Timescale required for:

- Orbit scattering by close neighbour
- Relaxation & dynamical evaporation
- Mass segregation (equipartition of energy per particle).

Gravitational softening

Reducing the number of particles N over-estimates the short-range interactions

Solution = gravitational softening

Idea : replace the potential energy 1/r (force 1/r^2)
with a flatter gradient at low r (vanishing forces)

1/r can be replaced with **1/(r+a)** or **1/sqrt(r^2+a^2)**
a = softening length

First shape prefered to damp forces without cancelling energy gradient
Second shape : force doeesn't stay about constant below a but falls to zero

Advantage : - often inherent to grid solver
- no singularity at r=0

But : should not be used for low-N systems

Particle Mesh schemes (PM)

Ingredients :

- The mass is distributed in particles, with individual velocities
- A grid is used to solve gravity

Steps :

- Compute the grid density from particles (PIC interpolation)
- Solve Poisson on the grid (Poisson solver)
- Interpolate the force to each particle $(d_t\mathbf{v} = - d_x\Phi)$
- Integrate the particle motion (time integration $d_t\mathbf{x} = \mathbf{v}$)

First step (PIC) is usually done via multi-linear interpolation between particle positions and grid nodes.

- Quick algorithm
- Accurate enough (see force interpolation later)

Particle-Particle Particle-Mesh schemes (P3M)

The softening length is often linked to the Poisson solver,
and not smaller than the grid cell size.

The softening length can often be too large :
not a formal problem if the true 2body-relaxation timescale is Gyrs,
but lowers the resolution

P3M uses:
- Grid-based gravity on long ranges
- Direct summation on short scales
- A carefully-chosen softening length.

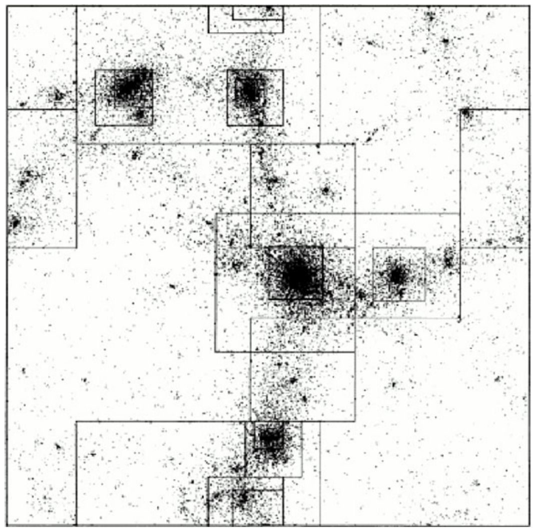
Adaptive Particle-Particle Particle-Mesh schemes (AP3M)

Pro : P3M allows a carefully-chosen softening length for short-range interactions

Cons :

- Resolution jump at boundaries
- In a very heterogeneous system (e.g. cosmology at low redshifts) the PP part dominates over the PM part (in CPU time). This means very high accuracy but unfeasible – tends toward the direct particle-particle scheme.

Solution : refine the mesh in regions with clustered particles
=> AP3M : adaptive mesh + PP on short scales (Couchman 1991).



Couchman 1995 AP3M-SPH

« Tree-code » schemes (TPM)

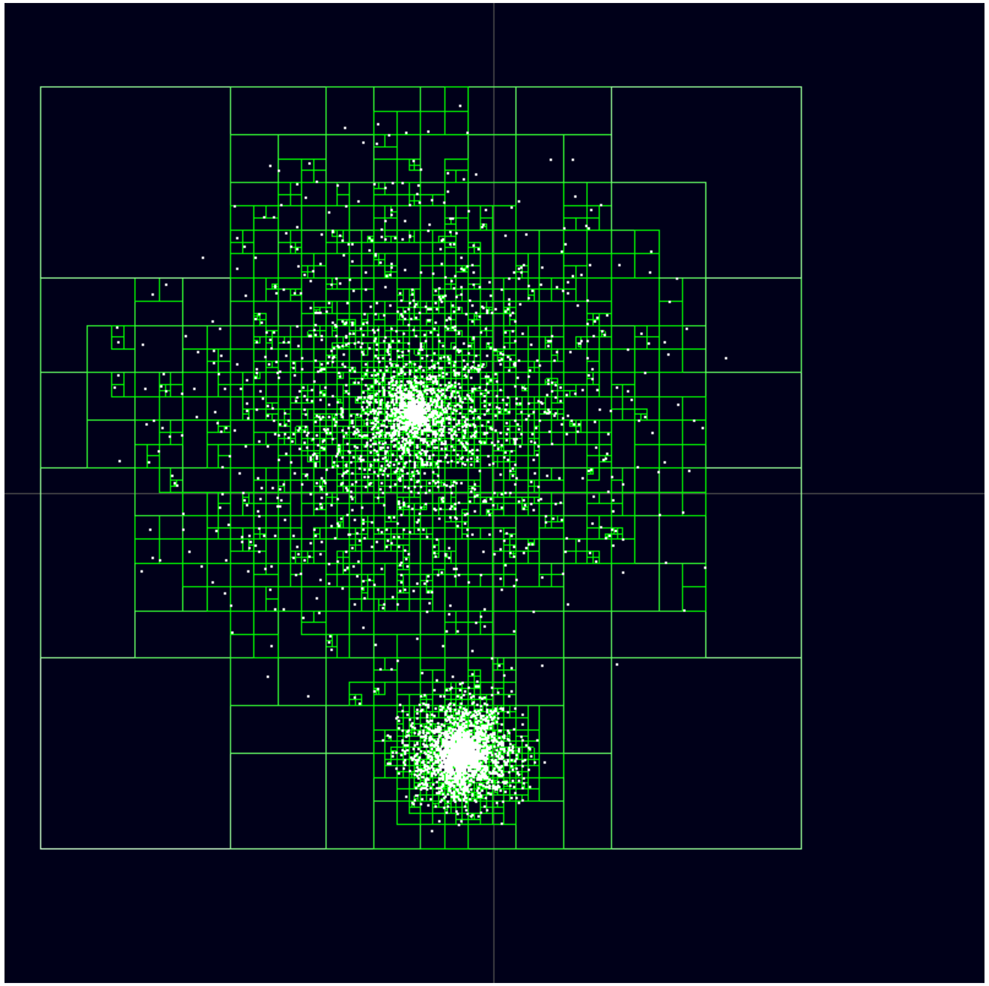
- Spatial decomposition in octs

- Until 1 particle per cell, or a low number of particle per cell is reached

- The tree of octs is used to compute forces :

P-P on short distances

P-node on large scales



Barnes & Hut 86

Particle Mesh schemes (PM)

Ingredients :

- The mass is distributed in particles, with individual velocities
- A grid is used to solve gravity

Steps :

- Compute the grid density from particles (PIC interpolation)
- Solve Poisson on the grid (Poisson solver)
- Interpolate the force to each particle ($d_t\mathbf{v} = - d_x\Phi$)
- Integrate the particle motion (time integration $d_t\mathbf{x} = \mathbf{v}$)

First step (PIC) is usually done via multi-linear interpolation between particle positions and grid nodes.

- Quick algorithm
- Accurate enough (see force interpolation later)

Poisson solvers : FTT scheme

Gravitational potential : $\Phi(\mathbf{x}) = -\int_{\mathbf{R}^3} \frac{G}{|\mathbf{x}-\mathbf{r}|} \rho(\mathbf{r}) dv(\mathbf{r}).$

This is the convolution (in 3D) of the density $\rho(\mathrm{r})$ and the $1/\mathrm{r}$ function
In practice softening should be applied to the $1/\mathrm{r}$ function : no change

In the Fourier space : $\mathcal{FT}(\Phi)\ (\mathbf{k}) = \mathcal{FT}\ (1/r)\ (\mathbf{k})\ \mathrm{x}\ \mathcal{FT}\ (\rho)\ (\mathbf{k})$

$\Rightarrow$ Direct product, complexity ~N

FFT complexity is Nlog(N)
FFT3D => set of FFT1D in each dimension

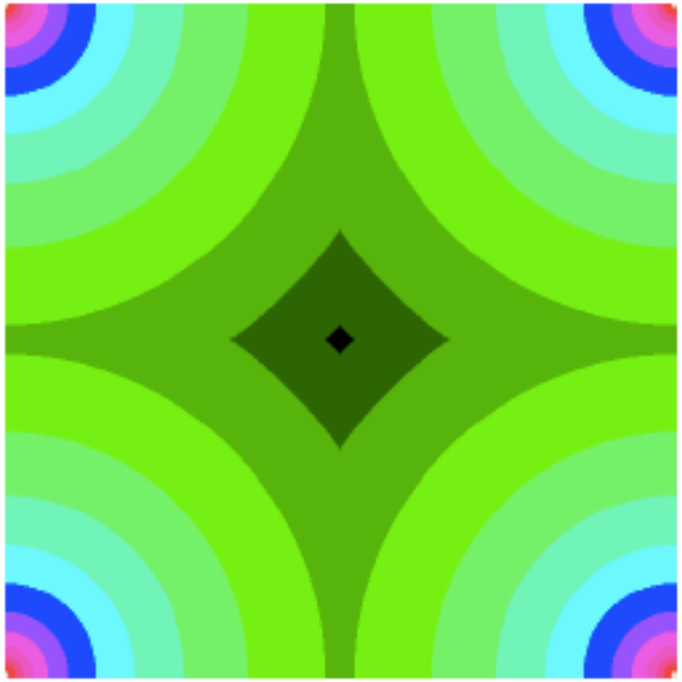
MPI ? CPU aspects are OK, although non-equal compute times
Non-local : tends to be memory-heavy
Good for vector-computing, GPU…

Poisson solvers : FTT scheme – boundary conditions

FFT assumes a periodic density distribution: useful in cosmology

For non-periodic systems:

« zero padding » technique
Use a 2Nx2Nx2N grid
Cancel the Green function on large distances
⇒ Periodic density still assumed
⇒ Forces with Fourier images are cancelled out



Poisson solvers : FTT scheme – boundary conditions

FFT assumes a periodic density distribution: useful in cosmology

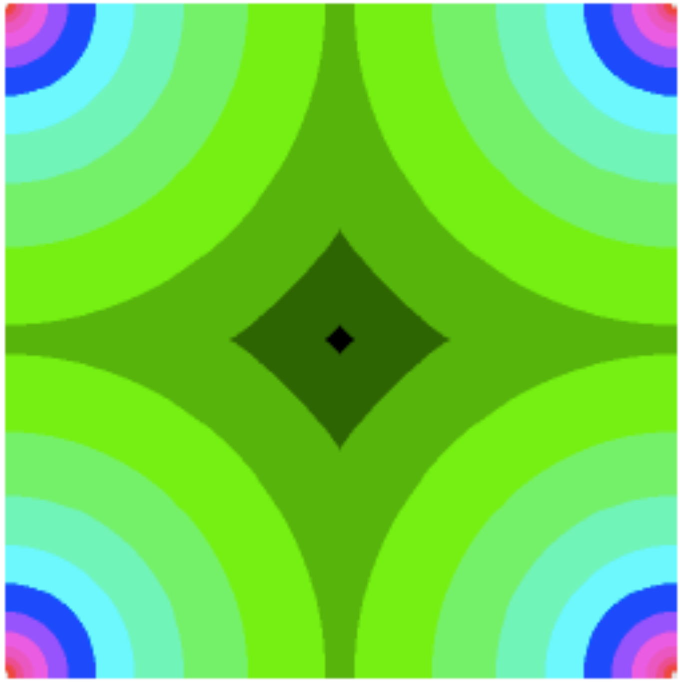
For non-periodic systems:

« zero padding » technique **(x8)**

Or

Use surface masses on the box limits (faces, edges, corners) to screen-out the Fourier images

Costly in real space but easy in the Fourier space **(~ x1.5)**

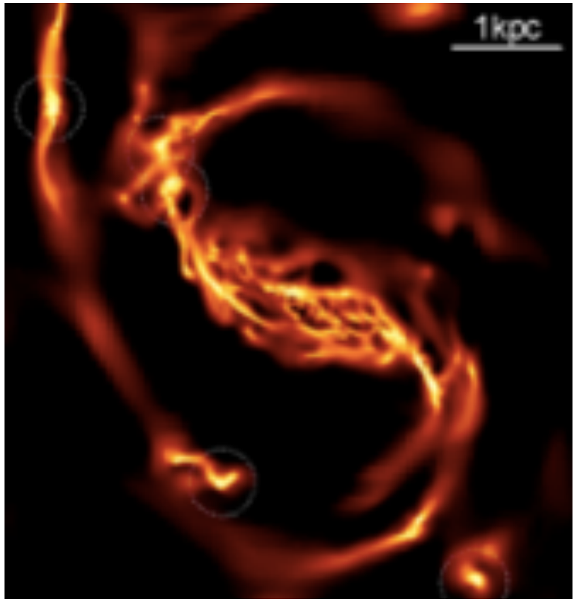


(James 1977, JCP 25, 71)

Astrophysical fluid dynamics:

N-body, gravity and hydrodynamics

Applications in cosmology, galaxy and star formation



Frédéric Bournaud – CEA Saclay
frederic.bournaud@cea.fr

The lectures

- **Self-gravitating systems :**
  - Vlassov-Poisson systems
  - N-body modelling and particle-mesh
  - Force softening
  - PM, P3M, TPM methods
  - Poisson solvers
  - Self-force correction
  - Time intragrators
  - multi-resolution and AMR

- Hydrodynamics
- Gravity-hydro simulations in practice
- Recent highlights and outstanding challenges

The lectures

- **Self-gravitating systems :**
  Vlassov-Poisson systems
  N-body modelling and particle-mesh
  Force softening
  PM, P3M, TPM methods
  Poisson solvers
  Self-force correction
  Time intragrators
  multi-resolution and AMR

- Hydrodynamics
- Gravity-hydro simulations in practice
- Recent highlights and outstanding challenges

Poisson solvers : Relaxation solvers

Idea : start with an initial guess and converge toward a solution of Poisson Eq.

- **Jacobi Method:**

  in 2D $\phi_{i,j}^{n+1} = \frac{1}{4}\left(\phi_{i+1,j}^{n} + \phi_{i-1,j}^{n} + \phi_{i,j+1}^{n} + \phi_{i,j-1}^{n}\right) - \frac{1}{4}\rho_{i,j}$

  Slow convergence (up to ~N^2 iterations or more)

- **Over-relaxation (Gauss-Seidel method):**

  Force faster convergence by removing a fraction (between 0 and 100%) of the previous-order solution

  Formal convergence for N^2 iterations at most, usually faster

- **Conjugate Gradient:**

  Uses mostly local data (MPI…)

  Reaches numerical truncation accuracy in ~N iterations

  Low sensitivity to the initial guess

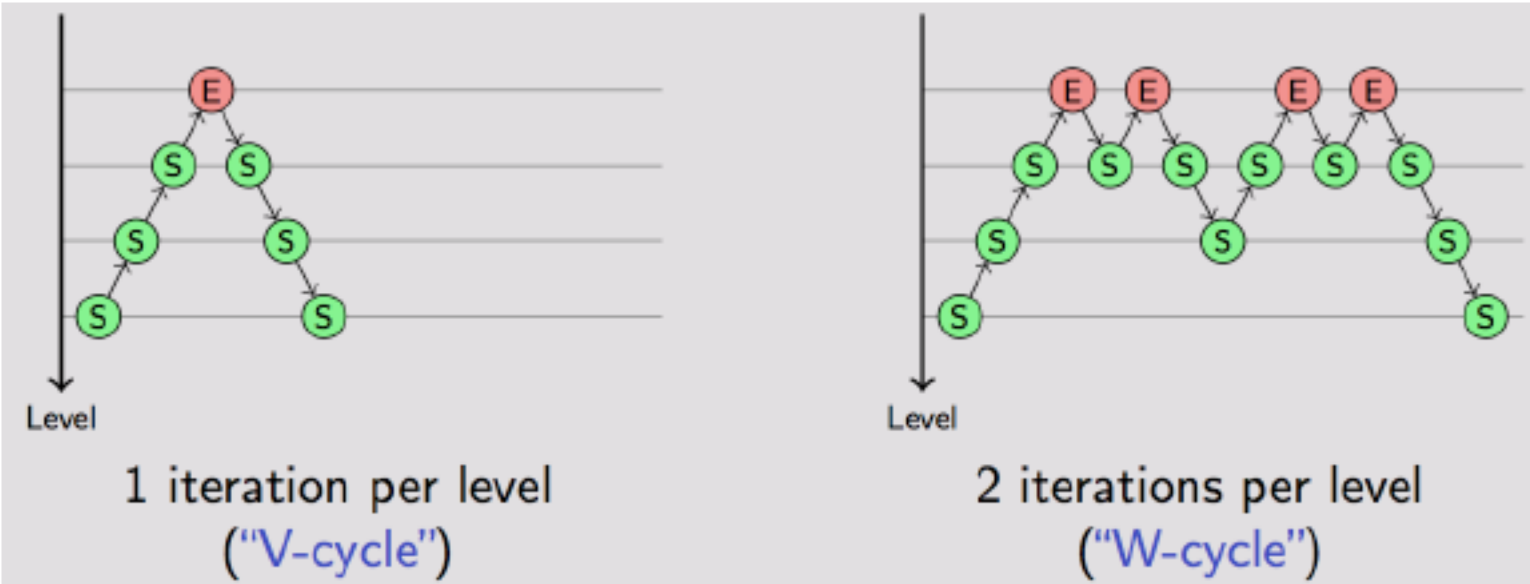
Multi-grid schemes for Poison solvers

- Perform a few realxation iterations on fine levels first (smoothing the initial guess)

- Restrict residuals to the coarse levels

- Iterate relaxation algorithm on the coarse grid

- Interpolate and correct on the fine grid

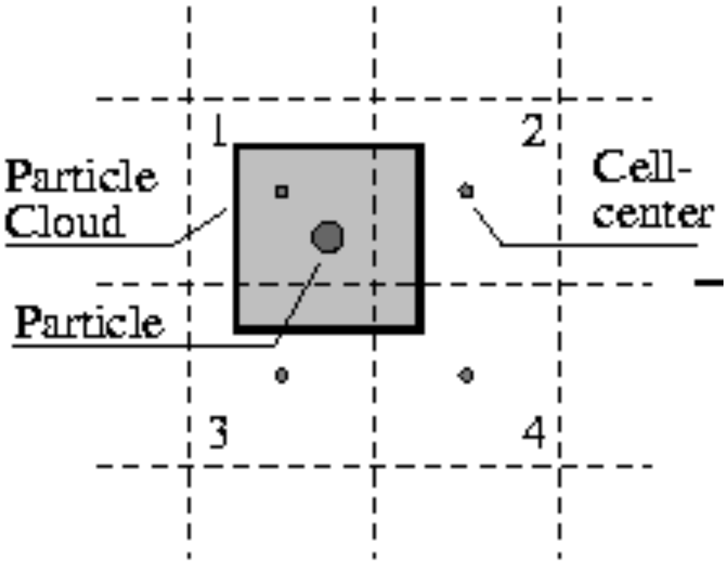


Various iteration cycles
Convergence quasi independent from the initial guess (Guillet & Teyssier 2011)

Force interpolation

Now we have the gravitational potential on a grid.
Force interpolation should ensure momentum conservation:

- no self force
- pairs of particles undergo opposite forces

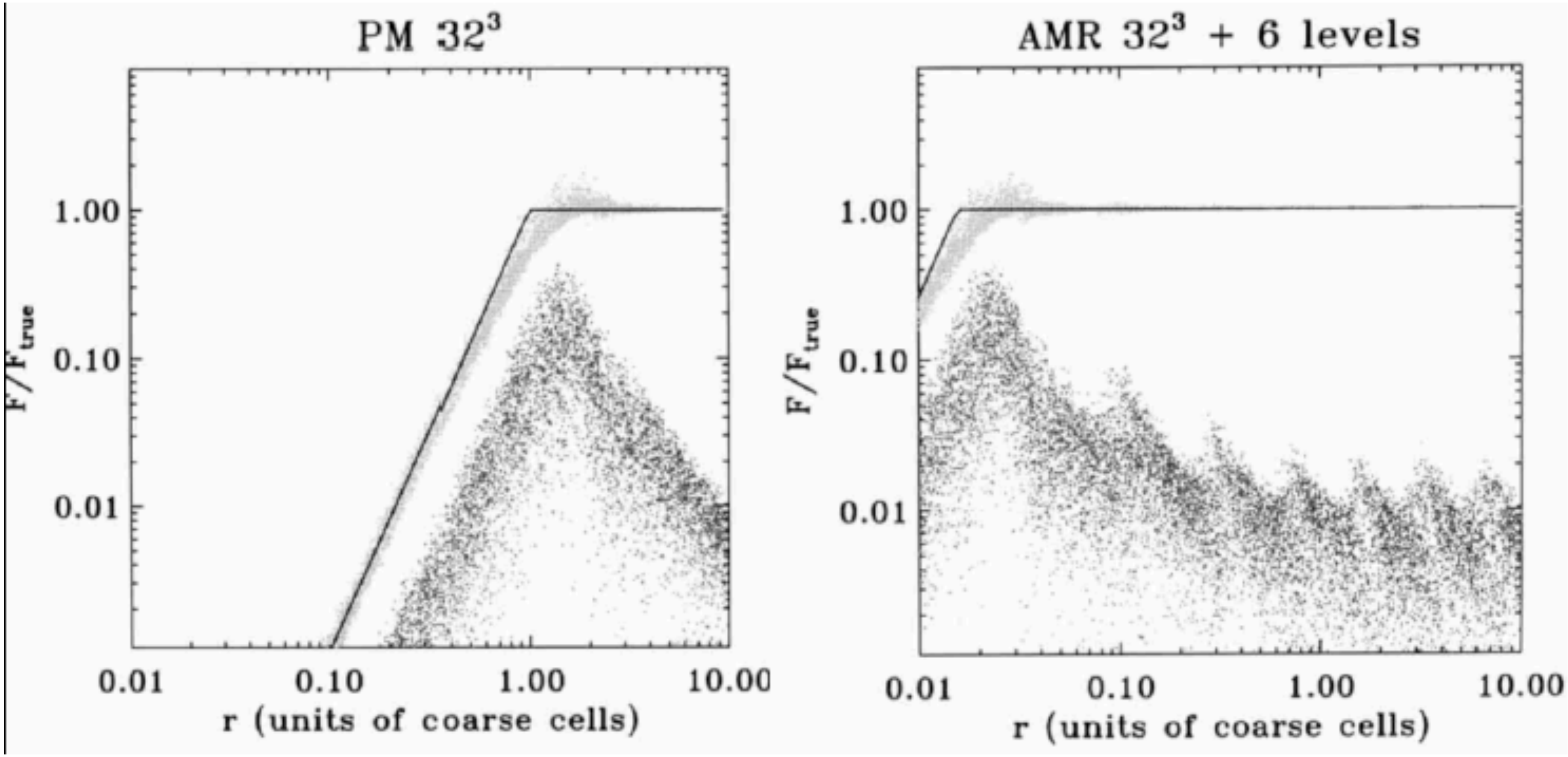


Multi-linear interpolation similar to the Mesh interpolation scheme ensures zero self-force, on Cartesian grids.

Interpolation between 8 grid nodes (in 3D) for each particle gives the gravitational force

Other geometries often have a self-force (depending on the interpolation scheme)
=> self-force can be computed or estimated, then corrected.

Force accuracy



(Teyssier 2009)

Force accuracy reaches 1% in 2-3 cells.
Adaptive grids induce changes in the interpolation and softening, below the intrinsic accuracy.

Time integration

Explicit Euler integrator :

$$(\boldsymbol{x},\boldsymbol{v})^{n+1} = (\boldsymbol{x},\boldsymbol{v})^{n} + \mathrm{d}t\ \boldsymbol{f}((\boldsymbol{x},\boldsymbol{v})^{n}) \qquad where \quad \boldsymbol{f}=(m\boldsymbol{v},-\mathrm{d}\Phi/\mathrm{d}\boldsymbol{x})$$

Implicit Euler integrator :

$$(\boldsymbol{x},\boldsymbol{v})^{n+1} = (\boldsymbol{x},\boldsymbol{v})^{n} + \mathrm{d}t\ \boldsymbol{f}((\boldsymbol{x},\boldsymbol{v})^{n+1})$$

Problems : in simple potentials, a single particle will continuously gain (respectively loose) gravitationnal potential energy.
Trajectories are non-reservible.

=> These integrators are not simplectic .

