Michela Mapelli

N-body techniques for astrophysics: Lecture 6 – GAS

PhD School in Astrophysics,
University of Padova
November 19-30, 2018

OUTLINE of this lecture:

1 – Equations of gas

2 – Mesh codes/ Adaptive mesh refinement codes (AMR)

3 – Smoothed-particle hydrodynamics (SPH) codes

4 – Some good reasons for choosing SPH versus some good reasons for choosing AMR

5 – Example codes: gasoline with gas, RAMSES

1. EQUATIONS of GAS

Not only GRAVITY (Newton equation) but also
EULER EQUATIONS IN CONSERVATIVE FORM (except for gravity):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0$$ **CONTINUITY**

$$\frac{\partial \rho u}{\partial t} + \nabla \cdot (\rho u \otimes u) + \nabla p = -\rho \nabla \phi$$ **MOMENTUM**

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot [\rho u (e + p/\rho)] = -\rho u \cdot \nabla \phi$$ **ENERGY**

mass density — ρ
specific energy — e
fluid velocity — u
thermal pressure — p
acceleration — ϕ

$$p = (\gamma - 1)\,\rho \left(e - \frac{1}{2} u^2 \right)$$

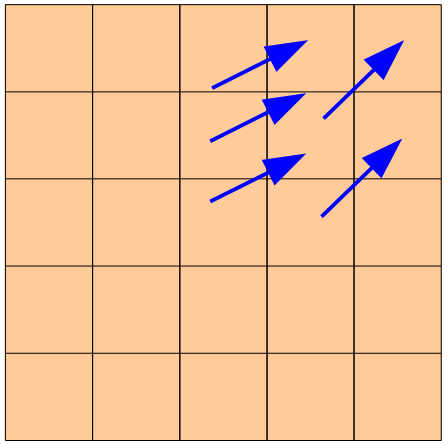
Much more complex than Newton alone..

1. EQUATIONS of GAS

Different methods to solve equations of gas numerically

EULERIAN:

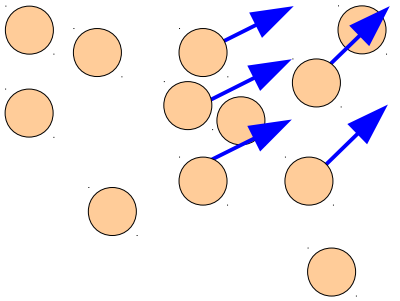
Mesh



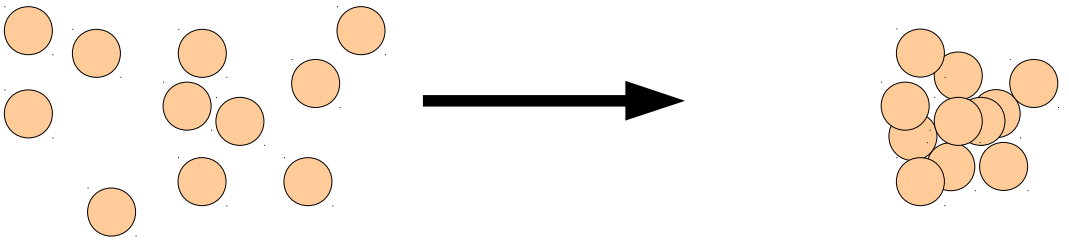
Advantages:
High accuracy
Low numerical viscosity
Shock capturing

LAGRANGIAN:

Particles



Advantages:
Resolution adjusts automatically (density increases where needed- eg collapse)



2. MESH CODES: GODUNOV SCHEME

Simplify: 1-dimensional case

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} = 0$$

$$\frac{\partial (\rho u)}{\partial t} + \frac{\partial (\rho u^2 + p)}{\partial x} = -\rho \frac{\partial \phi}{\partial x}$$

$$\frac{\partial \rho e}{\partial t} + \frac{\partial [\rho u \, (e + p/\rho)]}{\partial x} = -\rho \frac{\partial \phi}{\partial x}$$

0

0

No self-gravity

Let us define:

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho e \end{bmatrix}$$

physical quantity

$$F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u \, (e + p/\rho) \end{bmatrix}$$

its flux-momentum

2. GODUNOV SCHEME

We discretize space in *N* cells of size $\Delta x = x_{i+1/2} - x_{i-1/2}$

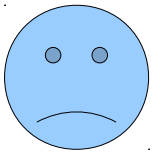
Integral form in volume of Euler equations is:

$$\frac{\partial}{\partial t}\int_{x_{i-1/2}}^{x_{i+1/2}} U(x,t)\,dx = -F(x_{i+1/2},t) + F(x_{i-1/2},t)$$

Integral form in time (between t_n and t_{n+1} with $t_{n+1}>t_n$)

$$\int_{x_{i-1/2}}^{x_{i+1/2}} U(x,t^{n+1})\,dx = \int_{x_{i-1/2}}^{x_{i+1/2}} U(x,t^{n})\,dx +$$

$$-\int_{t^n}^{t^{n+1}} F(x_{i+1/2},t)\,dt + \int_{t^n}^{t^{n+1}} F(x_{i-1/2},t)\,dt$$

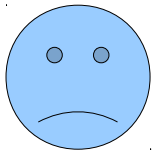


2. GODUNOV SCHEME

Now we define:

$$U_i^n = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} U(x, t^n)\, dx$$

$$F_{i\pm1/2} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} F(x_{i\pm1/2}, t)\, dt$$



$$U_i^{n+1} = U_i^n + \frac{\Delta t}{\Delta x}\left(F_{i-1/2} - F_{i+1/2}\right)$$

Looks like a numerical algorithm but is a correct analytic expression

BUT: in general, we do not know the exact form of analytic functions *U(x,t)* and *F(x,t)*

2. GODUNOV SCHEME

$$U_i^{n+1} = U_i^n + \frac{\Delta t}{\Delta x}\left(F_{i-1/2} - F_{i+1/2}\right)$$

Looks like a numerical scheme but is a correct analytic expression

BUT: in general, we do not know the exact form of analytic functions *U(x,t)* and *F(x,t)*

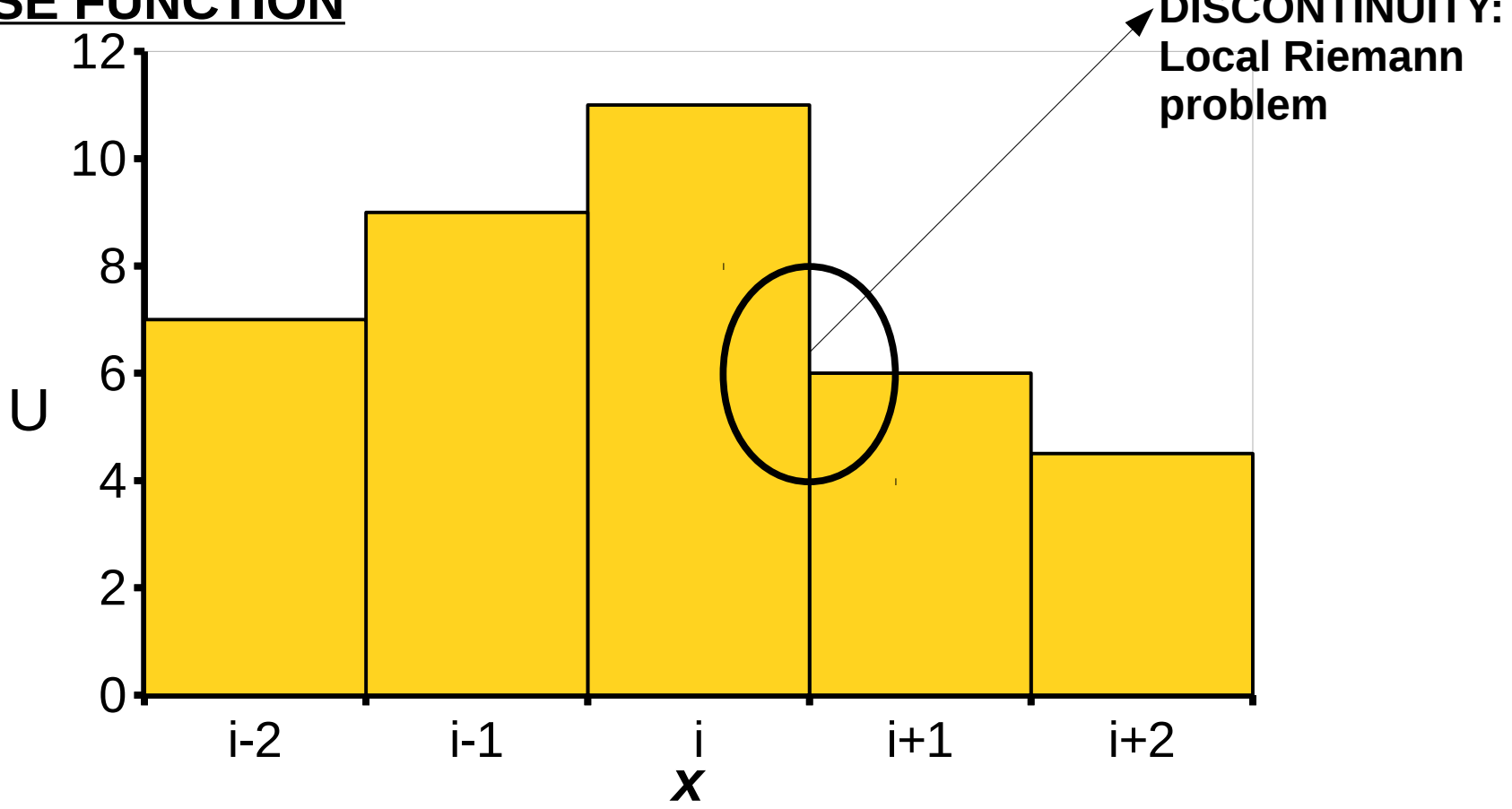
IF WE APPROXIMATE *U(x,t)* and *F(x,t)*, the equation becomes a numerical scheme: THE GODUNOV SCHEME for Euler equations