Time integration – symplectic integrators

Simplest (Euler-like) symplectic integrator :

$$(\boldsymbol{x},\boldsymbol{v})^{n+1} = (\boldsymbol{x},\boldsymbol{v})^{n} + \mathrm{d}t\ \boldsymbol{f}\,(\boldsymbol{x}^{n},\boldsymbol{v}^{n+1})$$

Second-order integrators :

- leap-frog

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^{n} + \mathrm{d}t\ \boldsymbol{v}^{n+1/2}$$

$$\boldsymbol{v}^{n+3/2} = \boldsymbol{v}^{n+1/2} + \boldsymbol{F}\,(\boldsymbol{x}^{n+1})\,\mathrm{d}t$$

- kick-drift-kick

  Slightly longer calculation but matched timesteps for **v** and **x**

Particle Mesh schemes (PM) - **summary**

Ingredients :

- The mass is distributed in particles, with individual velocities
- A grid is used to solve gravity

Steps :

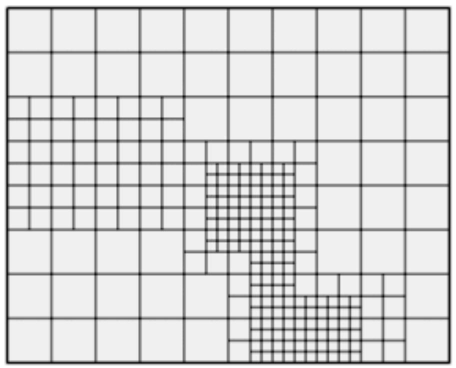
- **Compute the grid density from particles (PIC interpolation)**
- **Solve Poisson on the grid (Poisson solver)**
- **Interpolate the force to each particle** $(d_t \mathbf{v} = - d_x \Phi)$
- **Integrate the particle motion (time integration** $d_t \mathbf{x} = \mathbf{v})$

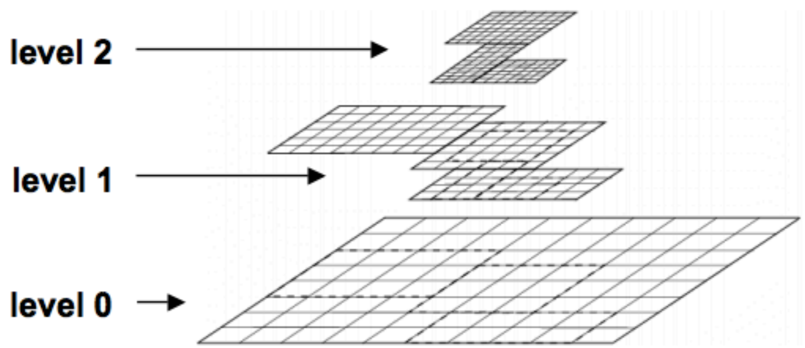
The basic PM loop is now complete… We can move to AMR techniques…

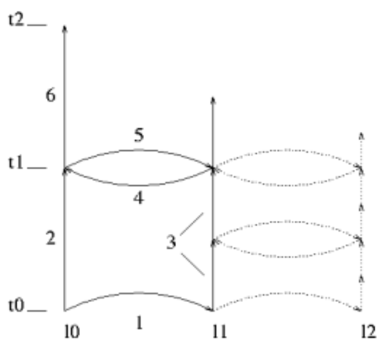
AMR techniques

An AMR structure will usually start from a « coarse level » uniform grid
=> reduce the number of intermediate levels
=> domain decomposition at least on the coarse level grid

Simplest version : patch-based refinement







Recursive time intergration :
- coarse+fine levels
- fine levels (sub-loops as required)
- synchronize and re-grid (if needed)
- iterate all levels again

Patch-based AMR

Reasons for patch-based strategies :

-- Simplicity

-- Lower rate of coarse/fine interpolation in critical zones (if controlled)

-- Suited for vector computing or GPU

-- No need for domain decomposition :

each processor can handle a level, or a patch (esp. if constrained size)

don't need to have parent/child cells on same CPU

Main drawbacks :

-- memory inefficient, especially if patch size is constrained

-- lack of real domain decomposition can be a drawback (analysis, sub-grid)

Cell-based AMR (tree-based)

Reasons for tree-based strategies :

-- Minimal memory for given refinment requirements

-- Simpler data structure (e.g. oct-tree) once hydro is coded

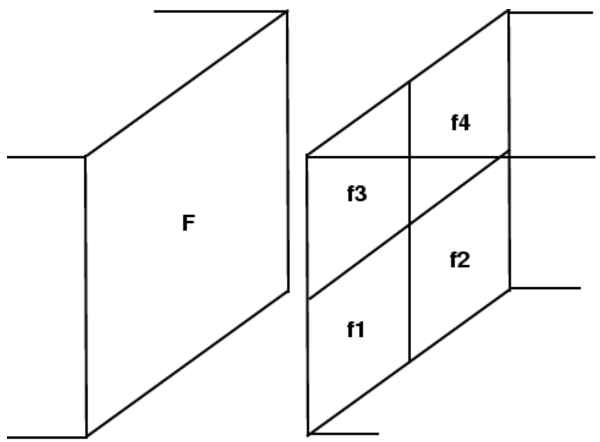
-- Adapted to any system (patches follow clustering)

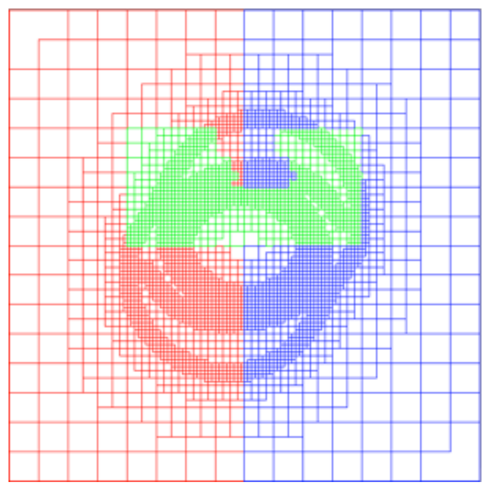
-- There is a real domain decomposition (a constraint but also advantages)

Main drawbacks :

-- fluid elements may undergo more refinement/de-refinement

-- communication costs can be higher for some « ideal » geometries



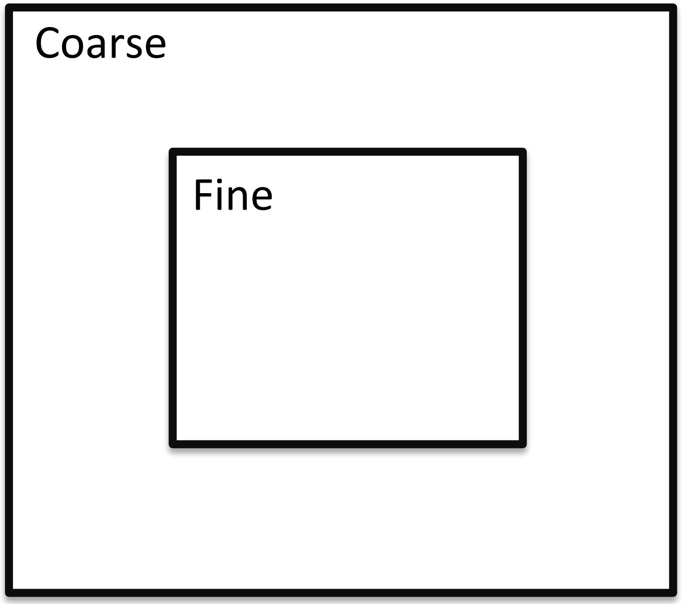


Example of RAMSES domain decomposition

AMR Poisson solver

- Relaxation methods can still be used (such as Conjugate Gradient)
- Multi-grid could still be used on any AMR level (esp. if patch-based), but generally in tree-based schemes the multi-grid becomes inherent to the AMR structure

Basic approach : « Pandora » scheme. Coarse levels ignore the content of fine levels



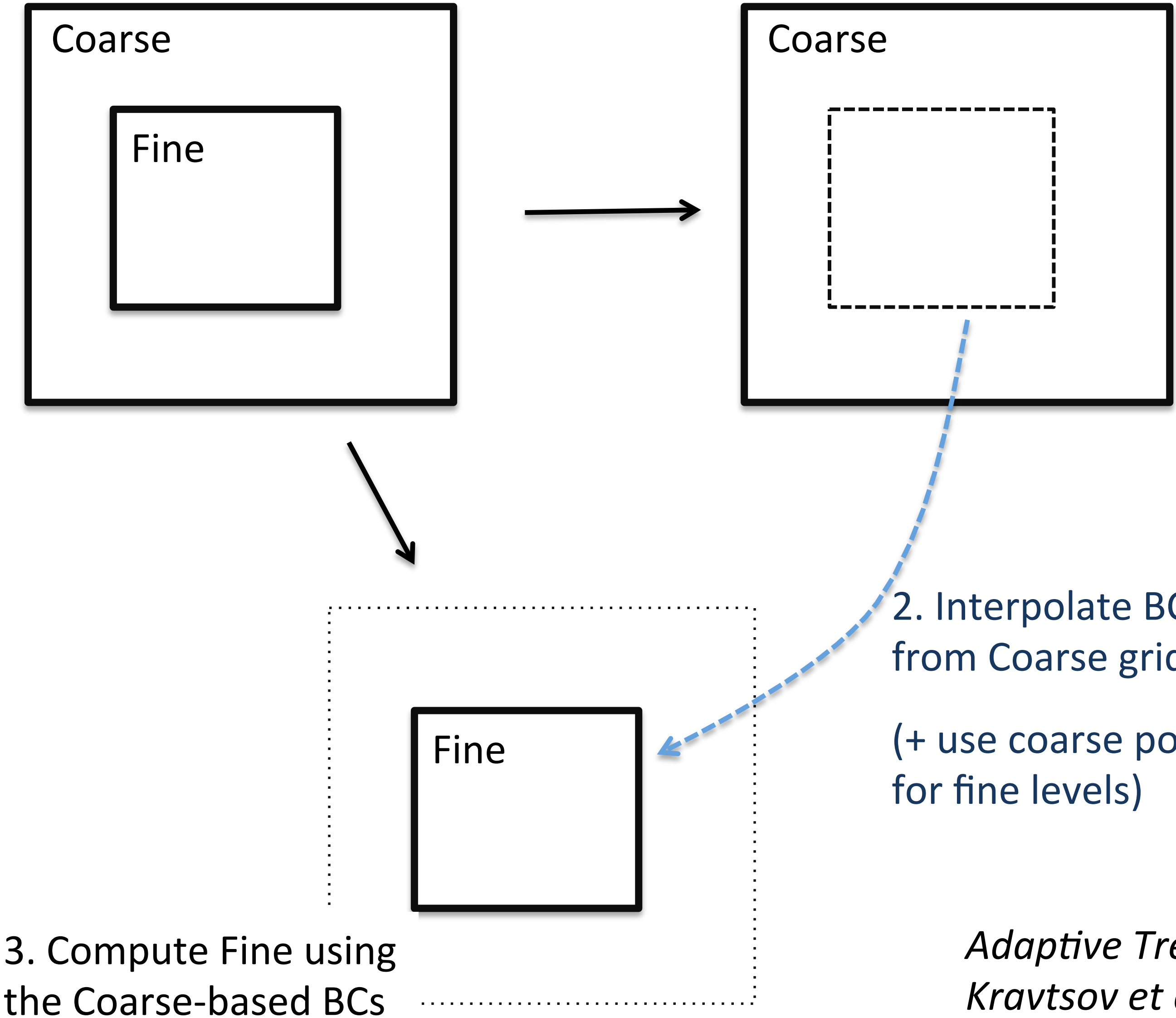
Coarse level potential computed independently :
 - using all particles (even those in Fine region)
 - applied to Coarse-region particles

Finel level potential is sum of:
 - coarse-only potential (computed everywhere with coarse-only particles)
 + fine-ony potential (Fine region as closed box)
 => applied to Fine-region particles

Always isolated B.C. -- requires simple boundaries (patches, single octs..)

PANDORA code – Villumsen et al. 1989 ApJS

AMR Poisson solver



1. Compute Coarse independently of refinements
(with any choosen BCs)

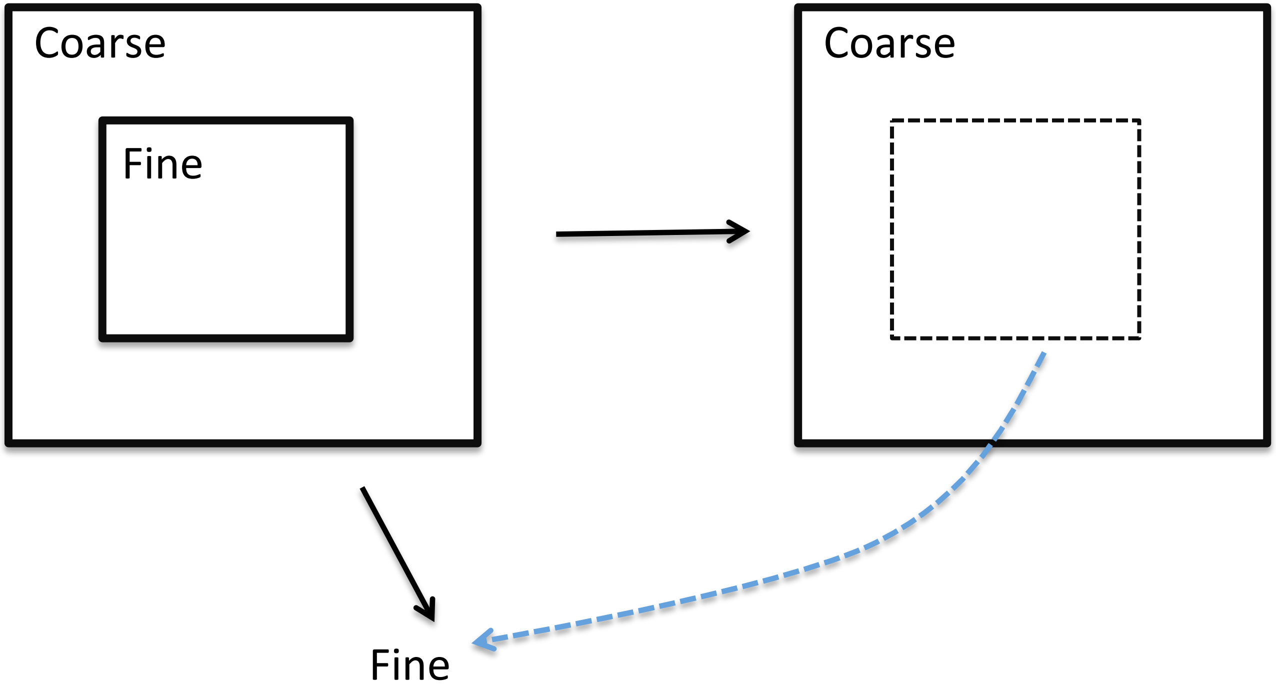
2. Interpolate BCs for fine level from Coarse grid result

(+ use coarse potential as guess for fine levels)

3. Compute Fine using the Coarse-based BCs

Adaptive Tree scheme
Kravtsov et al. 1997 ApJS (ART code)
Miniati & Collela JCP 227 (2007)

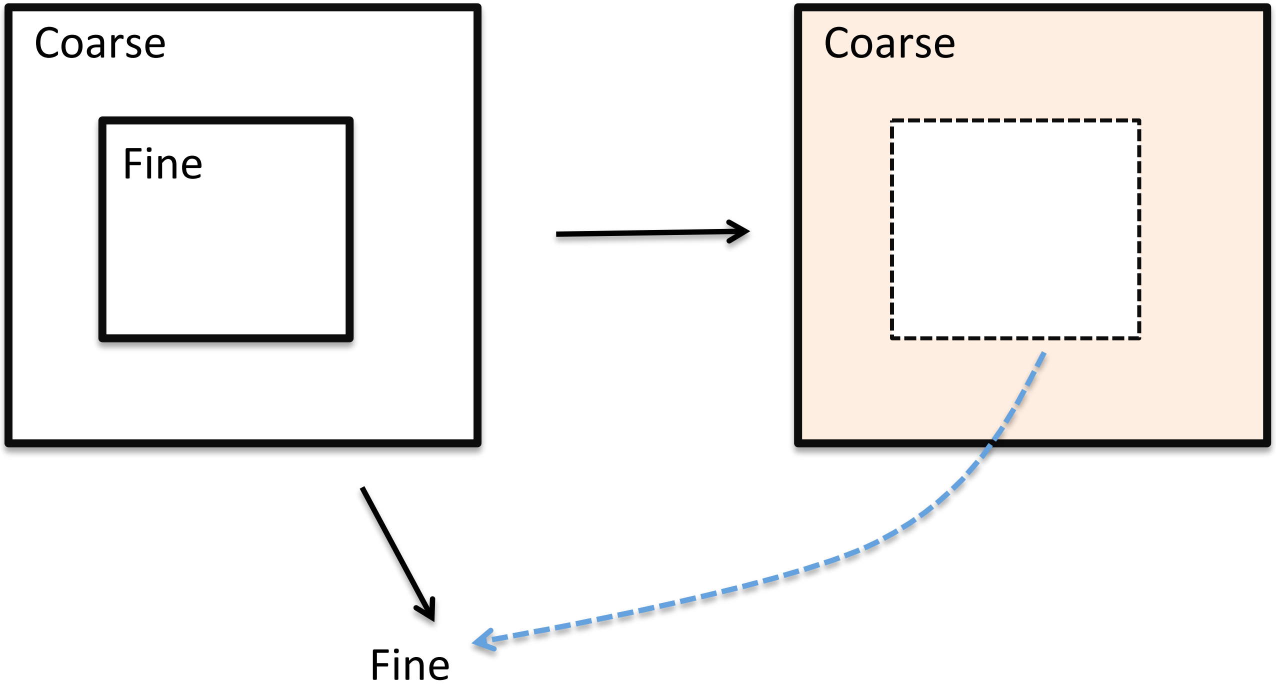
AMR Poisson solver with one-way interface



1. Compute Coarse independently of refinements
(with any choosen BCs)

In the original ART-based solvers, all particles (including fine ones) are passed to the coarse level to compute the coarse density, and the coarse potential is computed everywhere => extra calculations, the density and potential are computed twice in the » fine » volume.
Extra MPI communications too.

AMR Poisson solver with two-way interface



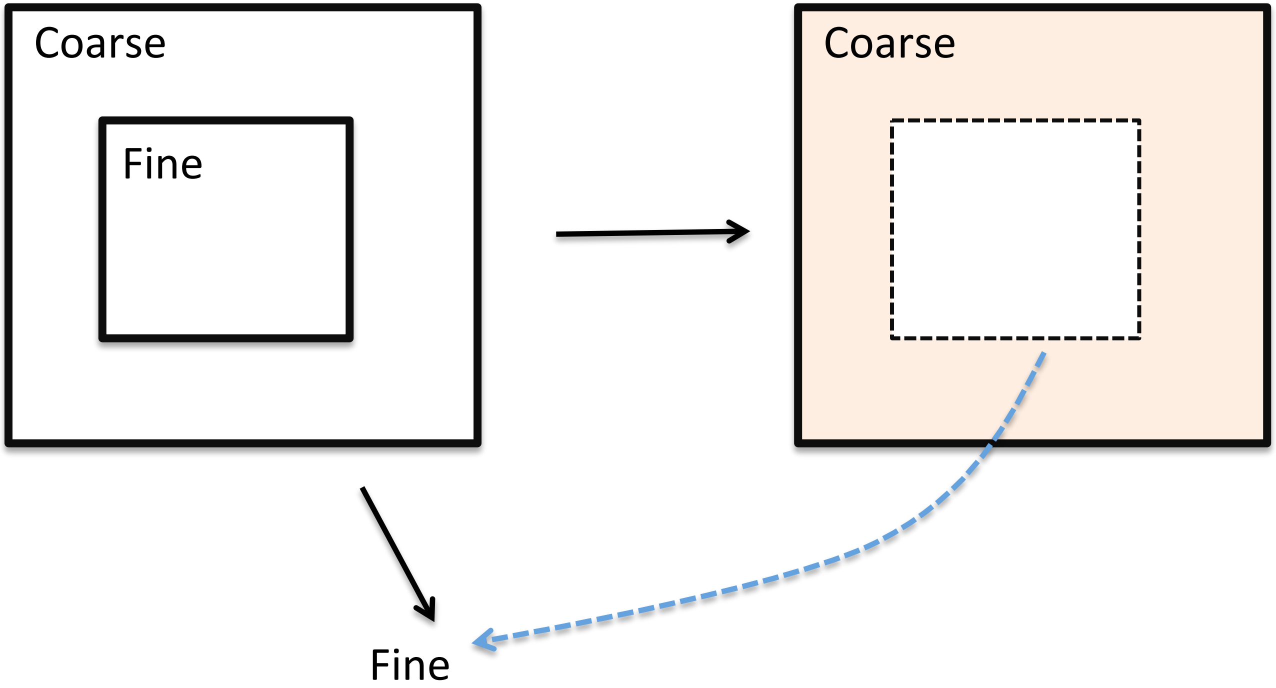
1. Compute Coarse independently of refinements
(with any choosen BCs)

Coarse density interpolation and potential calculation performed only in the « Coarse – Fine volume ».

Use *Coarse* as a boundary for *Fine*
But also *Fine* as an inner boundary for *Coarse*

Miniati & Collela JCP 227 (2007)

AMR Poisson solver with two-way interface



1. Compute Coarse independently of refinements (with any choosen BCs)

- Solve $\Delta\Phi_c=\rho_c$ on the « C-F » volume only
- Use as Dirichlet+Neuman B.C. at the boundary
- Solve $\Delta\Phi_F=\rho_F$ on the « F » volume
- Iterate until convergence at both the C-F and F levels.

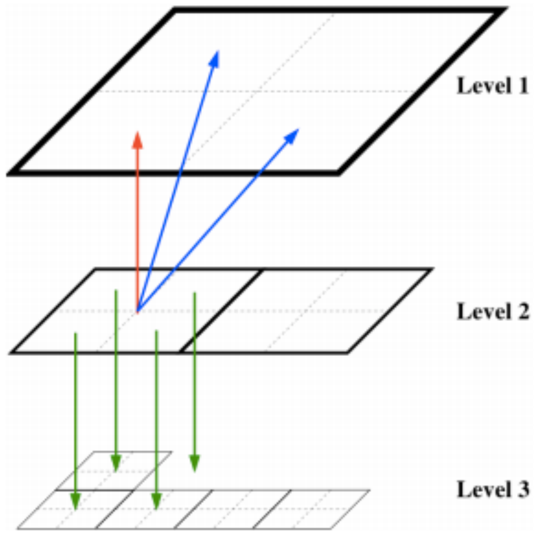
Miniati & Collela JCP 227 (2007)

Oct-tree structures for AMR (here RAMSES)

1 oct = small group of 2^{Ndim}

8 associated pointers:

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



Fully threaded tree (Khokhlov 1998)

Courtesy R. Teyssier

Any cell can be « leaf » (active) or « split » (inactive).

All levels exist from level 0 (full box) to the maximal level,
but a minimal refinement level (usually >0) is defined.

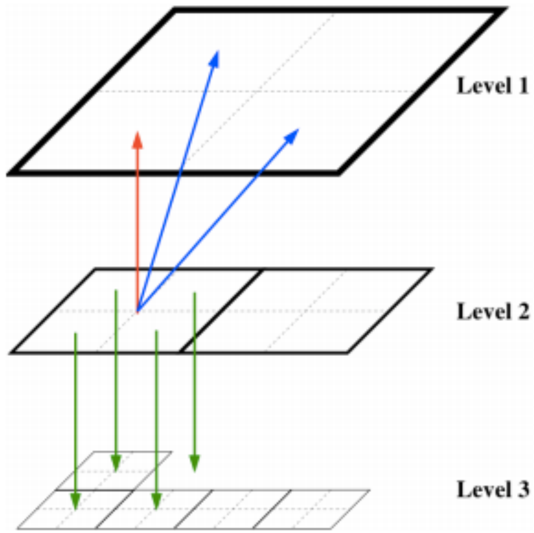
Levels can be sub-cycled or not (usually yes, see later CFL condition)

Oct-tree structures for AMR (here RAMSES)

1 oct = small group of 2^{Ndim}

8 associated pointers:

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

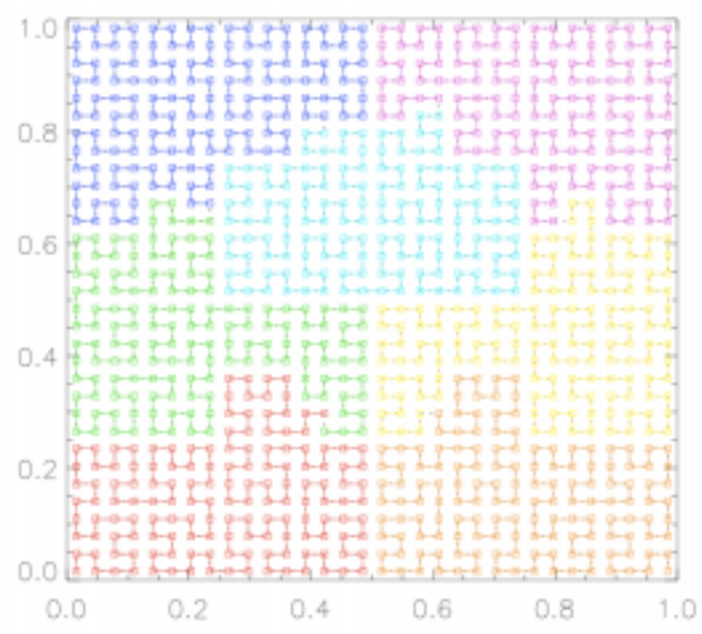


Courtesy R. Teyssier

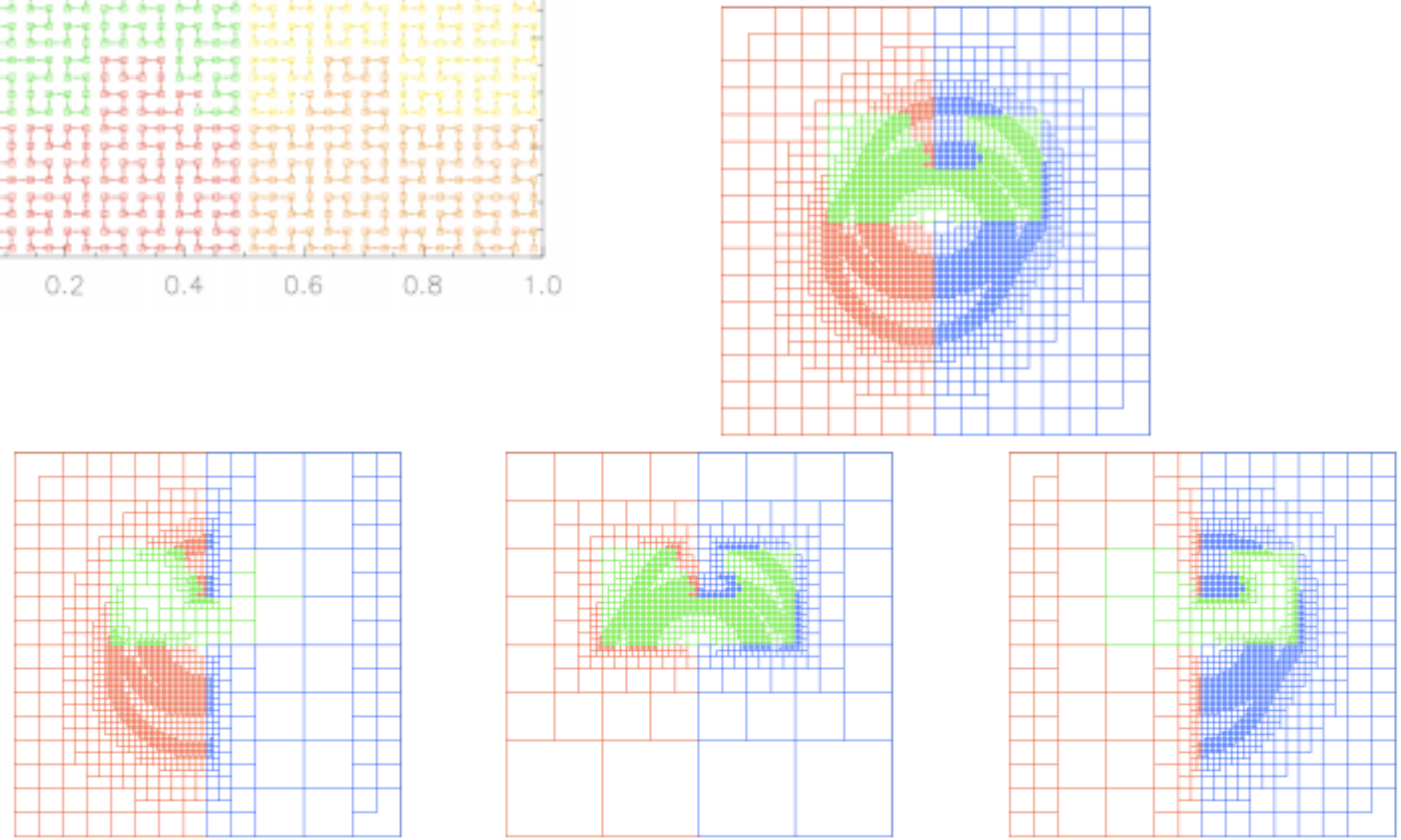
Refinement criteria:

- « constrained refinement » : no more than 1 level difference between neighbors
- Physical criteria : mass or number of particles per cell,
  physical scale-lengths to resolve, gradients, zoom regions…
- Mathematical smoothing to ensure convex enveloppe

Oct-tree and domain decomposition in MPI



- Space-filling curve (Peano-Hilbert)
- Ghost copies of the locally-essential tree in the memory of each CPU



Hydrodynamics : particle-based and grid-based

Now adding the gas (continuous fluid) in the system:

Back to the initial questions (Vlassov-Poisson) : particles or mass in grids ?

Mass in grid now has a reasonable memory cost : one single velocity in a given spatial resolution element => no need for a 6D grid.

<u>Grid-based:</u> One single velocity (+thermal dispersion) in each spatial cell

$\Rightarrow$ formally correct only for infinitely small cells,
or at least cells smaller than the dissipation scale(s) of the turbulence cascade

$\Rightarrow$ Otherwise gas is very dissipative/viscous or needs to be artificially heated

<u>Particles:</u> Allows a distribution of velocities inside the spatial resolution elements
No formal dissipation of the kinetic energy at the mesh resolution limit
But is this really modelling a continuous fluid ?

Hydrodynamics : particle-based and grid-based

Now adding the gas (continuous fluid) in the system:

Back to the initial questions (Vlassov-Poisson) : particles or mass in grids?

Mass in grid now has a reasonable memory cost : one single velocity in a given spatial resolution element => no need for a 6D grid.

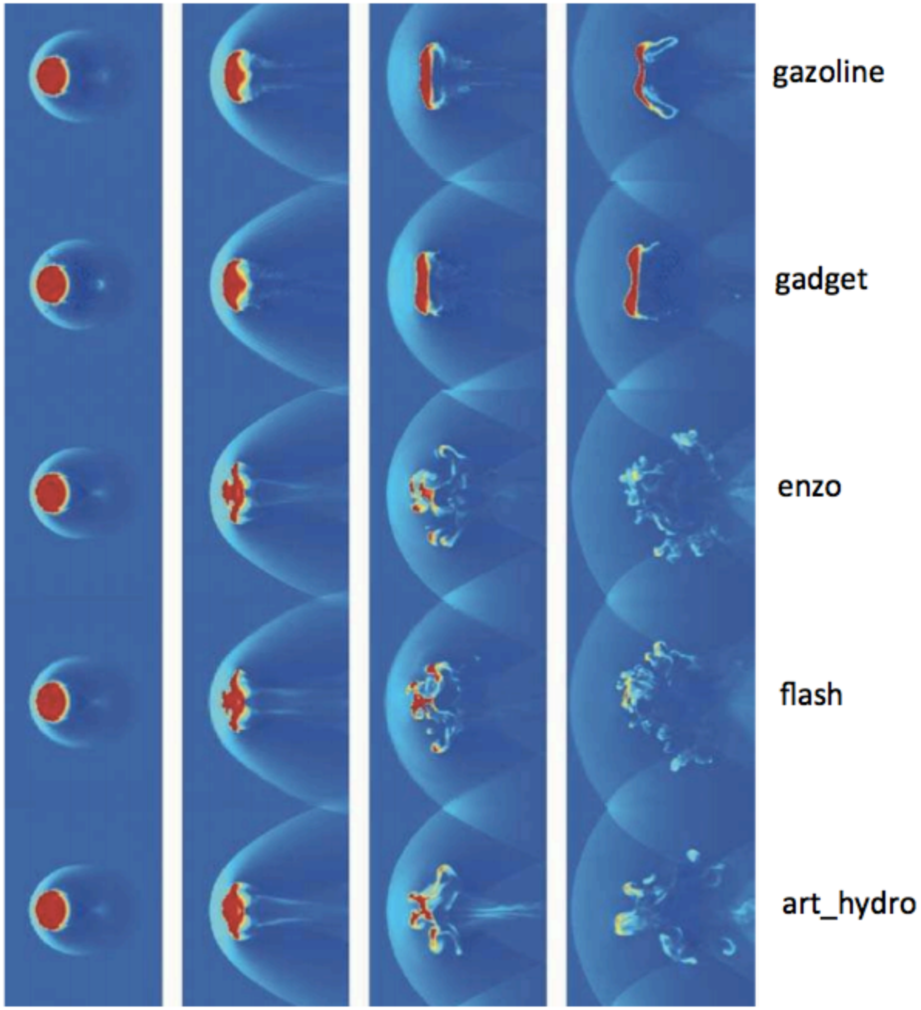
Grid-based: One single velocity (+thermal dispersion) in each spatial cell

⇒ formally correct only for infinitely small cells,
or at least cells smaller than the dissipation scale(s) of the turbulence cascade

⇒ Otherwise gas is very dissipative/viscous or needs to be artificially heated

Particles: Allows a distribution of velocities inside the spatial resolution elements
No formal dissipation of the kinetic energy at the mesh resolution limit
But is this really modelling a continuous fluid ?

Hydrodynamics : particle-based and grid-based

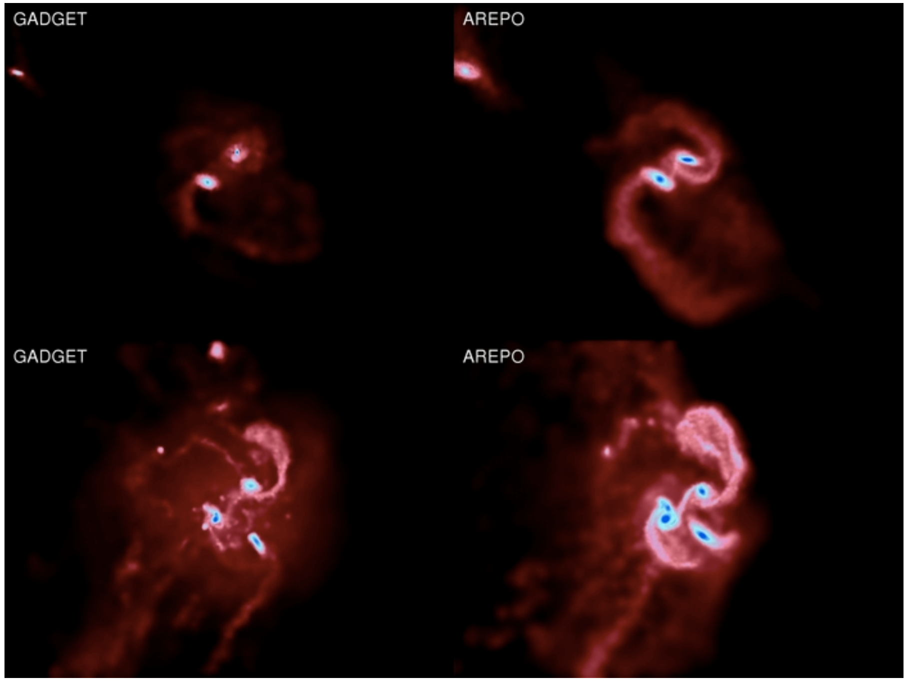


Agertz et al., Tasker et al. 2007

- Comparison a similar resolution
- SPH has been improved
- Is this really relevant at all if the critical scales (spatial and thermal) are not resolved in cosmo/galaxy/ISM simulations in practice ?
- Comparisons on a given computer can be **very** different from ideal comparison of hydro solvers !
- Most important may be the achievable spatial resolution, mass resolution, temperature floor (i.e. minimal Jeans mass) on a given computer => code resolution+**scaling**

=> it is probably where grids and AMR win the comparison in practice

Hydrodynamics : particle-based and grid-based

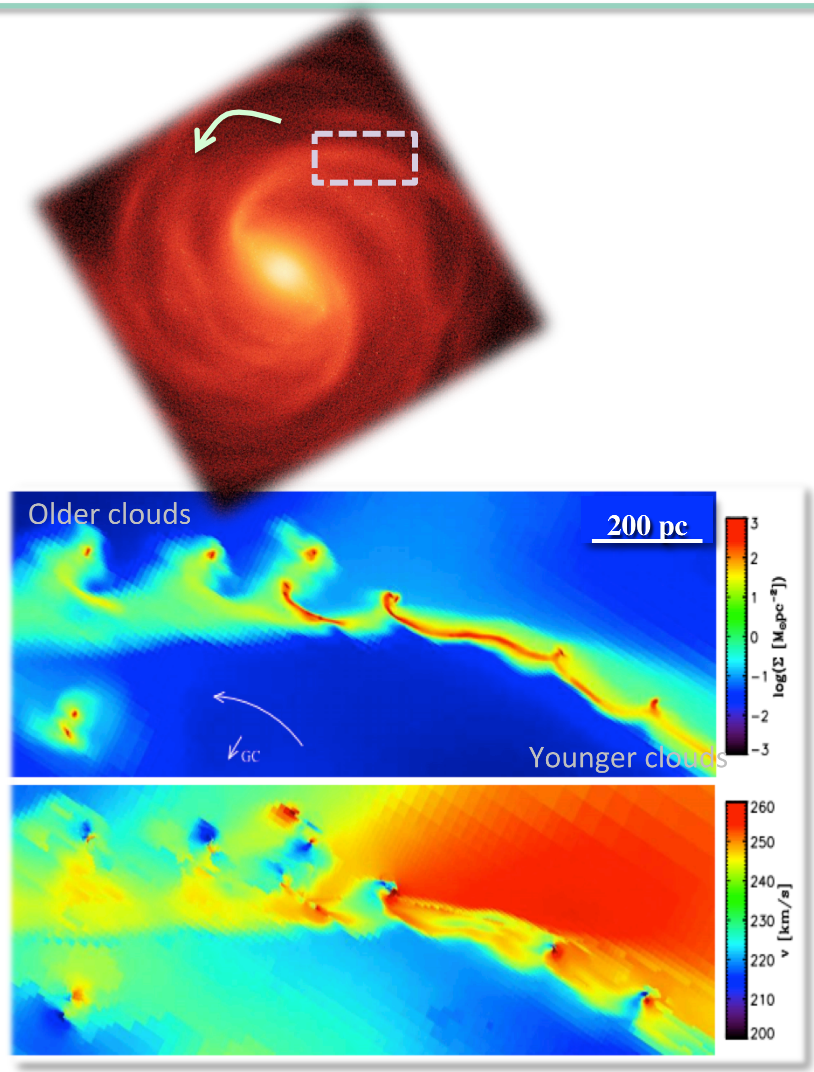


Keres, Springel et al.

- Comparison a similar resolution
- SPH has been improved
- Is this really relevant at all if the critical scales (spatial and thermal) are not resolved in cosmo/galaxy/ISM simulations in practice ?
- Comparisons on a given computer can be **very** different from ideal comparison of hydro solvers !
- Most important may be the achievable spatial resolution, mass resolution, temperature floor (i.e. minimal Jeans mass) on a given computer => code resolution+**scaling**

=> it is probably where grids and AMR win the comparison in practice

Hydrodynamics : particle-based and grid-based



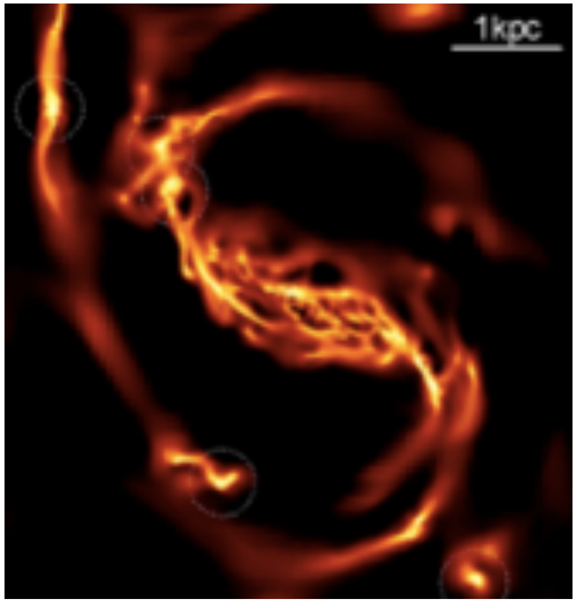
Need to resolve the Jeans+ KH instability scales before arguing how accurately the code handles it !

Renaud Bournaud et al. 2013 RAMSES

- Comparison a similar resolution
- SPH has been improved
- Is this really relevant at all if the critical scales (spatial and thermal) are not resolved in cosmo/galaxy/ISM simulations in practice ?
- Comparisons on a given computer can be **very** different from ideal comparison of hydro solvers !
- Most important may be the achievable spatial resolution, mass resolution, temperature floor (i.e. minimal Jeans mass) on a given computer => code resolution+**scaling**

=> it is probably where grids and AMR win the comparison in practice

Astrophysical fluid dynamics:
N-body, gravity and hydrodynamics
Applications in cosmology, galaxy and star formation

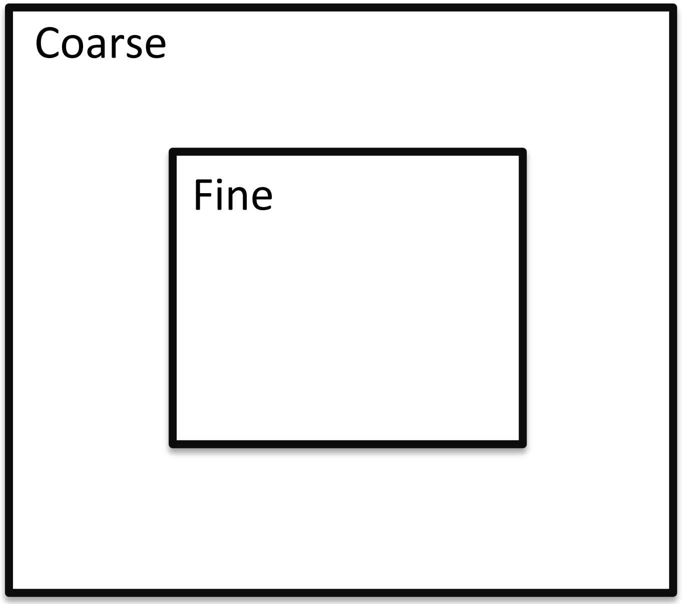


Frédéric Bournaud – CEA Saclay
frederic.bournaud@cea.fr

AMR Poisson solver

- Relaxation methods can still be used (such as Conjugate Gradient)
- Multi-grid could still be used on any AMR level (esp. if patch-based),
  but generally in tree-based schemes the multi-grid becomes inherent
  to the AMR structure

Basic approach : « Pandora » scheme. Coarse levels ignore the content of fine levels



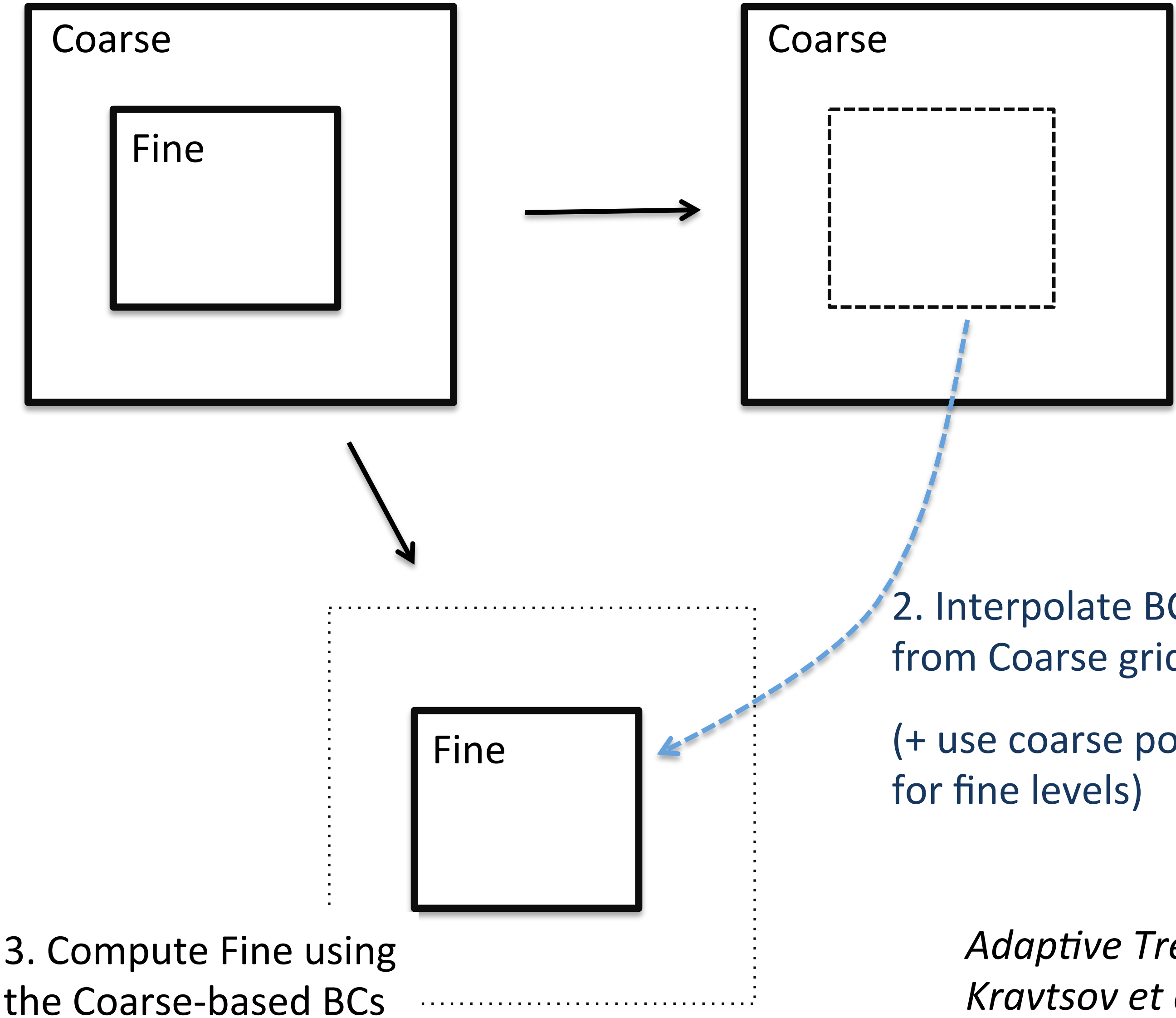
PANDORA code – Villumsen et al. 1989 ApJS

Coarse level potential computed independently :
- using all particles (even those in Fine region)
- applied to Coarse-region particles

Finel level potential is sum of:
- coarse-only potential (computed everywhere with coarse-only particles)
+ fine-ony potential (Fine region as closed box)
=> applied to Fine-region particles

Always isolated B.C. -- requires simple boundaries (patches, single octs..)

AMR Poisson solver



1. Compute Coarse independently of refinements
(with any choosen BCs)

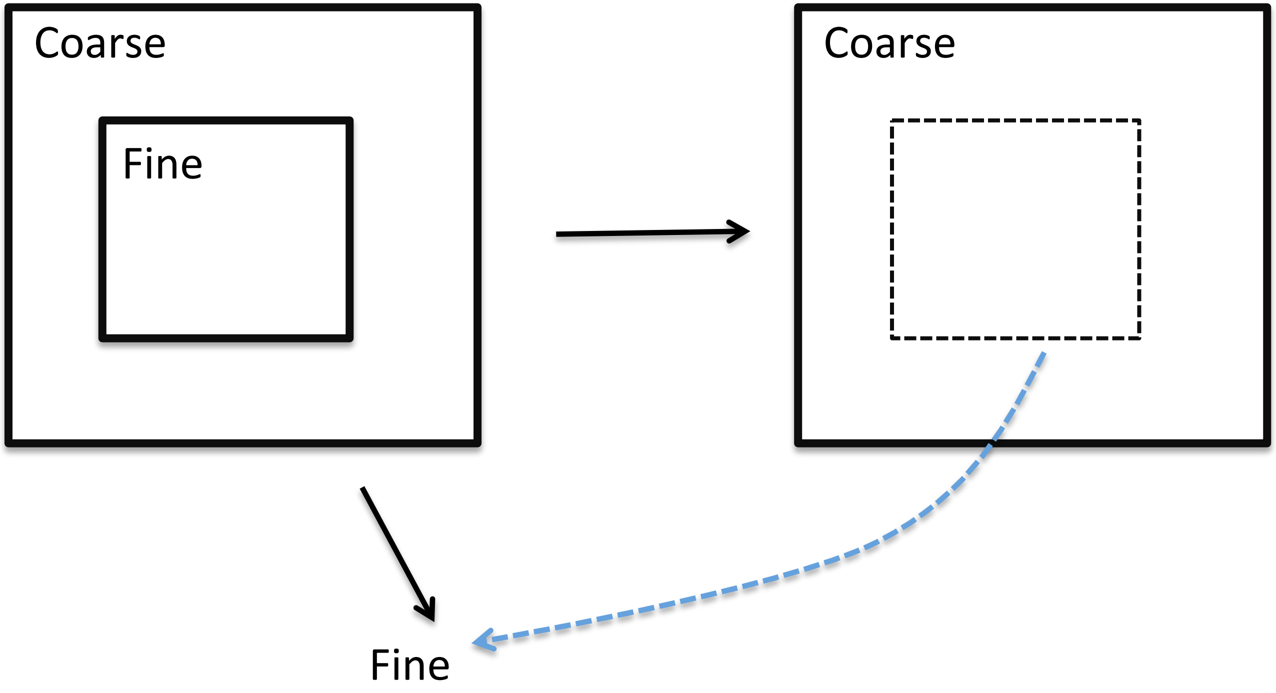
2. Interpolate BCs for fine level from Coarse grid result

(+ use coarse potential as guess for fine levels)

3. Compute Fine using the Coarse-based BCs

Adaptive Tree scheme
Kravtsov et al. 1997 ApJS (ART code)
Miniati & Collela JCP 227 (2007)

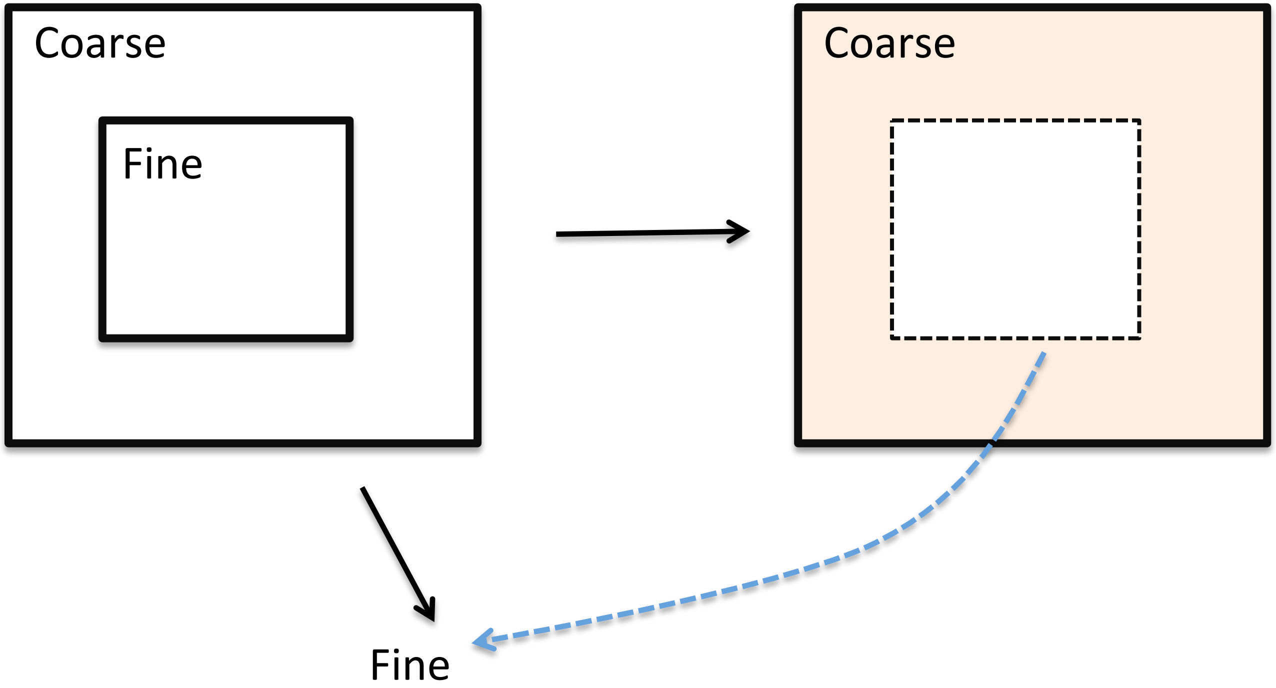
AMR Poisson solver with one-way interface



1. Compute Coarse independently of refinements (with any choosen BCs)

In the original ART-based solvers, all particles (including fine ones) are passed to the coarse level to compute the coarse density, and the coarse potential is computed everywhere => extra calculations, the density and potential are computed twice in the » fine » volume.
Extra MPI communications too.