In 1st order Godunov method, approximation is a PIECE-WISE FUNCTION

2. GODUNOV SCHEME

In 1st order Godunov method, the approximation is a PIECE-WISE FUNCTION

e.g.



→ **Using piece-wise function generates DISCONTINUITY between cells := means solving a RIEMANN PROBLEM FOR EACH CELL (Riemann problem: *an initial value problem or Cauchy problem composed of a conservation equation together with piecewise constant data having a single discontinuity*)**

2. GODUNOV SCHEME

E.G. if flux $F = a\,U$ with a = constant

→ Solution of Riemann problem

$$F_{i+1/2} = a\,U_i^n \qquad if \quad a > 0$$

(flow from cell *i* to cell *i+1*)

$$F_{i+1/2} = a\,U_{i+1}^n \qquad if \quad a < 0$$

(flow from cell *i+1* to cell *i*)

The algorithm

$$U_i^{n+1} = U_i^n + \frac{\Delta t}{\Delta x}\left(F_{i-1/2} - F_{i+1/2}\right)$$

Becomes

$$U_i^{n+1} = U_i^n - c\,(U_i^n - U_{i-1}^n) \qquad if \quad a > 0$$

$$U_i^{n+1} = U_i^n - c\,(U_{i+1}^n - U_i^n) \qquad if \quad a < 0$$

Where

$$c = a\,\frac{\Delta t}{\Delta x}$$

(c = Courant factor, Courant et al. 1952)

2. GODUNOV SCHEME

GODUNOV 1ST ORDER UPWIND SCHEME (Courant et al. 1952)

$$U_i^{n+1} = U_i^n - c\left(U_i^n - U_{i-1}^n\right) \qquad if \quad a > 0$$

$$U_i^{n+1} = U_i^n - c\left(U_{i+1}^n - U_i^n\right) \qquad if \quad a < 0$$

where $c = a\dfrac{\Delta t}{\Delta x}$ **(c = Courant factor, Courant et al. 1952)**

- Time accuracy can be increased combining with Runge-Kutta

- Spatial accuracy can be increased with reconstruction procedure of cells

-Δt must satisfy $\Delta t \leq \dfrac{\Delta x}{|v_{\max}^n|}$ ← Maximum velocity in the entire computational domain at $t = t^n$

- STRENGTH: solves whichever system of non-linear equations with extremely simple algorithm

- WEAKNESS: solving a Cauchy problem at each cell interface is slow: approximate Riemann solvers were introduced to simplify

2. GODUNOV SCHEME

How do we include gravity in Euler equations?