AMR Poisson solver with two-way interface



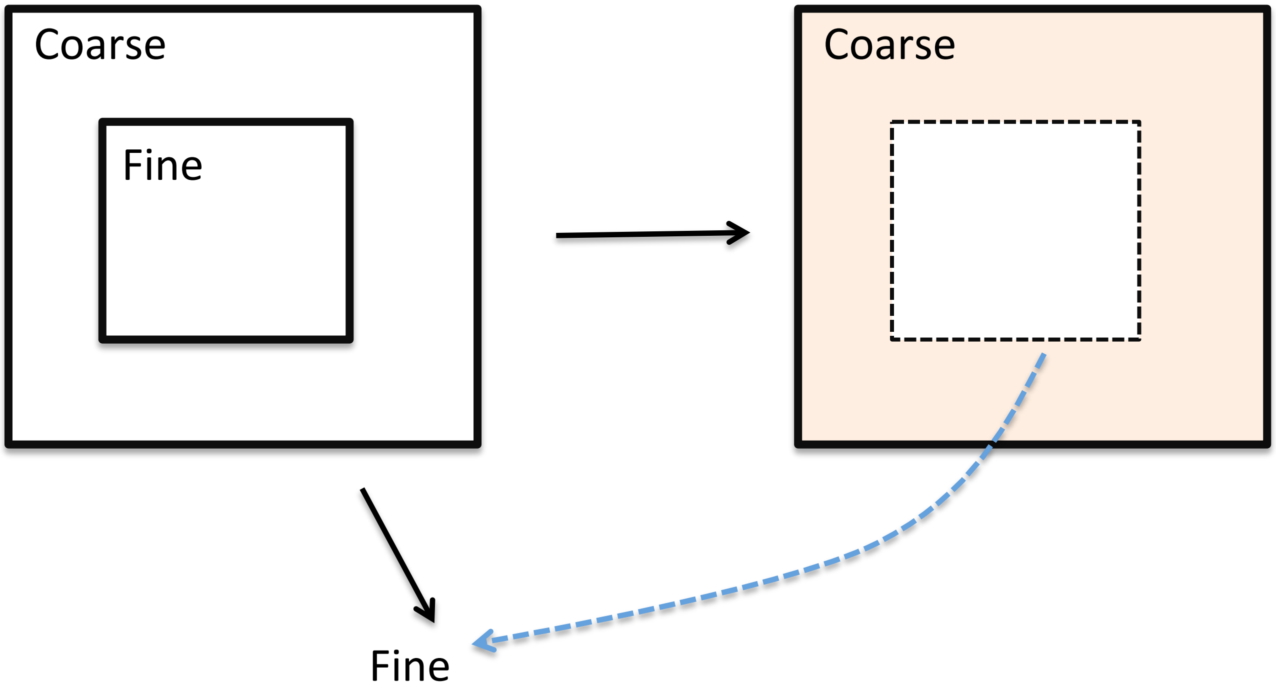
1. Compute Coarse independently of refinements
(with any choosen BCs)

Coarse density interpolation and potential calculation performed only in the « Coarse – Fine volume ».

Use *Coarse* as a boundary for *Fine*
But also *Fine* as an inner boundary for *Coarse*

Miniati & Collela JCP 227 (2007)

AMR Poisson solver with two-way interface

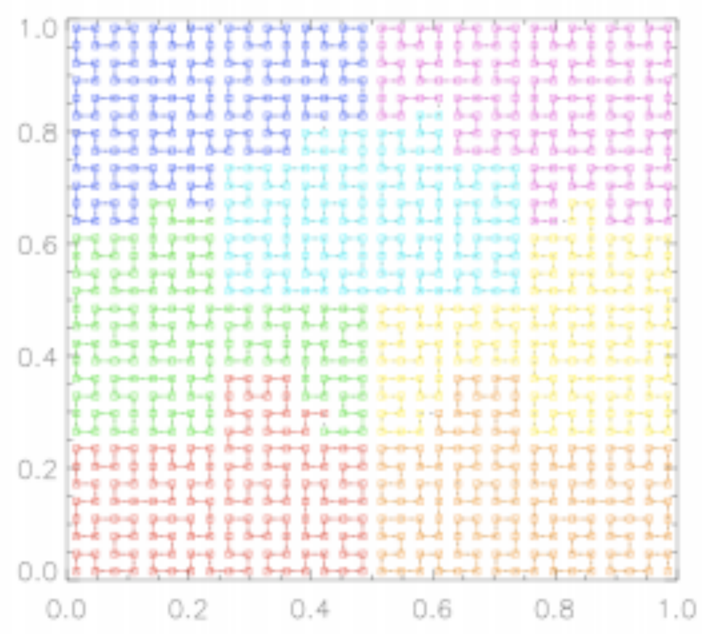


1. Compute Coarse independently of refinements
(with any choosen BCs)

- Solve $\Delta\Phi_c=\rho_c$ on the « C-F » volume only
- Use as Dirichlet+Neuman B.C. at the boundary
- Solve $\Delta\Phi_F=\rho_F$ on the « F » volume
- Iterate until convergence at both the C-F and F levels.

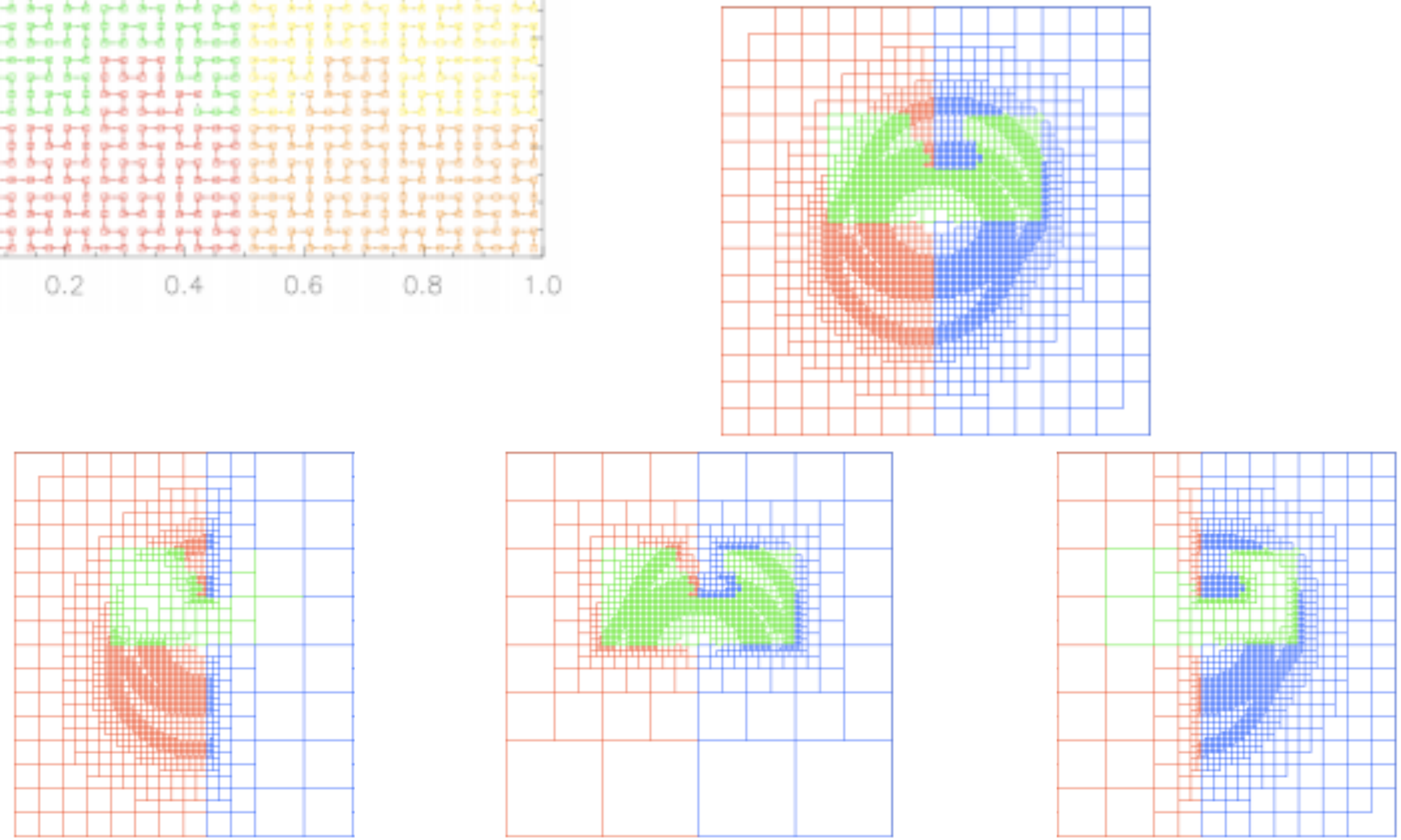
Miniati & Collela JCP 227 (2007)

Oct-tree and domain decomposition in MPI



- Space-filling curve (Peano-Hilbert)

- Ghost copies of the locally-essential tree in the memory of each CPU



Hydrodynamics : particle-based and grid-based

Now adding the gas (continuous fluid) in the system:

Back to the initial questions (Vlassov-Poisson) : particles or mass in grids?

Mass in grid now has a reasonable memory cost : one single velocity in a given spatial resolution element => no need for a 6D grid.

Grid-based: One single velocity (+thermal dispersion) in each spatial cell

$\Rightarrow$ formally correct only for infinitely small cells,
or at least cells smaller than the dissipation scale(s) of the turbulence cascade

$\Rightarrow$ Otherwise gas is very dissipative/viscous or needs to be artificially heated

Particles: Allows a distribution of velocities inside the spatial resolution elements
No formal dissipation of the kinetic energy at the mesh resolution limit
But is this really modelling a continuous fluid ?

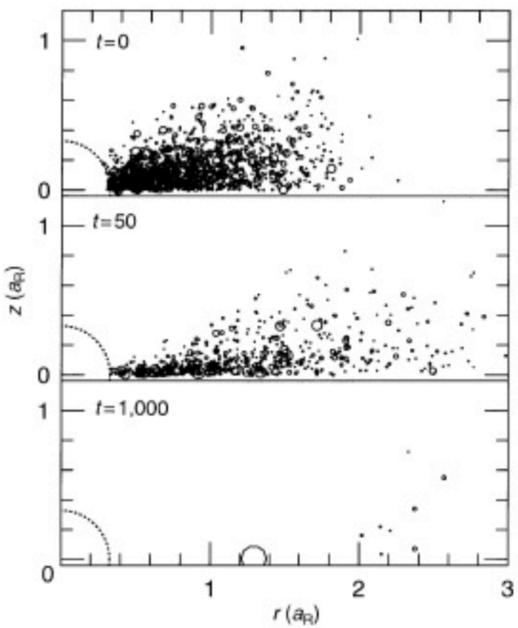
Beyond Vlassov-Poisson : N-body with collisions

- Particles with finite sizes and collisional particle-particle interactions
- « Sticky Particle » schemes, search for collisions between neighbours
  The PM grid can be directly used for this.
- Used for the cold turbulent ISM phases when not able to resolve the injection scale
- Simplest schemes : particles bouncing back with <1 restitution coefficient
- Accurate model of solid particle disks :
  planetary systems, planetary rings, silicate-phase proto-planetary disks
- Modeling of silicate bodies interactions (Salo, 1992, 1998) :
  - Finite-size bodies
  - Elasticity + friction

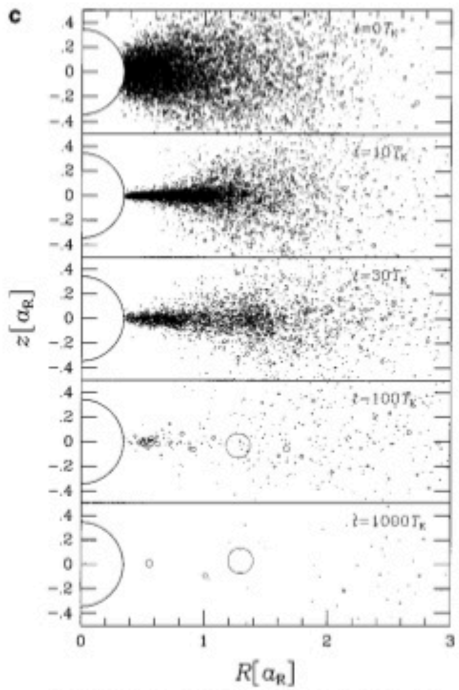
Beyond Vlassov-Poisson : N-body with collisions

Proto-lunar disk and Moon accretion
after a giant impact.

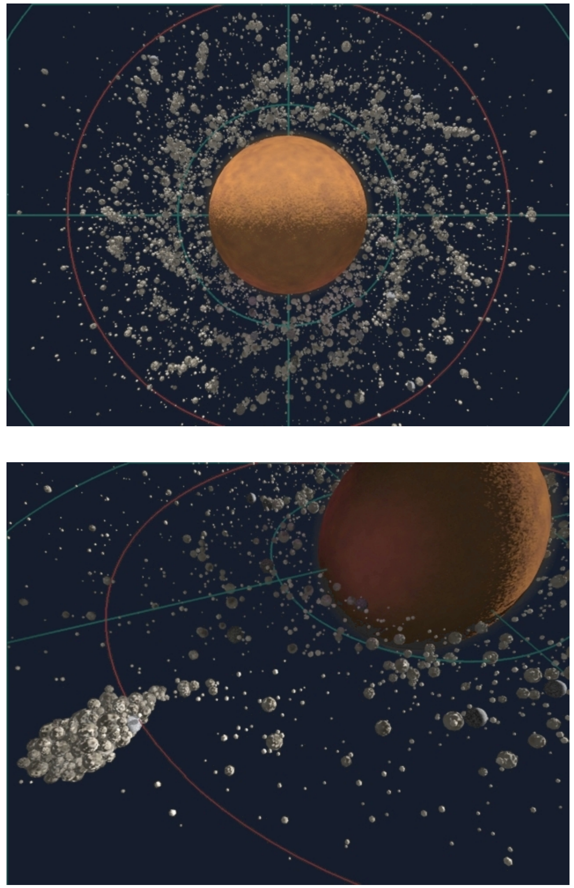
Self-gravity (PM) => density wave => transport
Accretion (sitcky part.) at the Roche radius.



[Ida et al., *Nature*, 1997]



[Kokubo et al., *Icarus*, 2000]



Kokubo, Ida et al. 2010

Hydrodynamics : particle-based and grid-based

Now adding the gas (continuous fluid) in the system:

Back to the initial questions (Vlassov-Poisson) : particles or mass in grids?

Mass in grid now has a reasonable memory cost : one single velocity in a given spatial resolution element => no need for a 6D grid.

Grid-based: One single velocity (+thermal dispersion) in each spatial cell

⇒ formally correct only for infinitely small cells,
or at least cells smaller than the dissipation scale(s) of the turbulence cascade

⇒ Otherwise gas is very dissipative/viscous or needs to be artificially heated

Particles: Allows a distribution of velocities inside the spatial resolution elements
No formal dissipation of the kinetic energy at the mesh resolution limit
But is this really modelling a continuous fluid ?

Hydrodynamics : CFL condition

Courant number $C = \Delta t\ (v_x / \Delta x + v_y / \Delta y + v_z / \Delta z)$
for the simplest case of a pressure-less system.

Numerical stability requires $C<1$, usually $C<1/2$

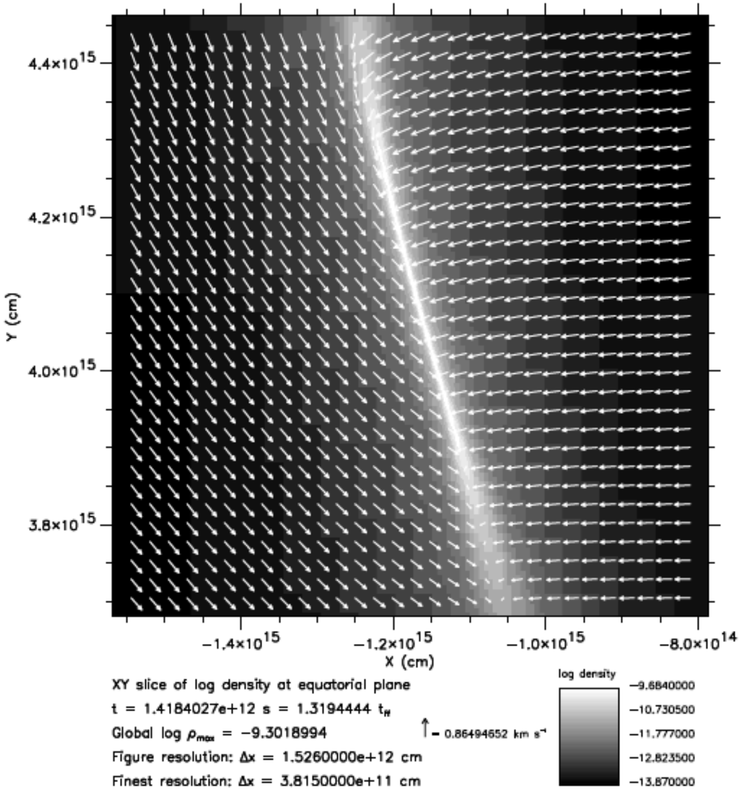
In the case of AMR, if the largest velocities are macroscopic (non thermal), sub-cycling of the levels ensures the same Courant number at all levels. In practice sub-cycling should not be used over more than 6-7 levels for best performance.

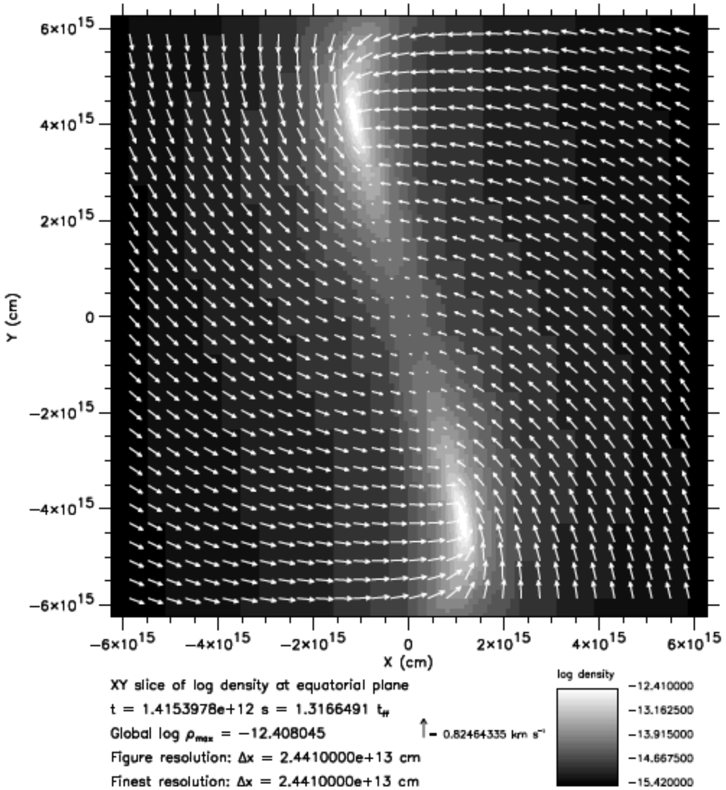
Difficulties can arise from the finite size of the coarsest level, especially with thermal processes at high T:

- High sound speed (for instance gas at >10^8K)
- High velcities from thermal release in low-density gas (feedback)

$\Rightarrow$ peuso-isolated systems are more difficult

Hydrodynamics : « Truelove criterion »

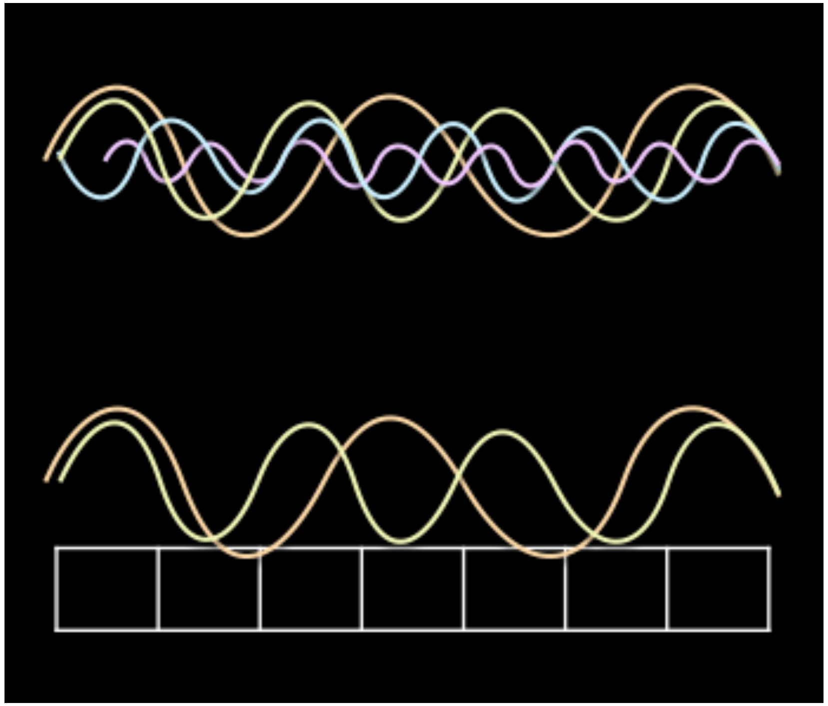




Problems of systems that should be stable, or at least should not fragment (smaller than their own Jeans length/mass).
Numerical fragmentation if the Jeans length is not resolved with a few (~ 4) resolution elements

Truelove et al. 1997, 1998
Bates & Burkert 1998

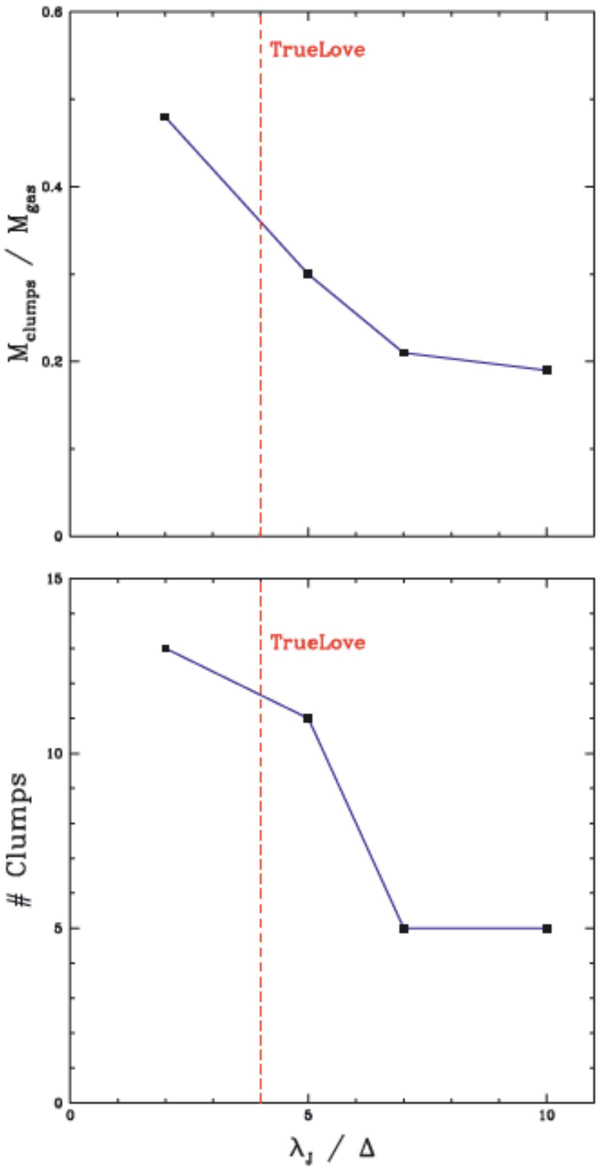
Hydrodynamics : « Truelove criterion »

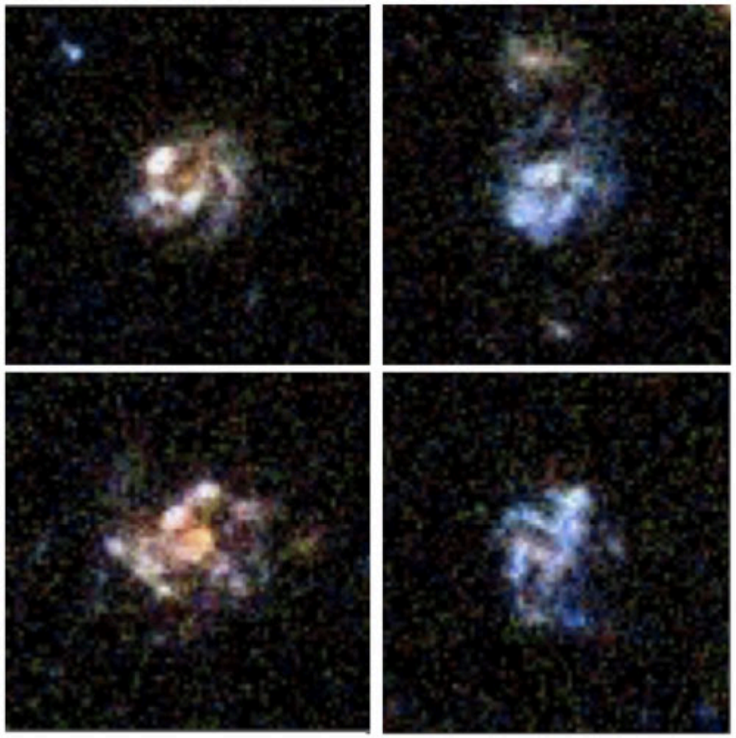


Problems of systems that should be stable, or at least should not fragment (smaller than their own Jeans length/mass).
Numerical fragmentation if the Jeans length is not resolved with a few (~ 4) resolution elements

Truelove et al. 1997, 1998
Bates & Burkert 1998

Hydrodynamics : « Truelove criterion »

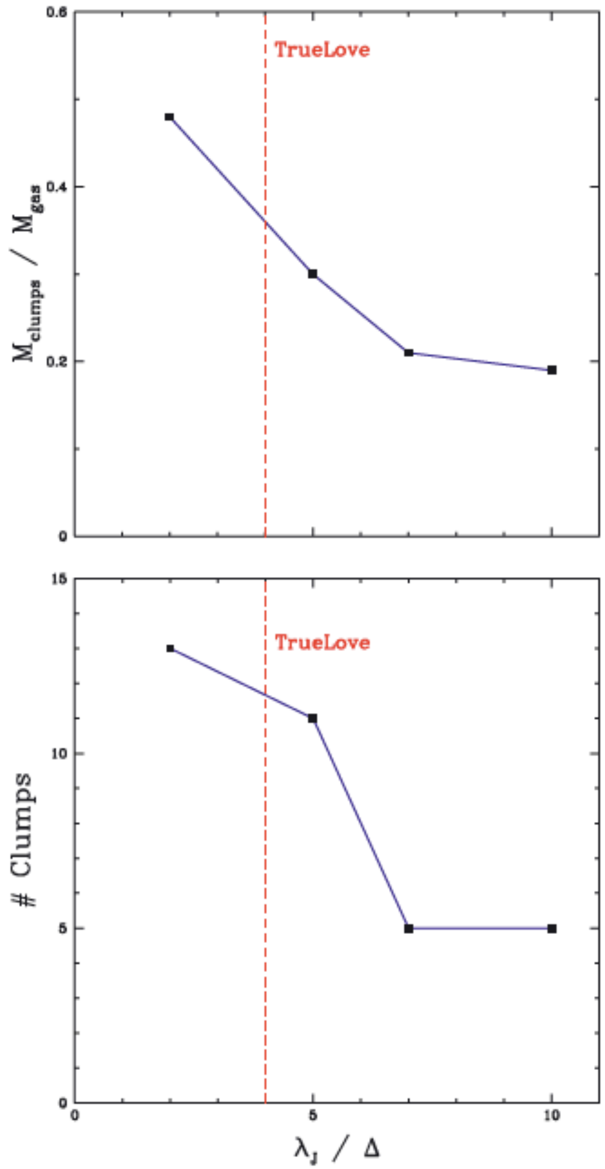


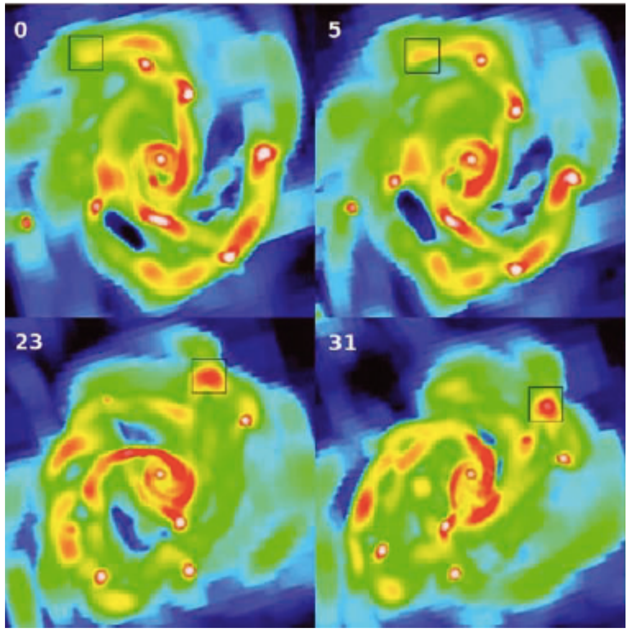


The situation is more complex for systems that should (sometimes) be Jeans-unstable.
Examples : ISM, molecular clouds, galactic disks…
Numerical fragmentation vs. artifical stabilisation ?

Ceverino et al. 2010, 2012

Hydrodynamics : « Truelove criterion »

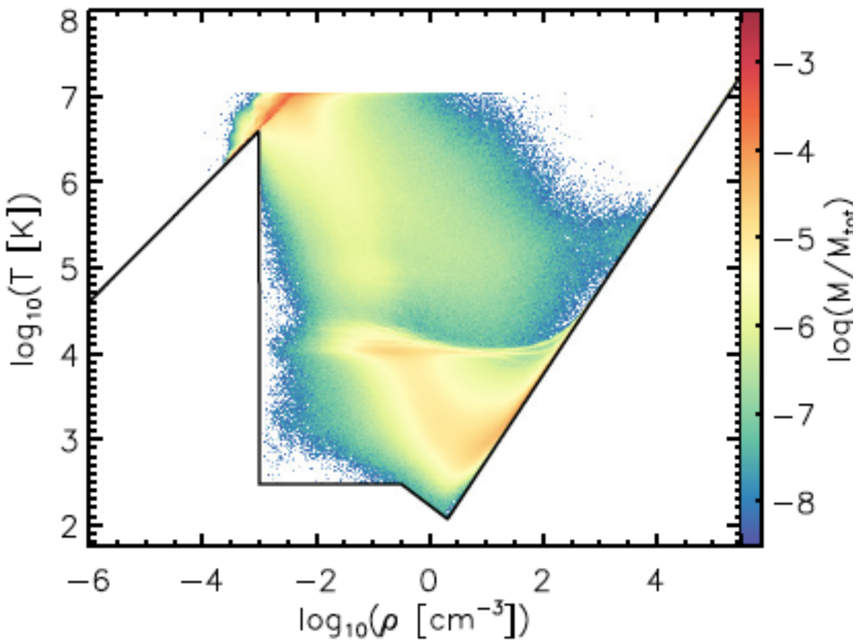


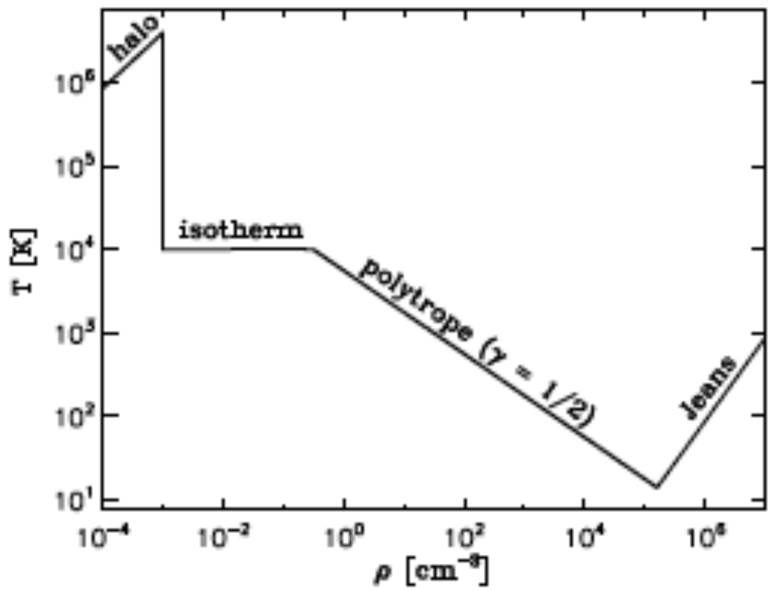


The situation is more complex for systems that should (sometimes) be Jeans-unstable.
Examples : ISM, molecular clouds, galactic disks…
Numerical fragmentation vs. artifical stabilisation ?

Ceverino et al. 2010, 2012

Hydrodynamics : Jeans-stability conditions





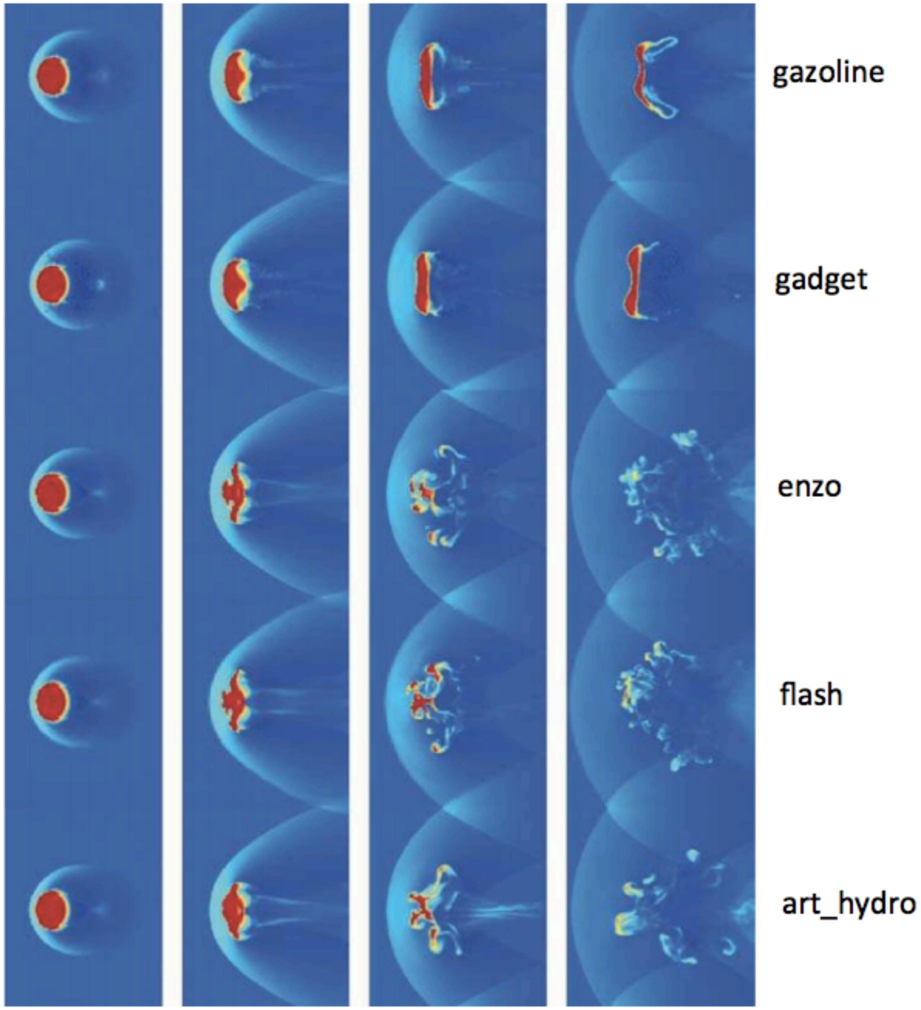
Typical temperature or pressure Floor to keep the Jeans-length resolved in ISM/galaxy simulations

Finest AMR resolution is 10 pc (left) and 0.1pc (right)

Turbulence rapidly biased to sub-sonic above not-so-high densities (300cm^{-3} at 10pc)
⇒ Lack of strong compression above these densities.
⇒ No « molecular » gas below 10pc resolution

Teyssier Chapon & Bournaud 2010, Perret+13, Renaud+14

Hydrodynamics : particle-based and grid-based

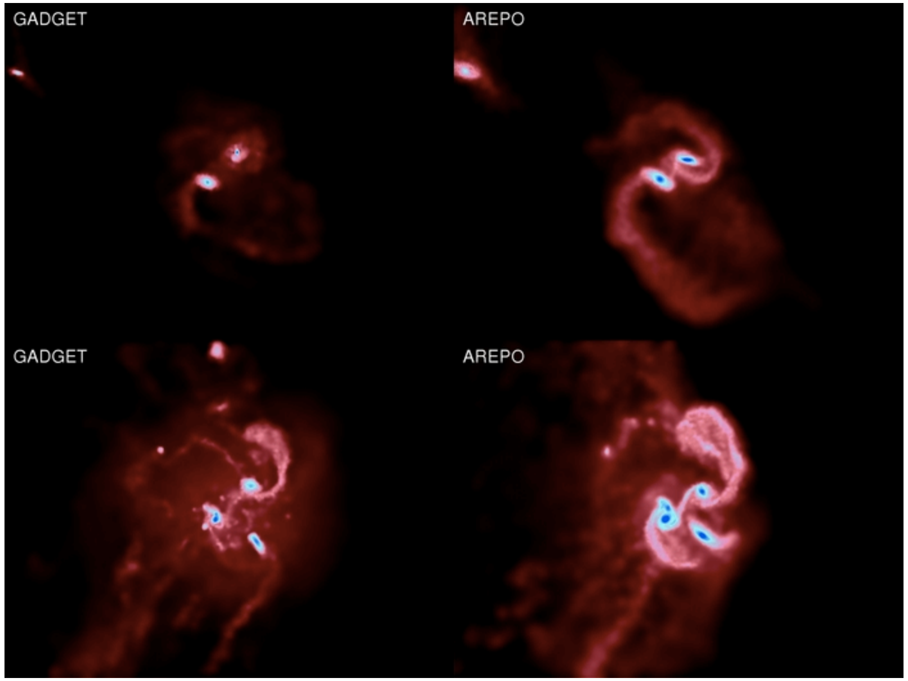


Agertz et al., Tasker et al. 2007

- Comparison a similar resolution
- SPH has been improved
- Is this really relevant at all if the critical scales (spatial and thermal) are not resolved in cosmo/galaxy/ISM simulations in practice ?
- Comparisons on a given computer can be **very** different from ideal comparison of hydro solvers !
- Most important may be the achievable spatial resolution, mass resolution, temperature floor (i.e. minimal Jeans mass) on a given computer => code resolution+**scaling**

=> it is probably where grids and AMR win the comparison in practice

Hydrodynamics : particle-based and grid-based

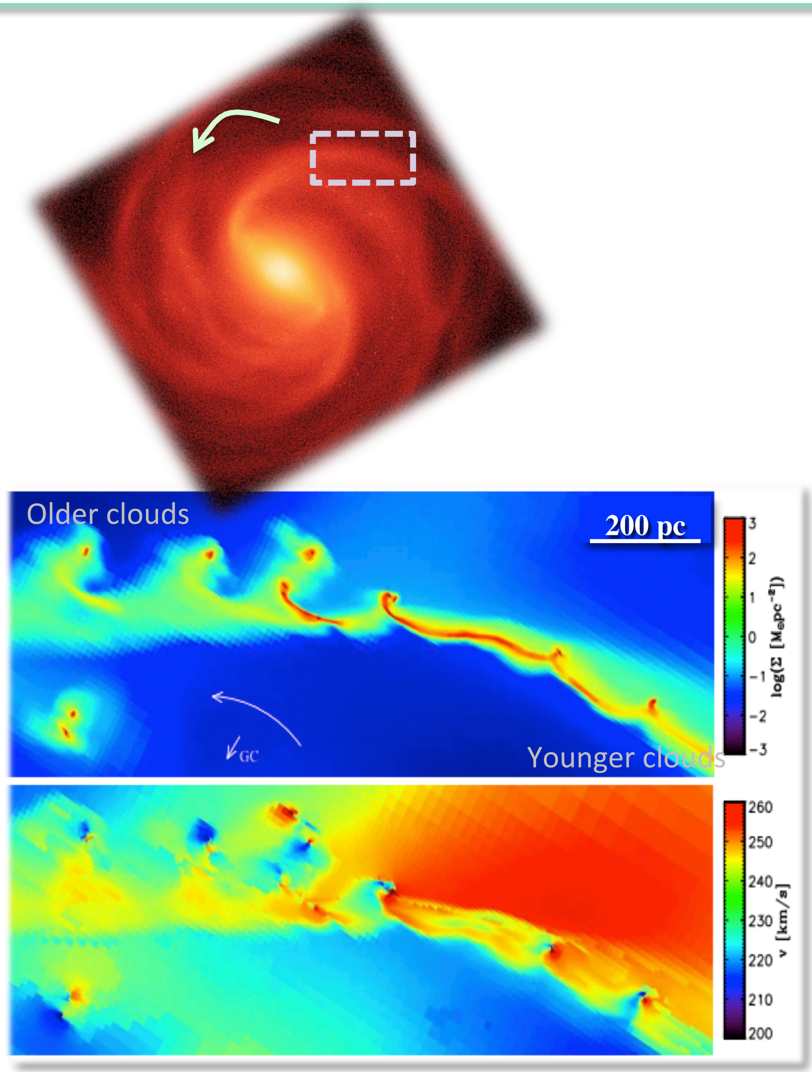


Keres, Springel et al.

- Comparison a similar resolution
- SPH has been improved
- Is this really relevant at all if the critical scales (spatial and thermal) are not resolved in cosmo/galaxy/ISM simulations in practice ?
- Comparisons on a given computer can be **very** different from ideal comparison of hydro solvers !
- Most important may be the achievable spatial resolution, mass resolution, temperature floor (i.e. minimal Jeans mass) on a given computer => code resolution+**scaling**

=> it is probably where grids and AMR win the comparison in practice

Hydrodynamics : particle-based and grid-based



Need to resolve the Jeans+ KH instability scales before arguing how accurately the code handles it !

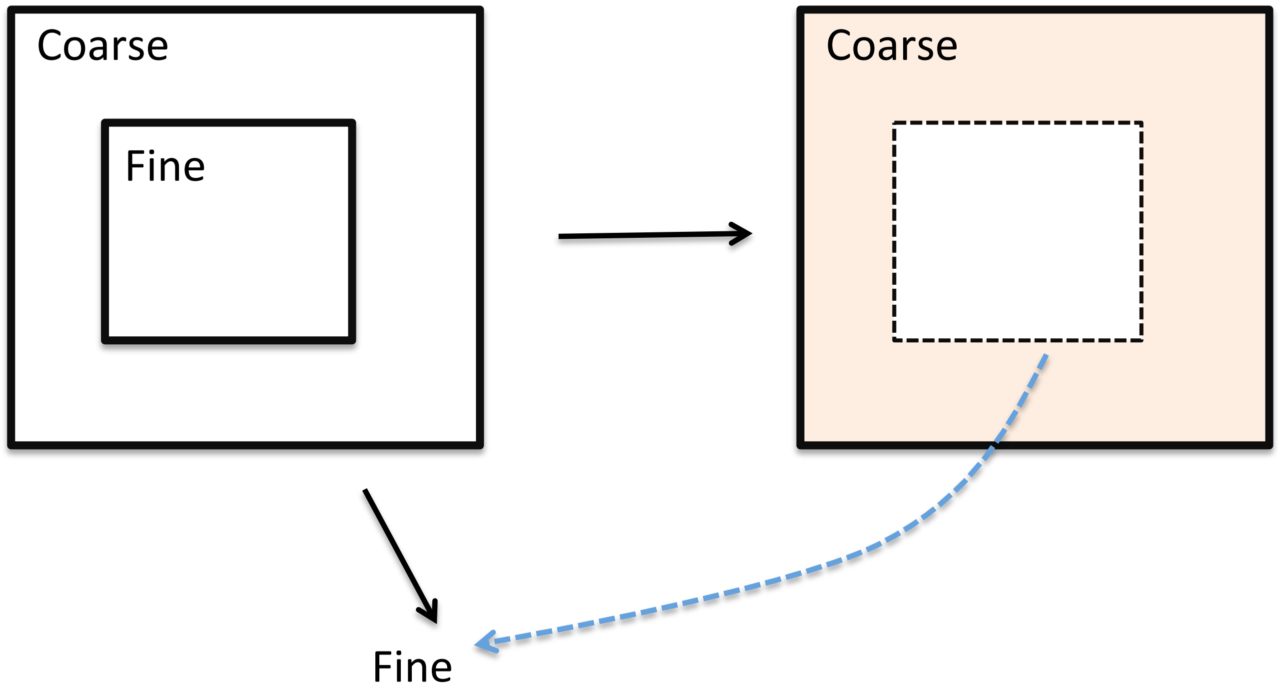
Renaud Bournaud et al. 2013 RAMSES

- Comparison a similar resolution
- SPH has been improved
- Is this really relevant at all if the critical scales (spatial and thermal) are not resolved in cosmo/galaxy/ISM simulations in practice ?
- Comparisons on a given computer can be **very** different from ideal comparison of hydro solvers !
- Most important may be the achievable spatial resolution, mass resolution, temperature floor (i.e. minimal Jeans mass) on a given computer => code resolution+**scaling**

=> it is probably where grids and AMR win the comparison in practice

Poisson solvers on an oct-tree AMR :

The refined regions do not need to « know » where they are, and how many of them there are – just use a multi-grid approach applied to any coarse/fine cell encountered along the space-filling curve.



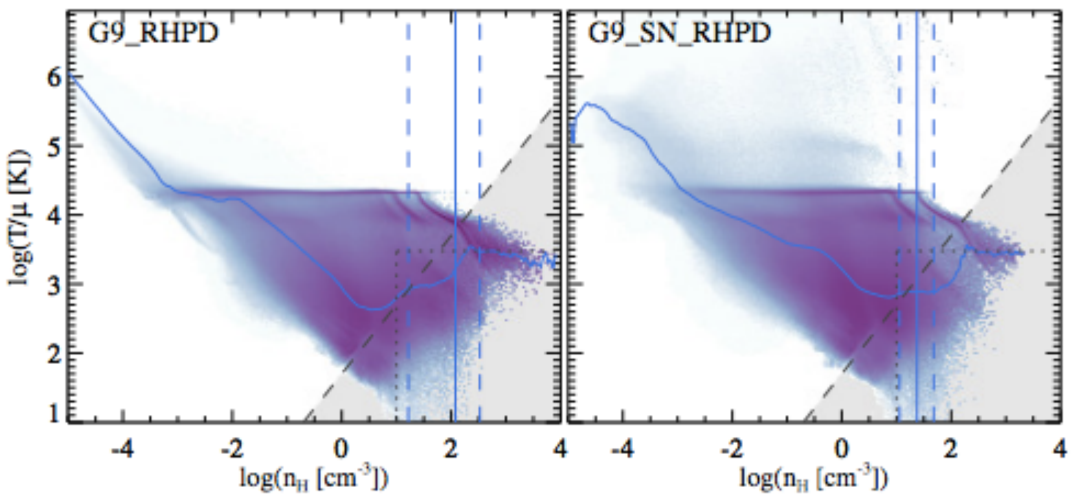
- Solve $\Delta\Phi_c=\rho_c$ on the « C-F » volume only
- Use as Dirichlet+Neuman B.C. at the boundary
- Solve $\Delta\Phi_F=\rho_F$ on the « F » volume
- Iterate until convergence at both the C-F and F levels.

Poisson solvers on an oct-tree AMR :

The refined regions do not need to « know » where they are, and how many of them there are – just use a multi-grid approach applied to any coarse/fine cell encountered along the space-filling curve.

Jeans-based heating at high density and energy conservation :

If a pressure floor is added in the Euler equation, the future temperature/energy are not affected once the density decreases (in the most elegant versions)



Poisson solvers on an oct-tree AMR :

The refined regions do not need to « know » where they are, and how many of them there are – just use a multi-grid approach applied to any coarse/fine cell encountered along the space-filling curve.

Jeans-based heating at high density and energy conservation :

If a pressure floor is added in the Euler equation, the future temperature/energy are not affected once the density decreases (in the most elegant versions)

Exercices not done :

2D hydro code in fortran, Godunov scheme, ghost cells for (reflecting) boundaries.

To be parallelized (MPI) along one axis.
=> code in TP_hydro/Mono/src , solution in .MPI/src directory
test with one hot point in a corner of the box.

Key : - processors do not need to know where their physical domain is (no coordinates)
- better to sychronize the MPI send and receive instructions
- buffers for hydro variables work just like boundary conditions.

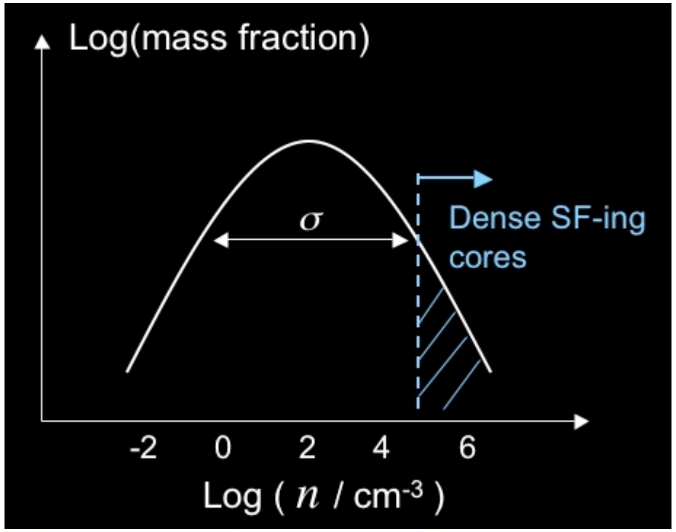
Sub-grid star formation models

- Star formation is known to be tighly correlated to dense gas (on parsec scale and larger scales)

- In the case of a turbulent, cold ISM phase, a large density PDF is produced

- The densest gas is converted into stars with a (very) arbitrary scheme. But the regulation is the dense gas rate production.

- In the ISM gravity takes over turbulent pressure at about 1-2x10^4 cm^{-3} (Elmegreen2004), ideally this critical density should be reached without artificial heating (uneasy, resolution of a few pc is needed)



$$\frac{dp}{d\ln x} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(\ln x - \overline{\ln x})^2}{2\sigma^2}\right]$$

$$\sigma^2 \approx \ln\left(1 + 3\mathcal{M}^2/4\right)$$

=> Naturally gives a power law for Σ_{SFR} - Σ_{gas}

Elmegreen 2002, Krumholz 2005, Renaud et al. 2012, Kraljic et al. 2014

Sub-grid star formation models

- In most simulations the sub-grid model is based on density and local free-fall time

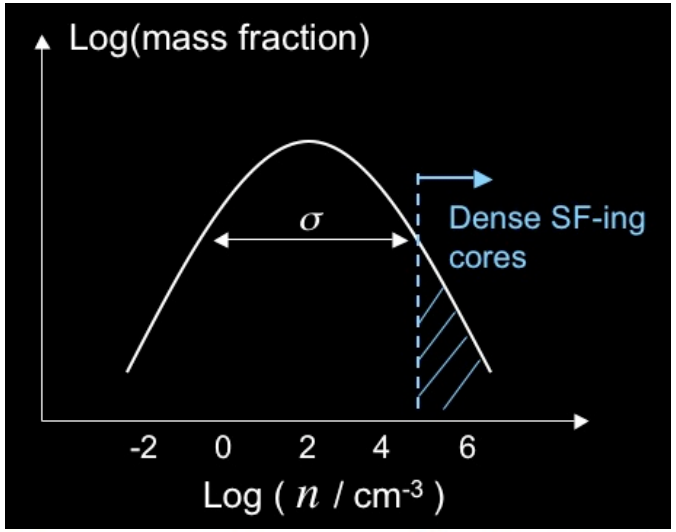
In each resolution element, 1-5% of gas is converted into stars per free-fall time

Efficiency 1-5% typically observed in CO clouds and HCN cores (Mc Kee 2007)

$$t_{ff} \sim 1/(G\rho)^{1/2}$$

- Other models probably more justified at resolutions lower than ~5-10pc : local virial criteria

(Hopkins et al. 2013, Perret Devriendt & Teyssier 2016)



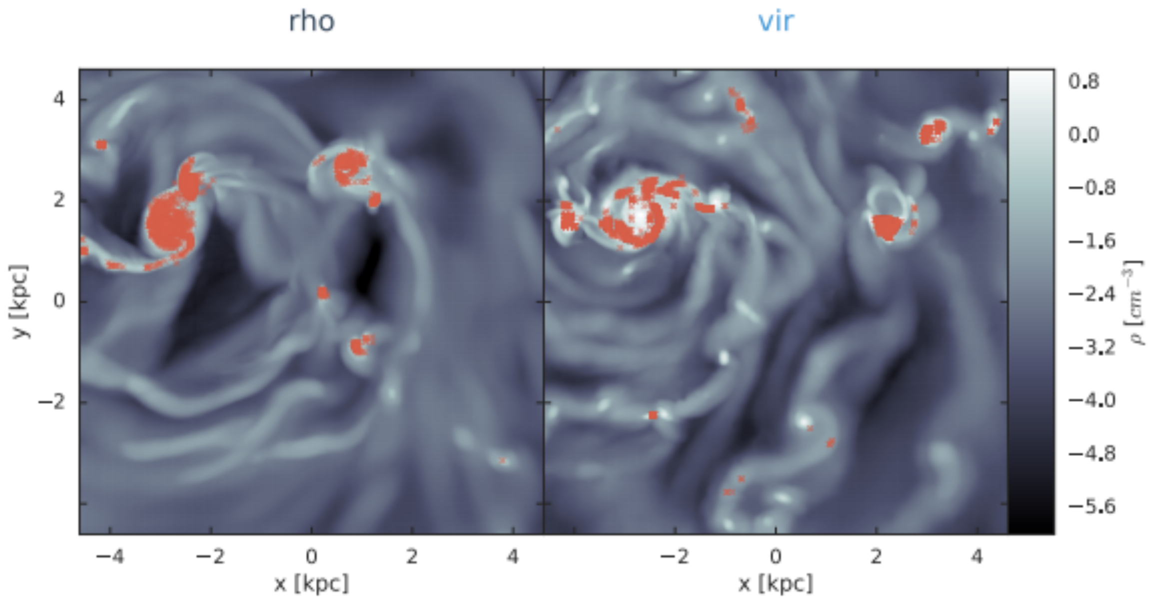
$$\frac{dp}{d\ln x} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(\ln x - \overline{\ln x})^2}{2\sigma^2}\right]$$

$$\sigma^2 \approx \ln\left(1 + 3\mathcal{M}^2/4\right)$$

=> Naturally gives a power law for Σ_{SFR} - Σ_{gas}

Elmegreen 2002, Krumholz 2005, Renaud et al. 2012, Kraljic et al. 2014

Sub-grid star formation models



$$\sigma_{eff}^2 + c_s^2 < \beta GM$$

$$\alpha = (\sigma_{eff}^2 + c_s^2)\delta r / \beta GM(< \delta r)$$

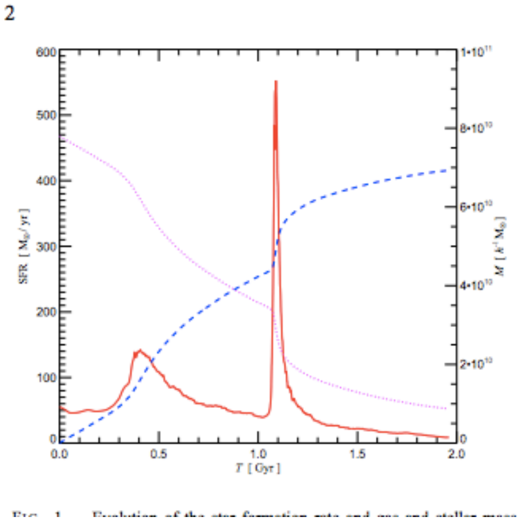
$$\alpha = \beta' \frac{|\nabla . v|^2 + |\Delta \times v|^2 + c_s^2}{G\rho} < 1$$

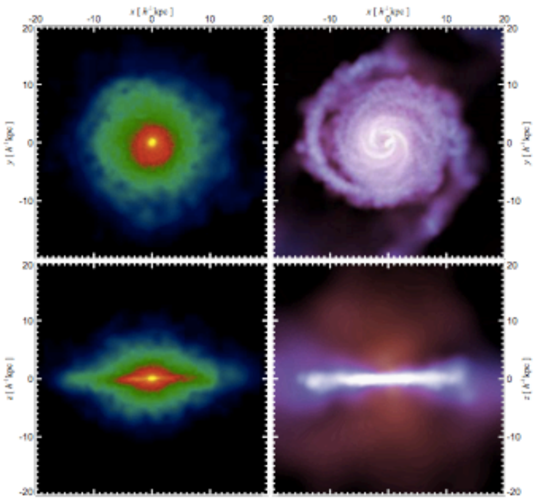
$$\beta' = 1/2 \qquad \epsilon = 1 \qquad \dot{\rho}_\star = \frac{\epsilon \rho}{t_{ff}}$$

- Other models probably more justified at resolutions lower than ~5-10pc : local virial criteria

(Hopkins et al. 2013, Perret Devriendt & Teyssier 2016)

Coupled SF+Feedback sub-grid models

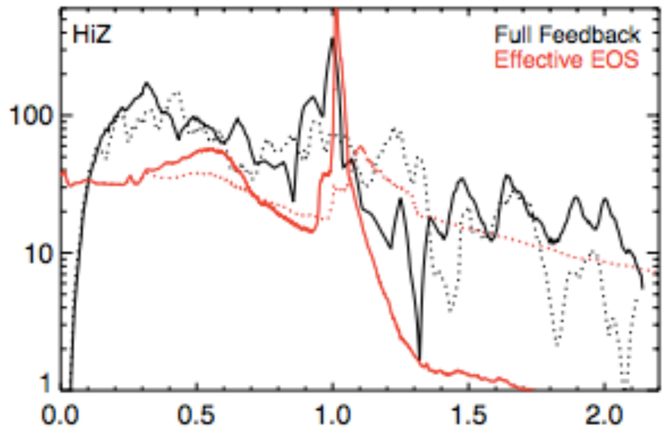




- Denser gas forms stars and is heated by young stars

- « effective EoS », typically polytropic raise of T for densities above ~$1 cm^{-3}$

For instance Springel & Hernquist 2005



Hopkins et al. 2013

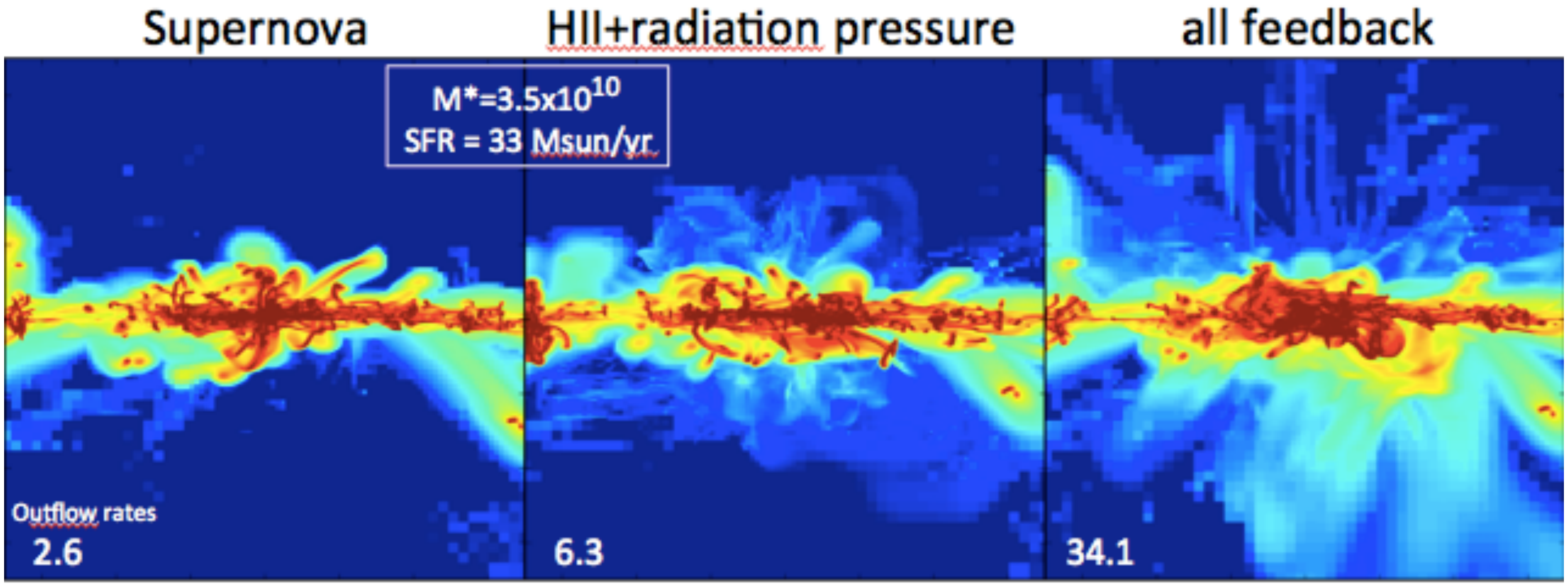
- Turbulent ISM with Mach>1 phases
- Star formation in cold, dense clouds
- Feedback from clustered star formation in cloudy gas

=> Very different star formation histories

(See also Jeremy Fensch's poster)

Explicit stellar feedback on galactic scales

Three runs with different feedback processes, all evolved for 80Myr



Outflow rate rapidly reaches 20-40 Msun/yr = SFR

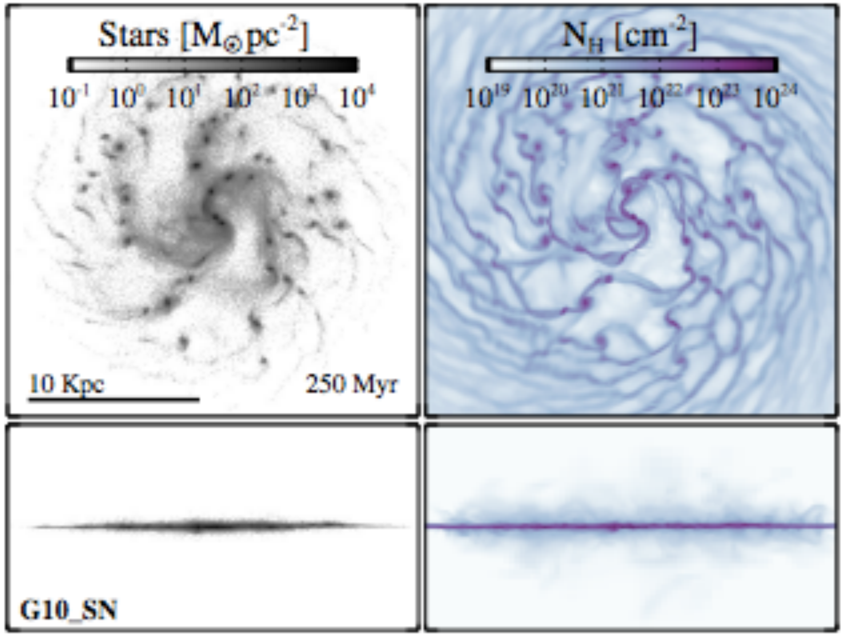
Supernovae alone don't do much

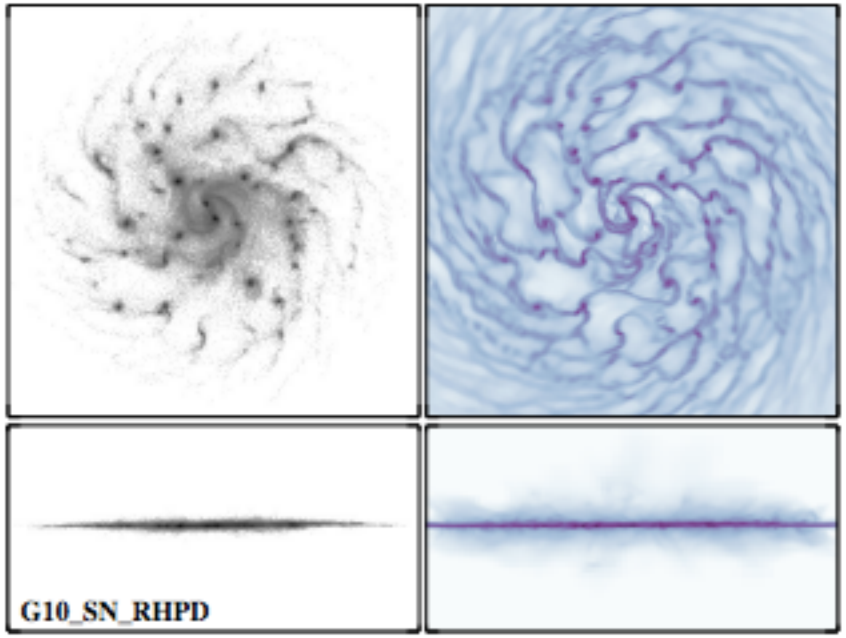
In general "feedback" remains largely sub-grid (arbitrary parameters often remain)

Bournaud 2015

Explicit stellar feedback on galactic scales

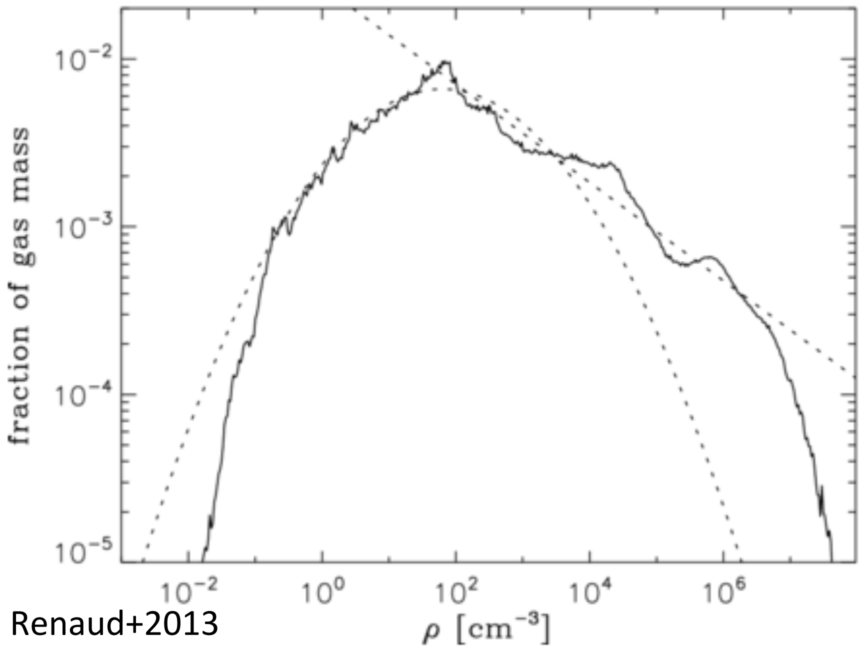
First simulations with explicit RT from young stars



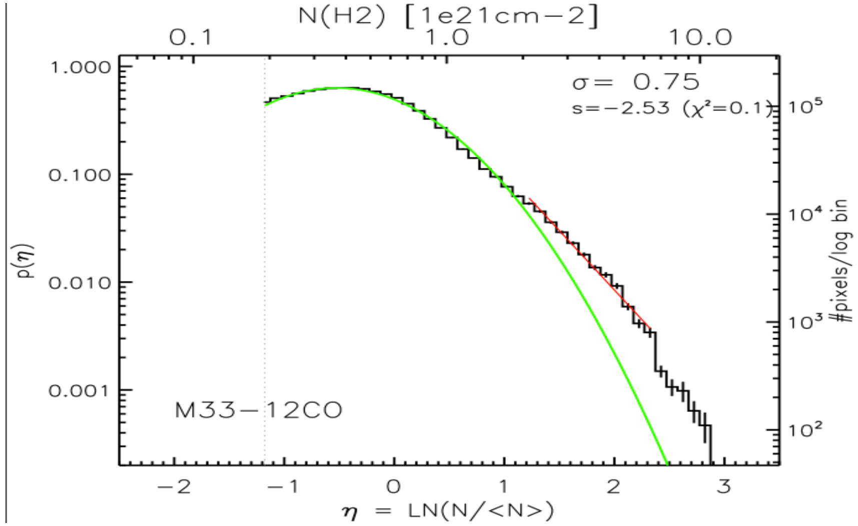


Rosdahl et al. 2015

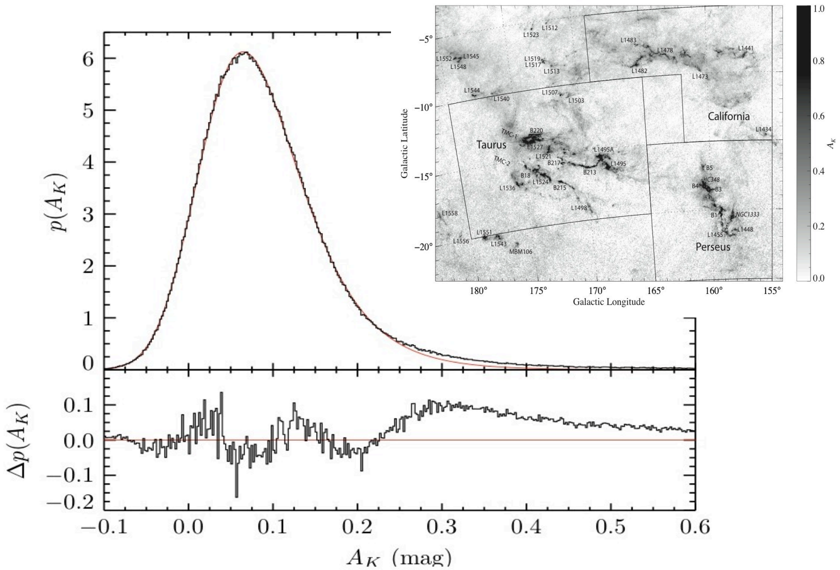
Structure of the turbulent ISM



Renaud+2013



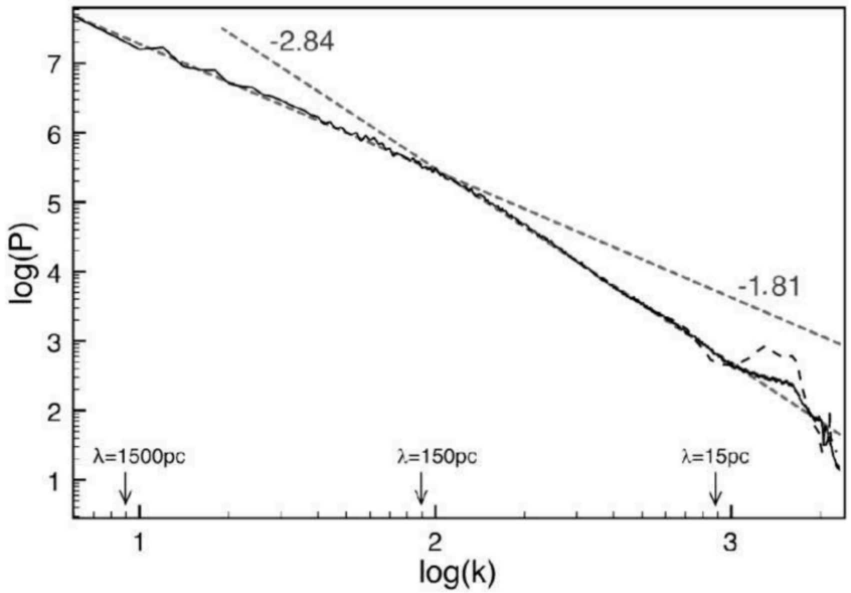
M33 - Druart+2015



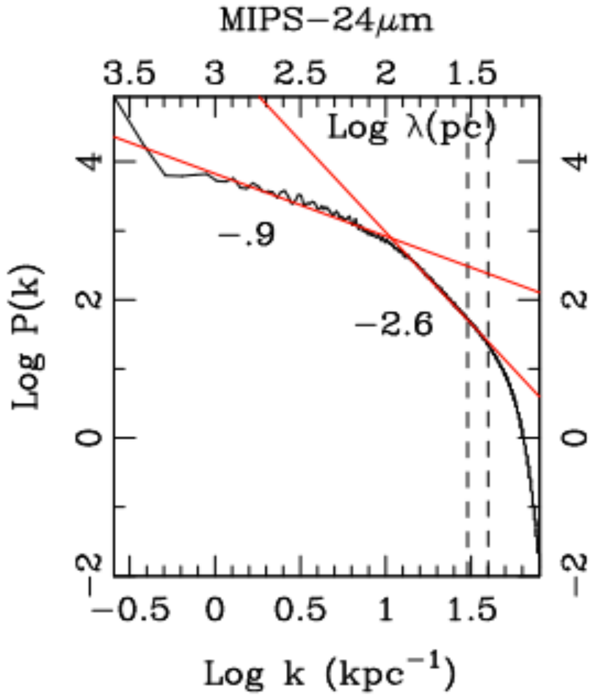
Taurus – Lombardi, Alves & Lada 2014

- Log-normal density PDF :
  expected for compressible, isothermal, supersonic turbulence
- Power-law tail at $>10^4$ cm^{-3} :
  expected for self-gravitating gas
- Tail contains ~2% of the ISM mass…
  sets the dense gas formation rate and SFR.

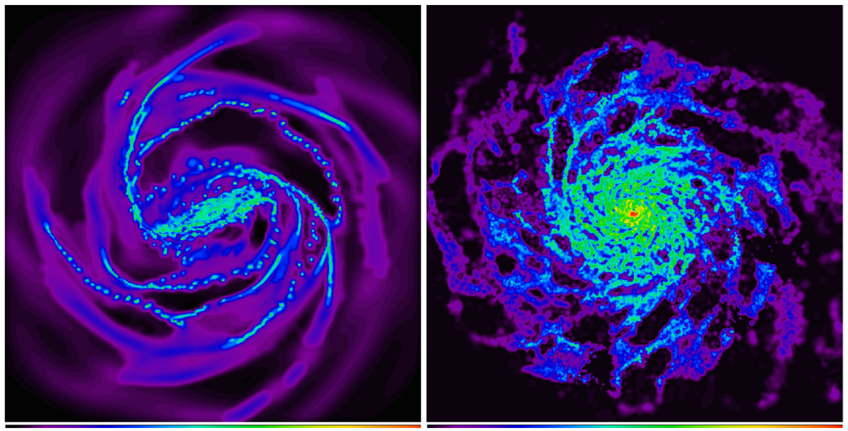
Structure of the turbulent ISM



Bournaud+10 simulation of M33-mass disk



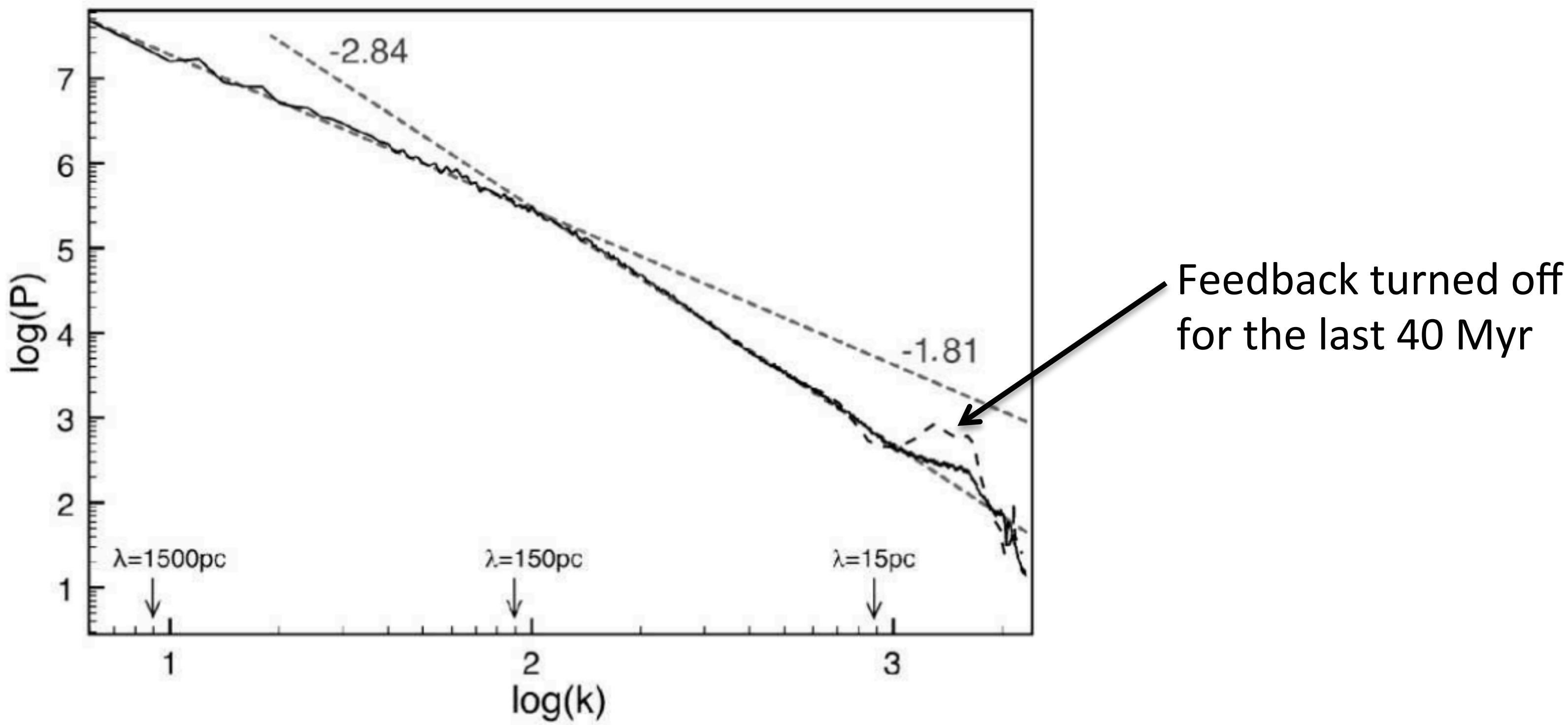
Combes et al. 2013 real M33 data



- Kolmogorov-like power spectrum below the mean Jeans length

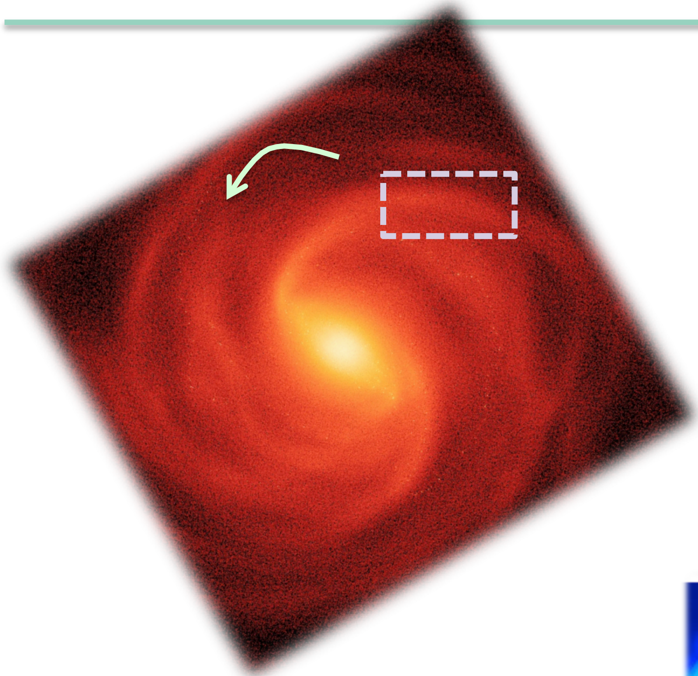
=> Consistent with all ISM data.
=> Injection-scale associated to gravity ?

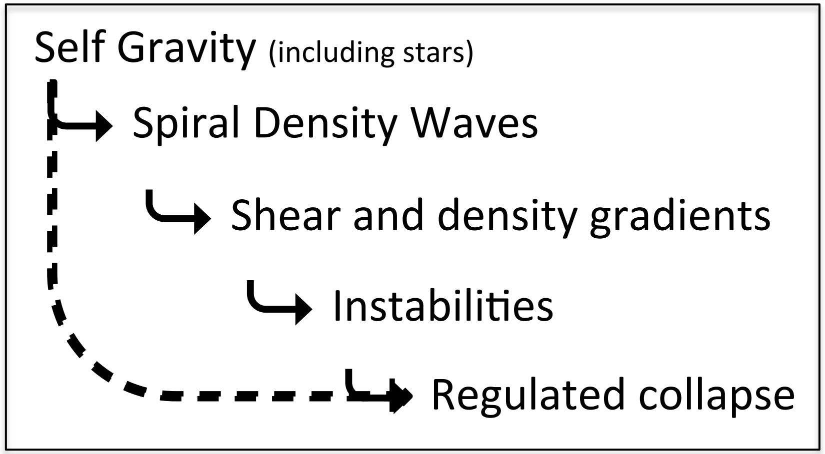
Did you say « feedback » ?



- The injection scale matches the most gravitationally-unstable scale
- The power spectrum is identical without feedback
- Feedback maintains the system in a steady state, by returning material from the « pseudo-dissipation scale » (in fact, the resolution limit) preventing gas from pilling-up in tiny bullets.

Gravitationnal and hydro instabilities and the turbulent cascade

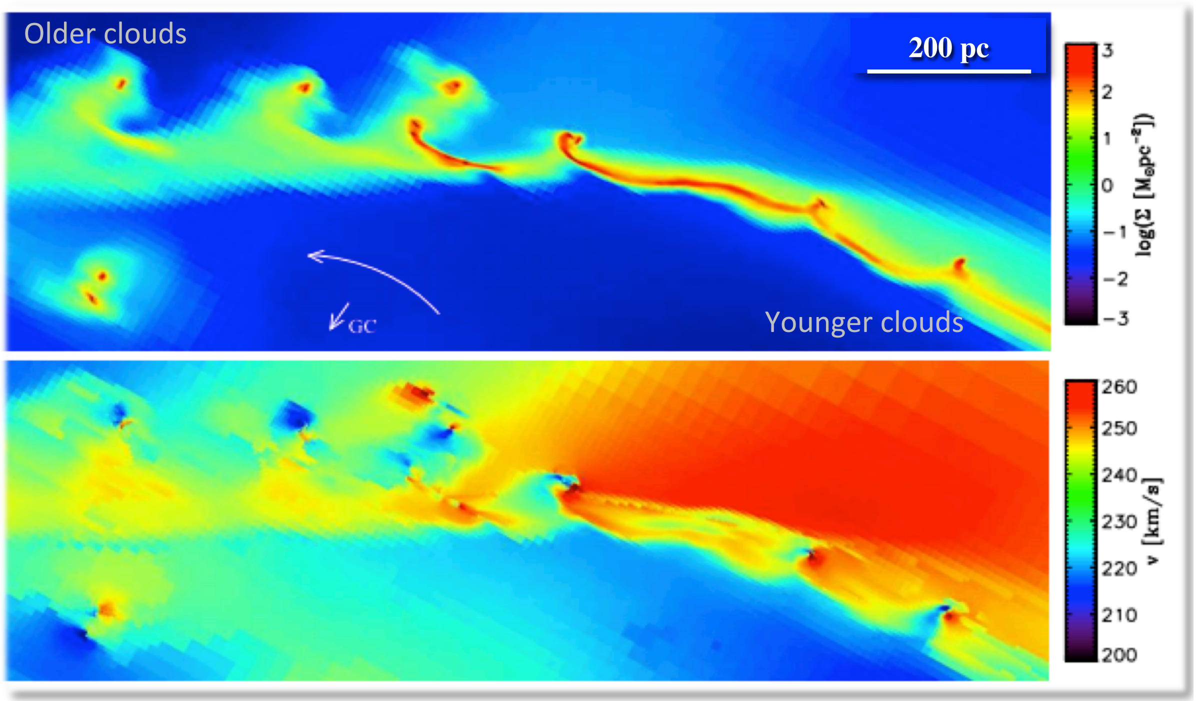




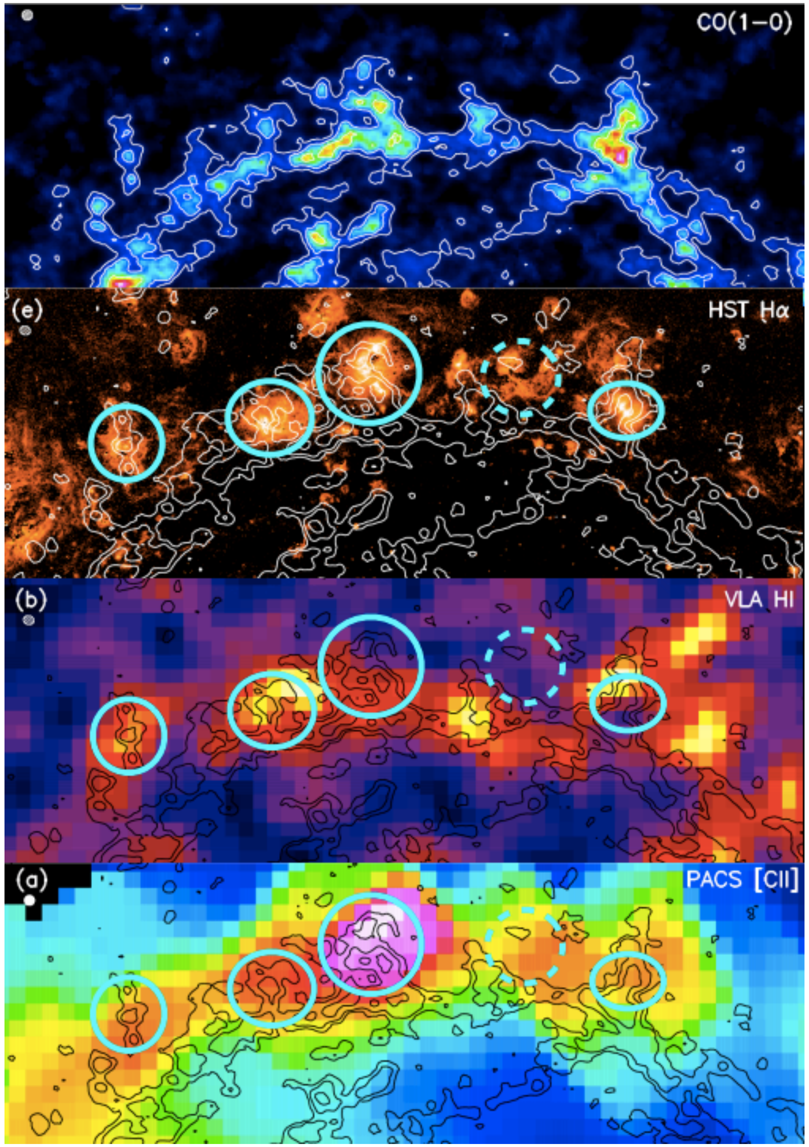
- A specific illustration of the gravito-hydro cascade and SFR regulation

- Note resemblence with « beads-on a string » clouds and « spurs » emanating from spiral arms

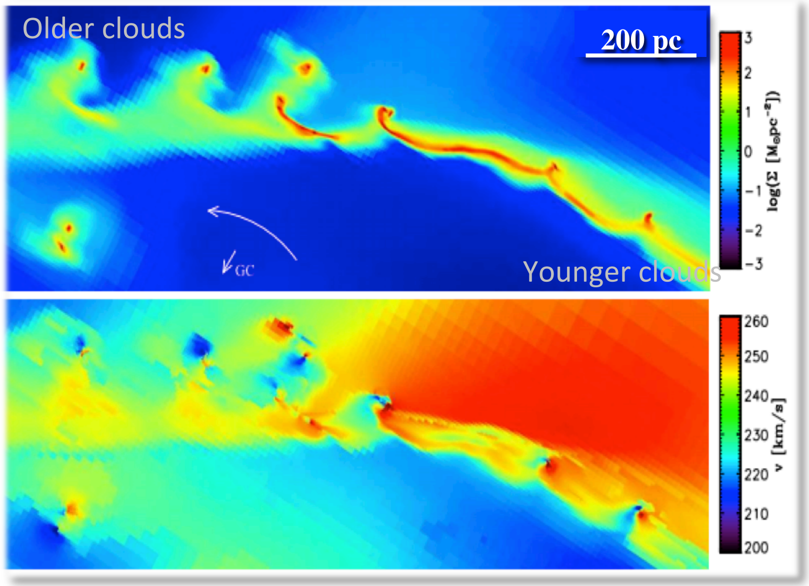
Renaud, Bournaud, Emsellem et al. 2013



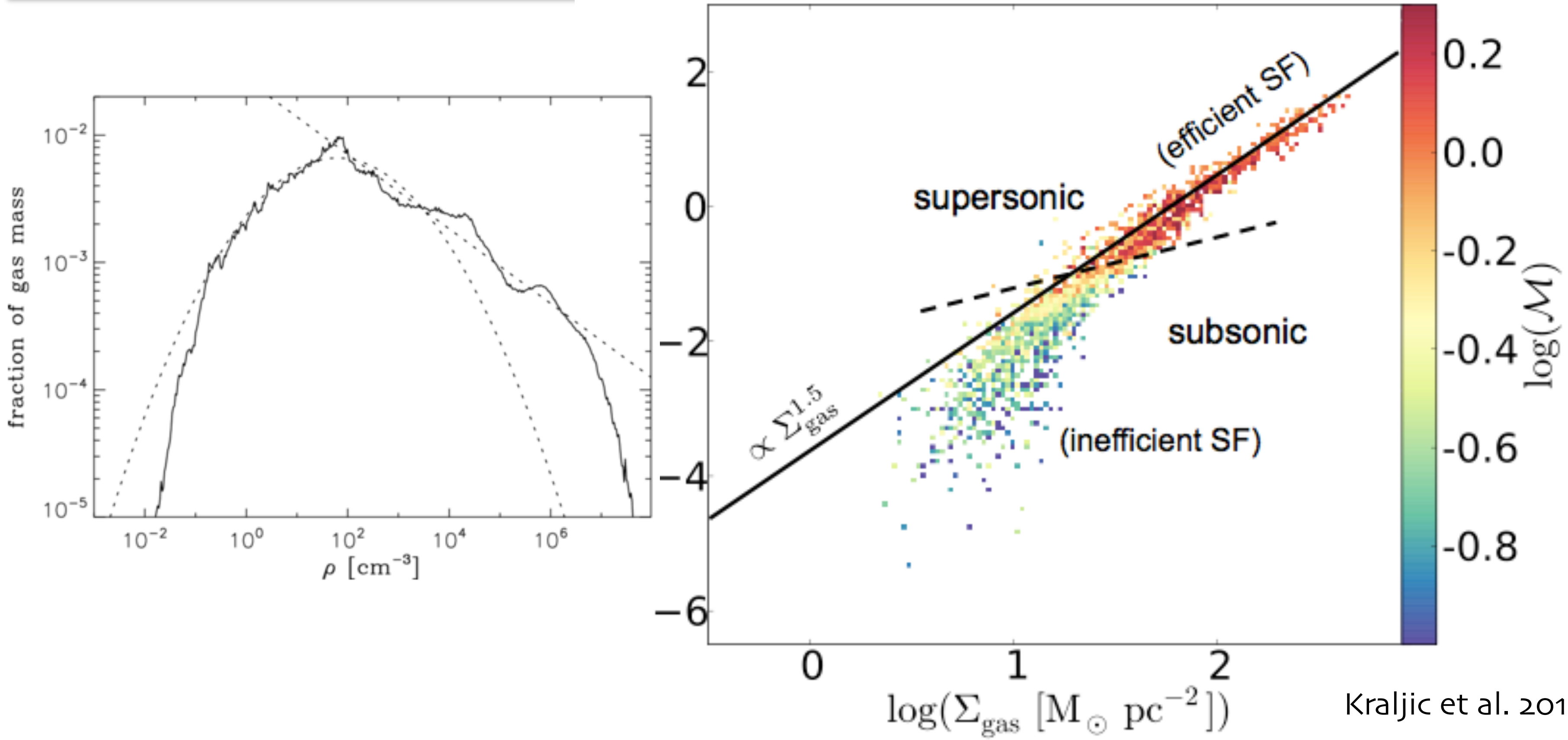
Gravitationnal and hydro instabilities and the turbulent cascade



- Spiral arms and spurs in M51
  (Schinnerer et al., IRAM/PAWS)
- Direct signature of SFR regulation by gravito-hydro turbulence
- Feedback-drven regulation would keep more molecular sites in spiral arms
  (Dobbs et al 2014)



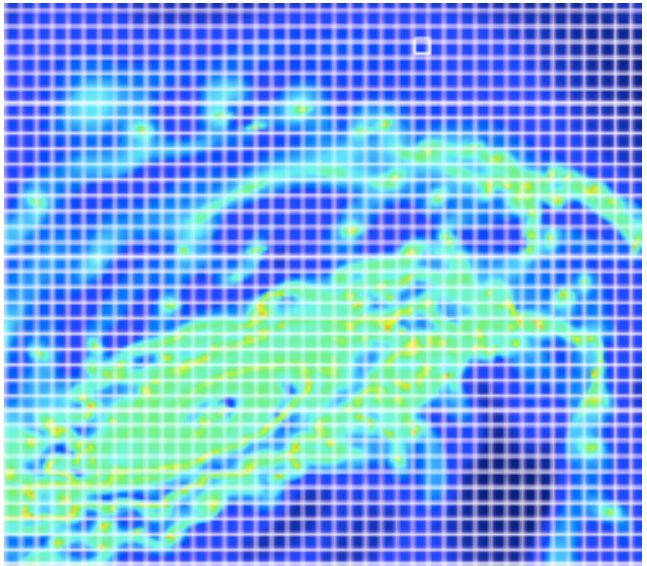
ISM turbulence and the « Schmidt-Kennicut law »

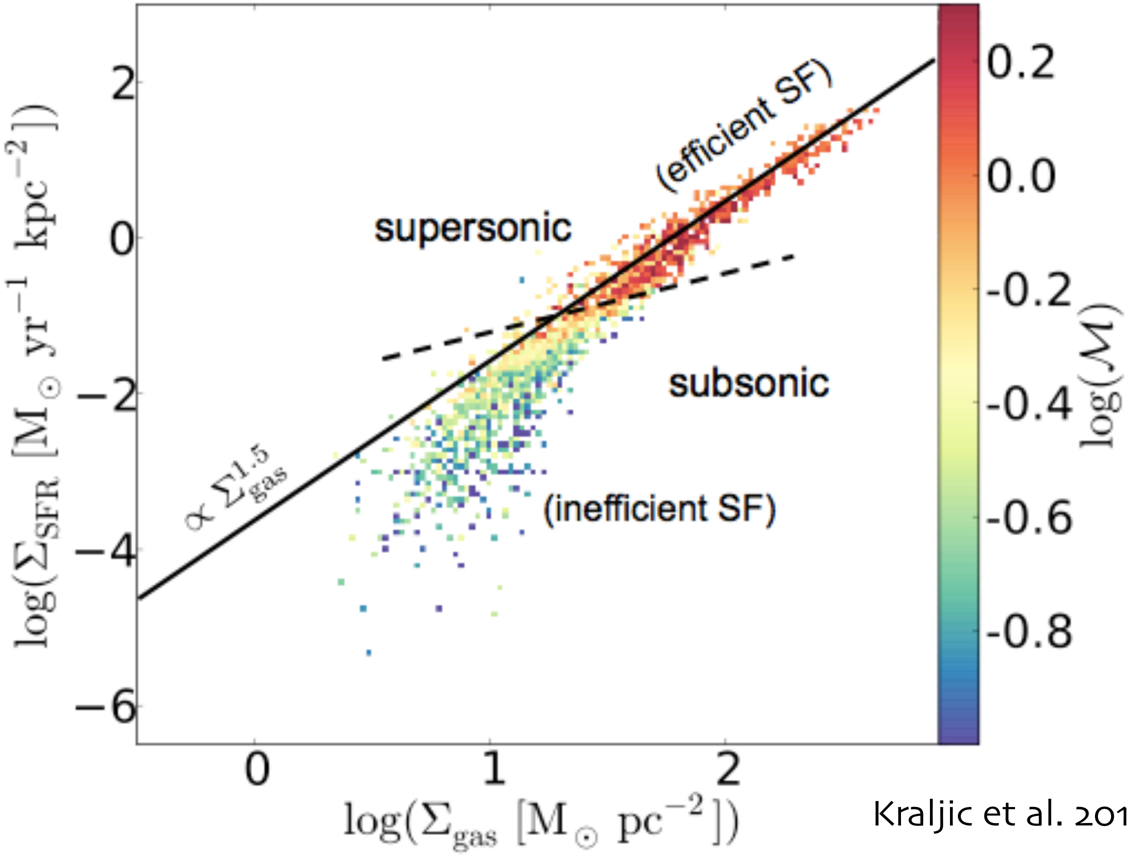


Kraljic et al. 2014

- This ISM turbulence naturally regulates SF (low amount of gas in the high-density structures)
- Kennicutt diagram directly matched :
  - low normalization of 'normal' star forming regions
  - innefficient regime at low densities : lack of cooling, sub-sonic turbulence transition.
- *Feedback is not the regulation source, is not the driver of turbulence, it keeps a steady state*

ISM turbulence and the « Schmidt-Kennicut law »

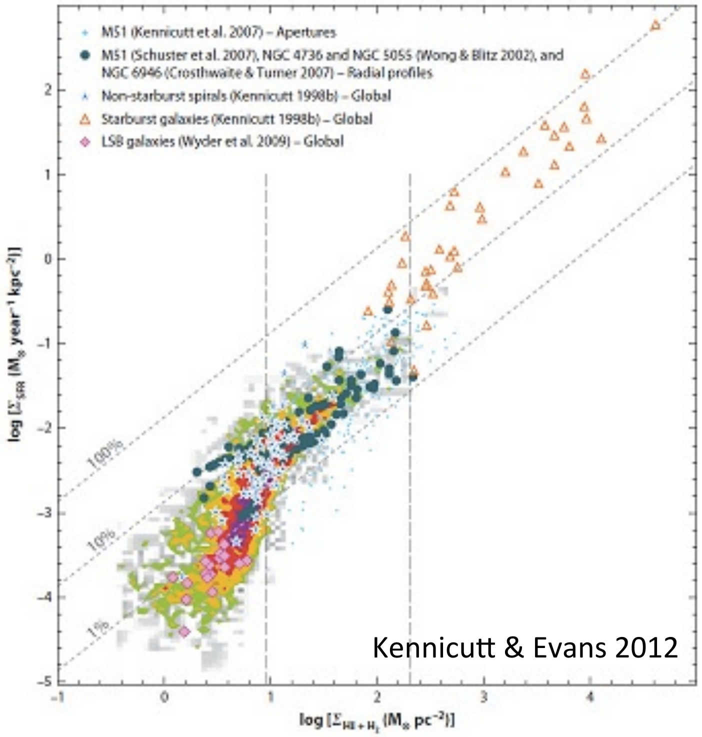


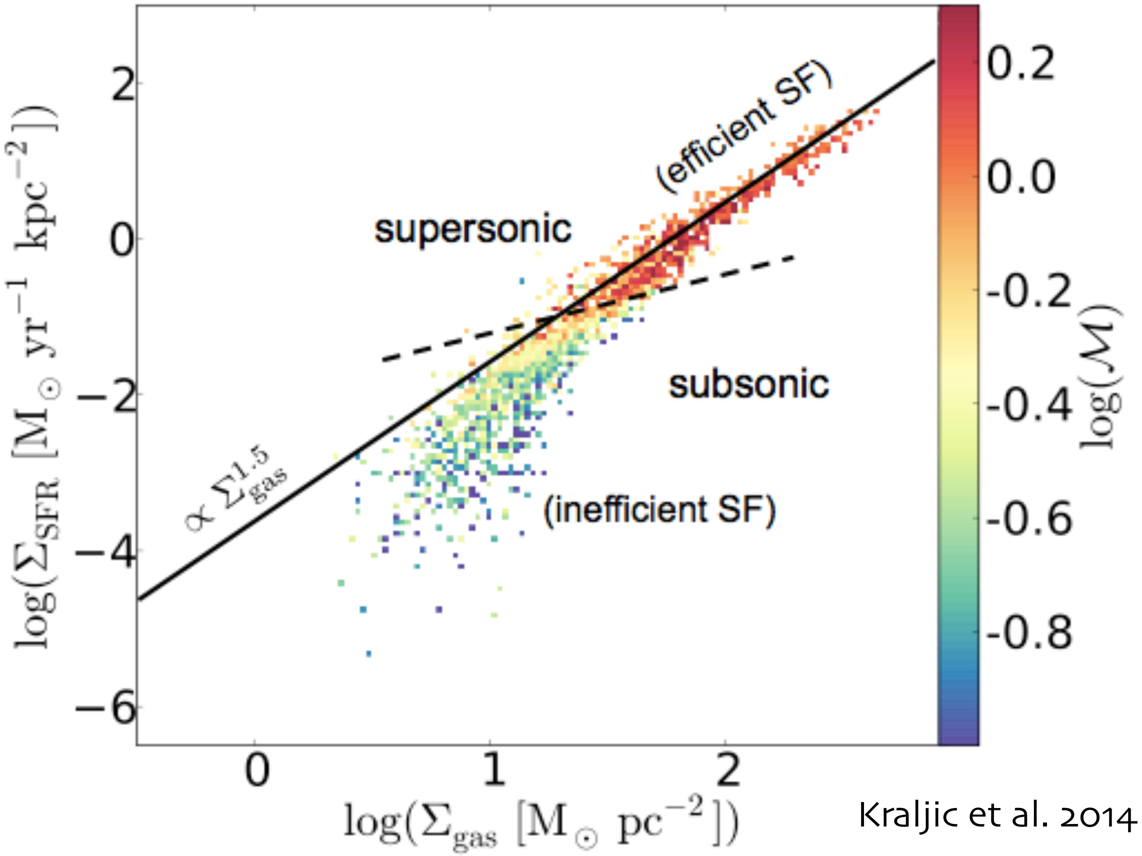


Kraljic et al. 2014

- This ISM turbulence naturally regulates SF (low amount of gas in the high-density structures)
- Kennicutt diagram directly matched :
  - low normalization of 'normal' star forming regions
  - innefficient regime at low densities : lack of cooling, sub-sonic turbulence transition.
- *Feedback is not the regulation source, is not the driver of turbulence, it keeps a steady state*

ISM turbulence and the « Schmidt-Kennicut law »

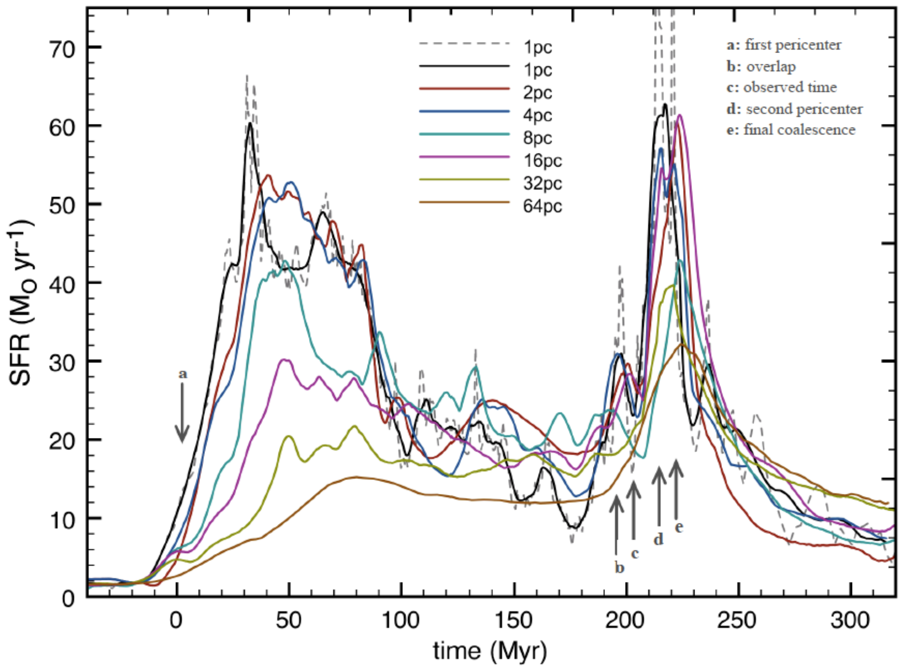


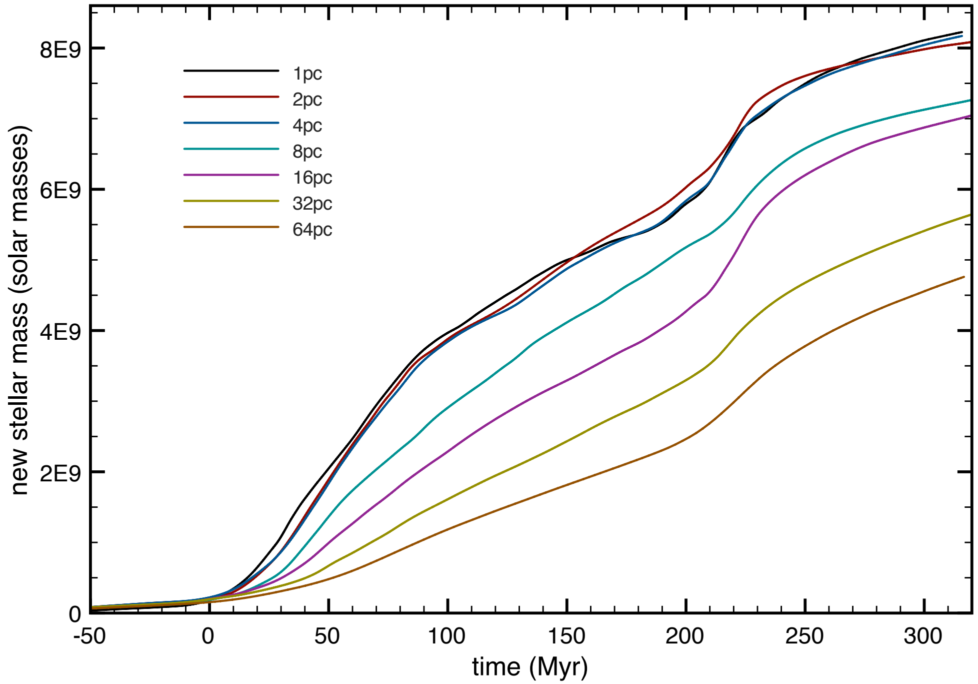


Kraljic et al. 2014

- This ISM turbulence naturally regulates SF (low amount of gas in high-density structures)
- Kennicutt diagram directly matched :
  - low normalization of ‘normal’ star forming regions
  - innefficient regime at low densities : lack of cooling, sub-sonic turbulence transition.
- *Feedback is not the regulation source, is not the driver of turbulence, it keeps a steady state*

Do simulations really resolve the critical scales ?



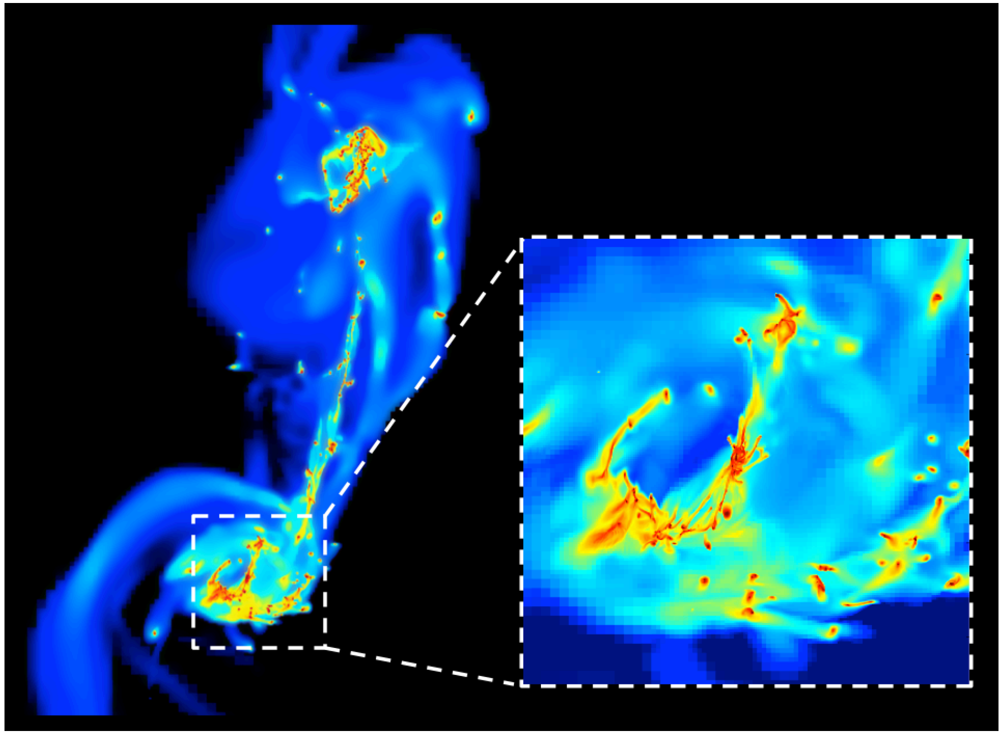


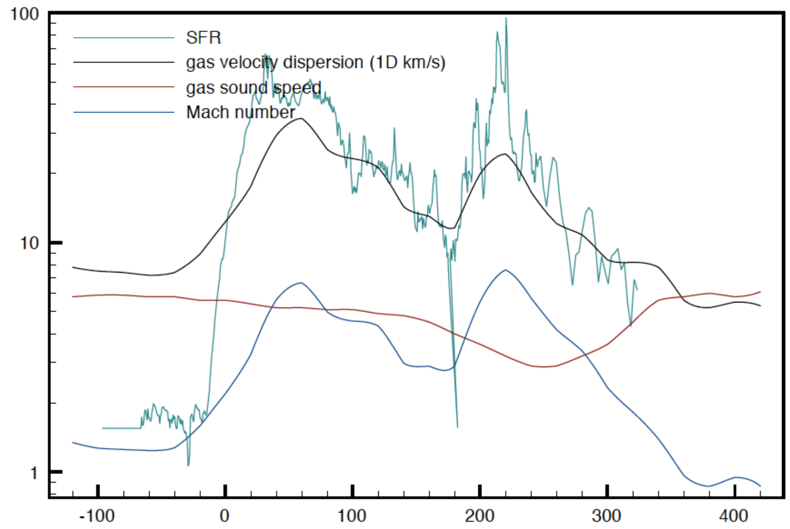
- Resolution effects are strong, then become almost negligible at ~5pc resolution, leaving only fluctuations.

- Gas becomes self-gravitating at ~10^4cm^{-3}, main structure size ~20-40pc

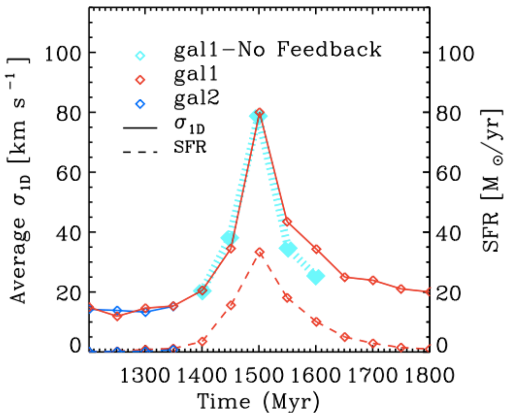
=> 4-6 resolution elements per cloud size => resolve the turbulent flow compressivity, at the scale where self-gravity takes over turbulent pressure toward star formation.

Starbursts in galaxy collisions



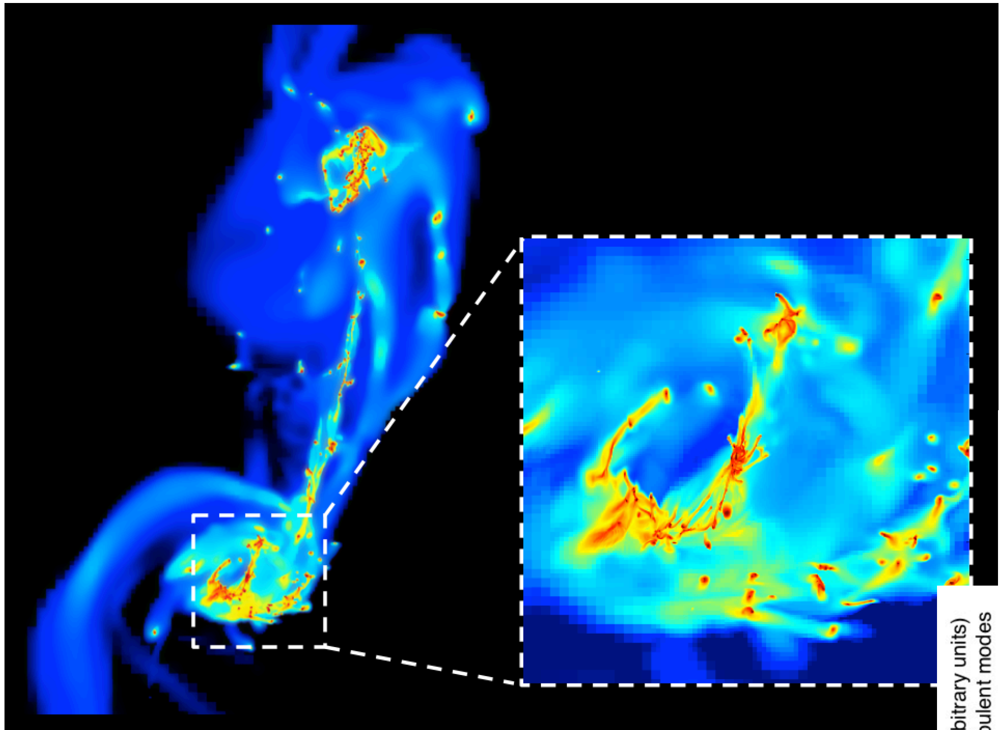


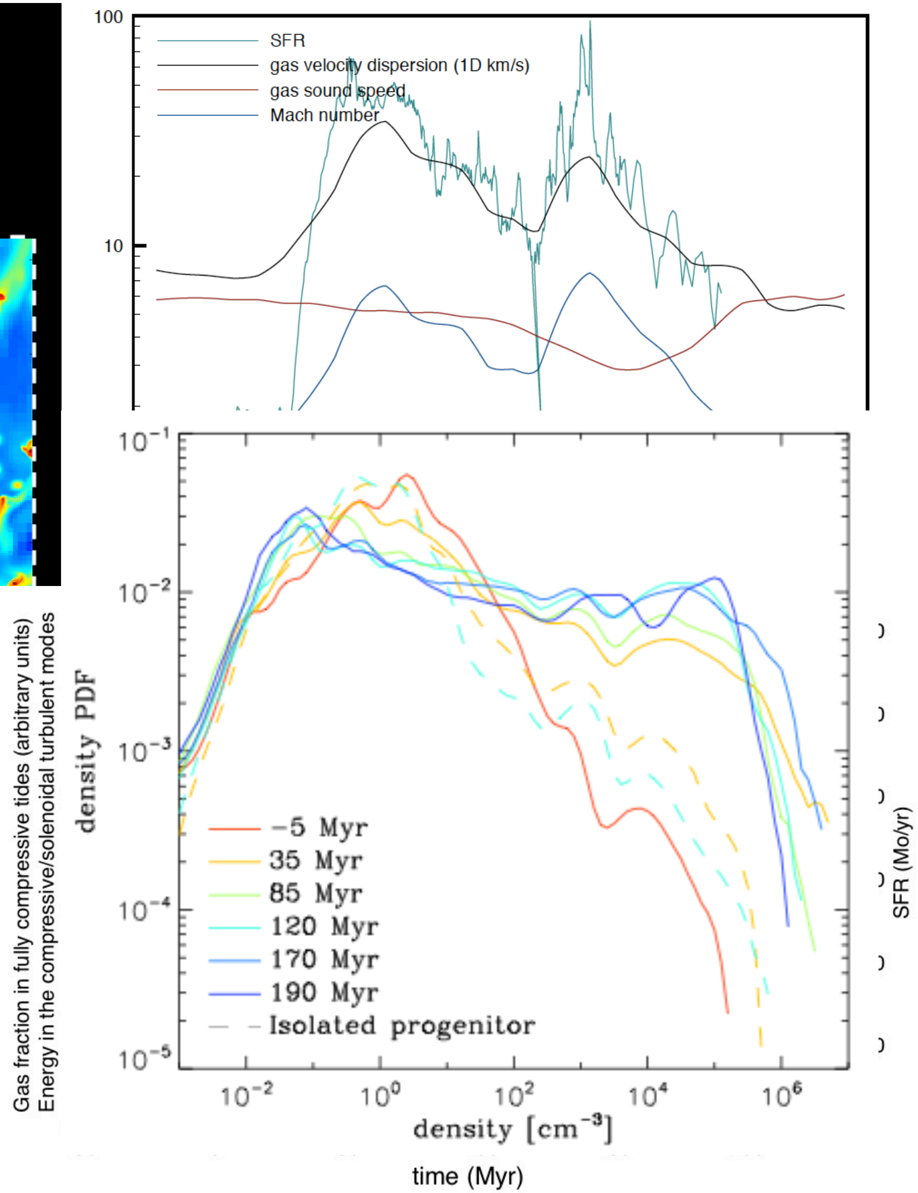
1-parsec resolution AMR simulation
of a galaxy collision similar to the « Antennae »
(Renaud, Bournaud et al. 2014,15)



Physical link between increased ISM turbulence
and starburst activity (not from feedback)

Starbursts in galaxy collisions





$$v_{turb} = v_{sol} + v_{comp}$$

$$\nabla . v_{sol} = 0 \qquad \nabla\times v_{sol} = \nabla\times v_{turb}$$
$$\nabla . v_{comp} = \nabla . v_{turb} \qquad \nabla\times v_{comp} = 0$$

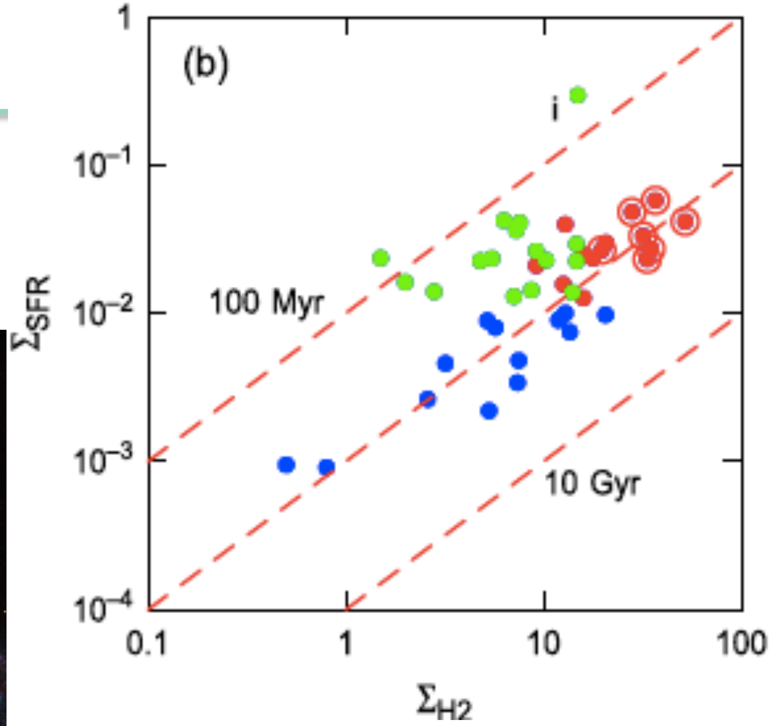
Helmholtz decomposition of the turbulent field shows a process like :

compressive gravity (tides) on kpc scale

=> compressive turbulence on pc scale

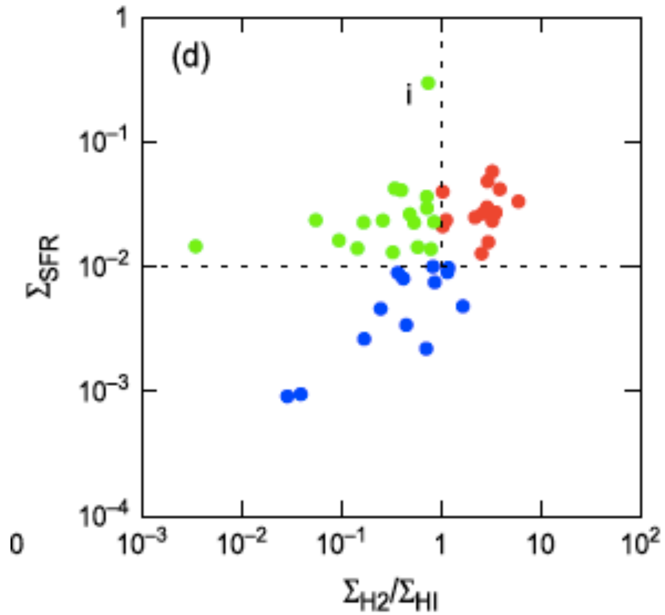
=> starburst

Starbursts in galaxy collisions

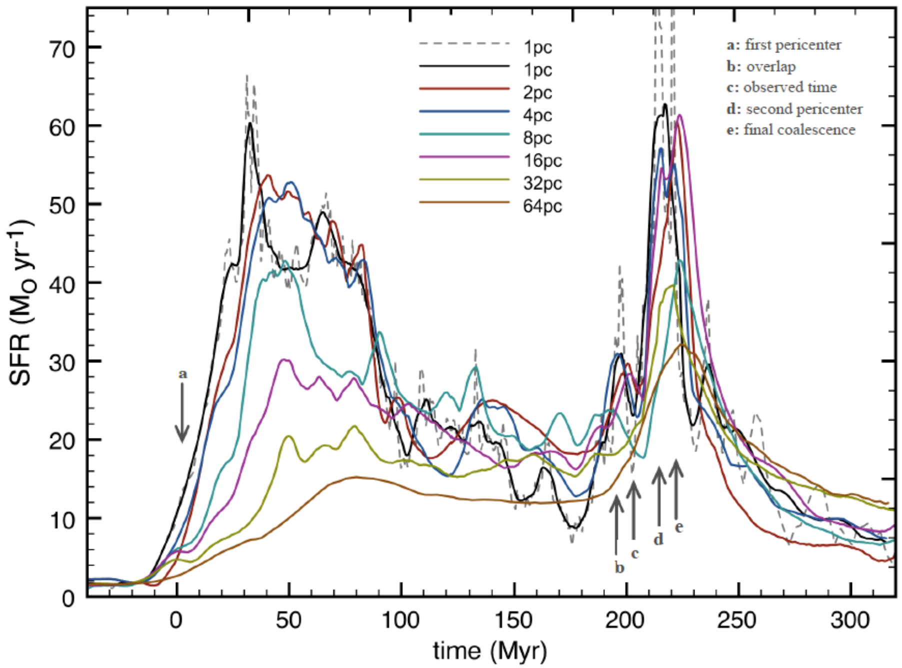


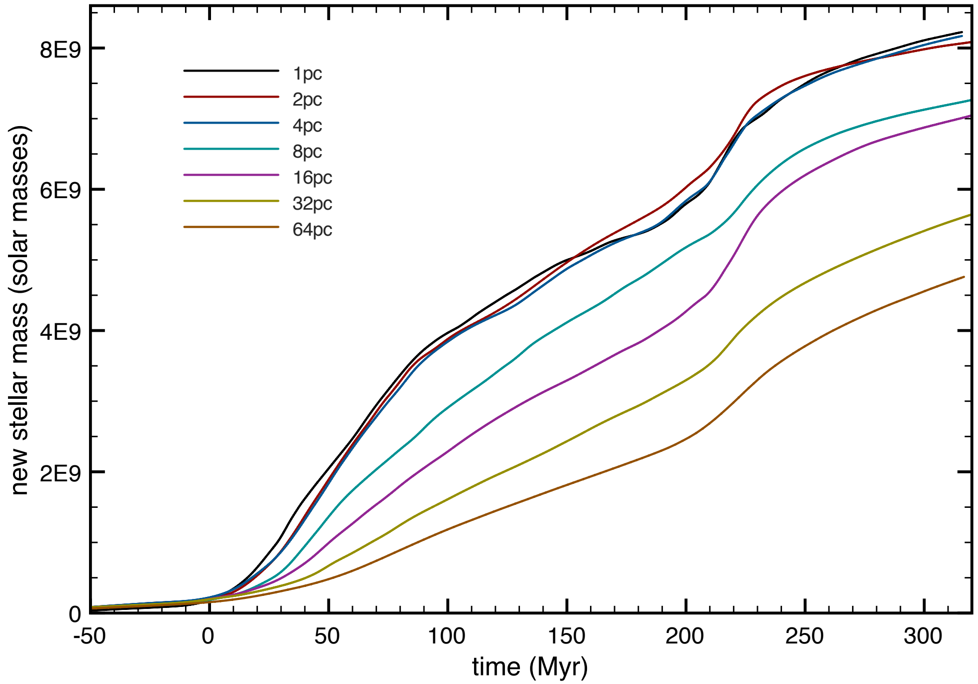
New ALMA data on NGC2207 (Elmegreen et al. 2016)
Many regions with low H2/HI but high SFR, generally high σ too

Consistent with non-log normal density distributions



Do simulations really resolve the critical scales for star formation ?





- Resolution effects are strong, then become almost negligible at ~5pc resolution, leaving only fluctuations.

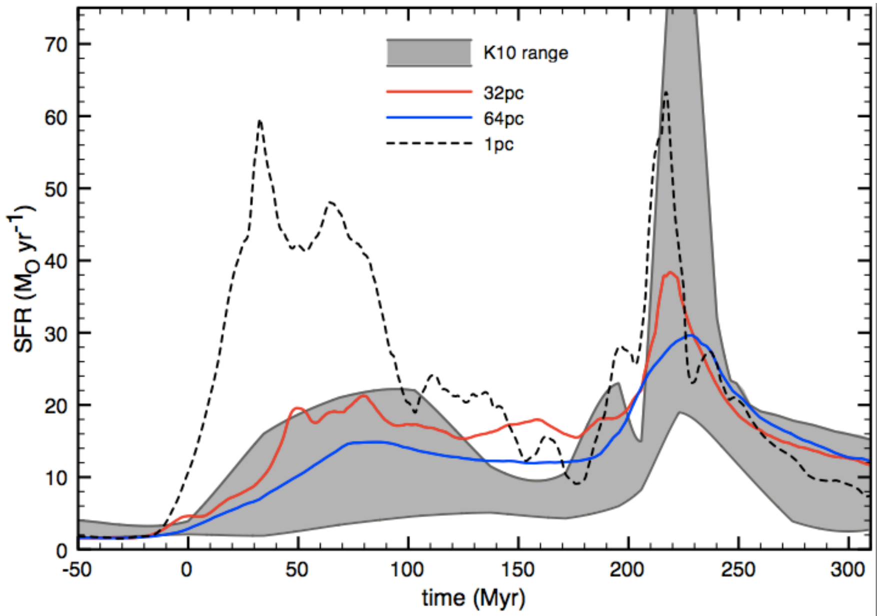
- Gas becomes self-gravitating at ~10^4cm^{-3}, main structure size ~20-40pc

=> 4-6 resolution elements per cloud size => resolve the turbulent flow compressivity, at the scale where self-gravity takes over turbulent pressure toward star formation.

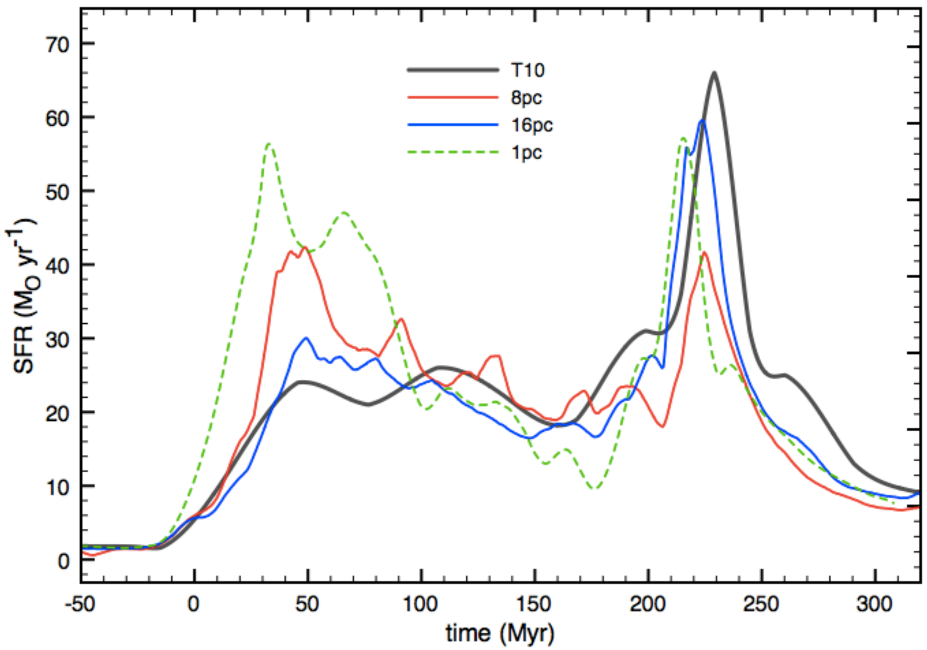
Do simulations really resolve the critical scales for star formation ?

Various Antennae-like mergers :

- At a given resolution, great agreement regardless of details of code, feedback, SF recipes, interaction orbit and galaxy parameters (disk/bulge masses and sizes)
- In any case, resolution effects much more important until ~5pc is reached



Comparing to Karl, Naab et al. 2010 (K10)
SPH simulations with ~50pc resolution
and various sub-grid parameters



Comparing to Teyssier et al. 2010 (T10)
Same code but EoS (fake) cooling and no feedback

Resolution is most important and sub-grid physics never recovers unresolved turbulence!