For example (RAMSES, Teyssier 2002)

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} + \frac{F_{i+1/2} - F_{i-1/2}}{\Delta x} = S_i^{n+1/2}$$

$$S_i^{n+1/2} = \begin{cases} 0, & \leftarrow \text{continuity} \\ \dfrac{\rho_i^n \nabla\phi_i^n + \rho_i^{n+1} \nabla\phi_i^{n+1}}{2}, & \leftarrow \text{momentum} \\ \dfrac{(\rho u)_i^n \nabla\phi_i^n + (\rho u)_i^{n+1} \nabla\phi_i^{n+1}}{2} & \leftarrow \text{energy} \end{cases}$$

2. GODUNOV SCHEME

How is gravity solved?

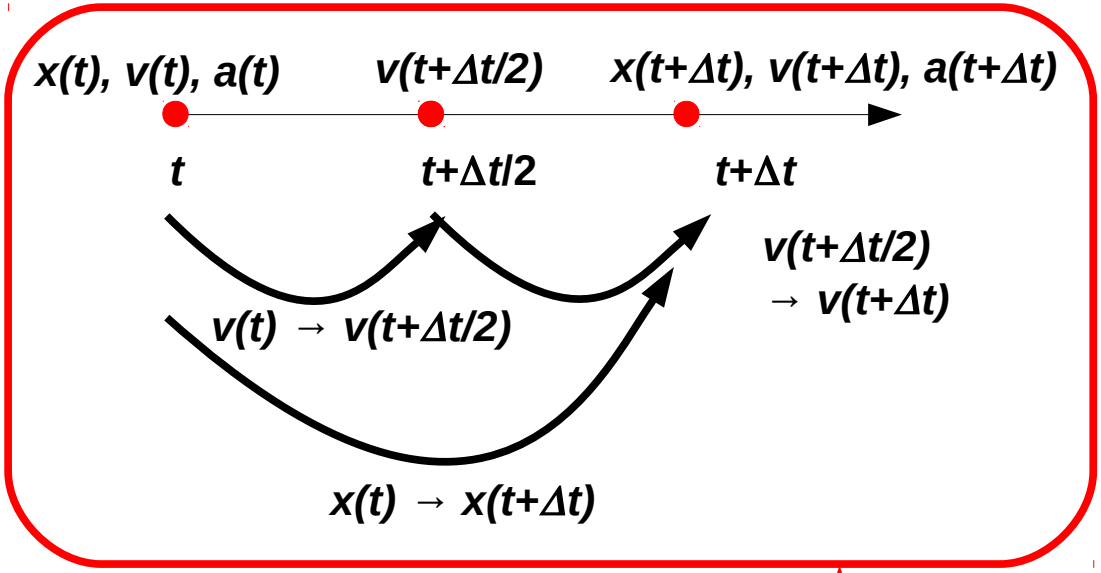
Godunov scheme for gas needs that TIMESTEP can change with time
→ not possible with Leapfrog

For example (RAMSES, Teyssier 2002)
Gravity is solved with SECOND ORDER MIDPOINT SCHEME
(reduces to 2nd order Leapfrog for constant timestep)

Prediction step:

$$\mathbf{v}^{n+1/2} = \mathbf{v}^{n} - \Delta\phi^{n}\frac{\Delta t^{n}}{2}$$

$$\mathbf{x}^{n+1} = \mathbf{x}^{n} + \mathbf{v}^{n+1/2}\Delta t^{n}$$

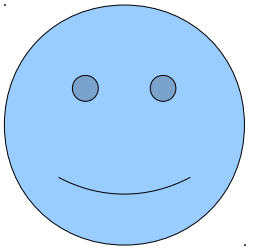


Calculates $\Delta\phi^{n+1}$ and then correction step:

$$\mathbf{v}^{n+1} = \mathbf{v}^{n+1/2} - \Delta\phi^{n+1}\frac{\Delta t^{n}}{2}$$

Leapfrog and midpoint are the same but for Δt – Δt^{n}

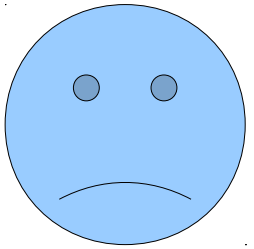
2. ADAPTIVE MESH REFINEMENT:



Why we should refine a mesh

1. if it is prohibitive to sample the entire space with the same (maximum) resolution (too many cells, too much memory)

2. if huge dynamical range is simulated (e.g. regions with density orders of magnitude larger → it is a waste of time to use max. resolution on very low-density regions)



And why we shouldn't

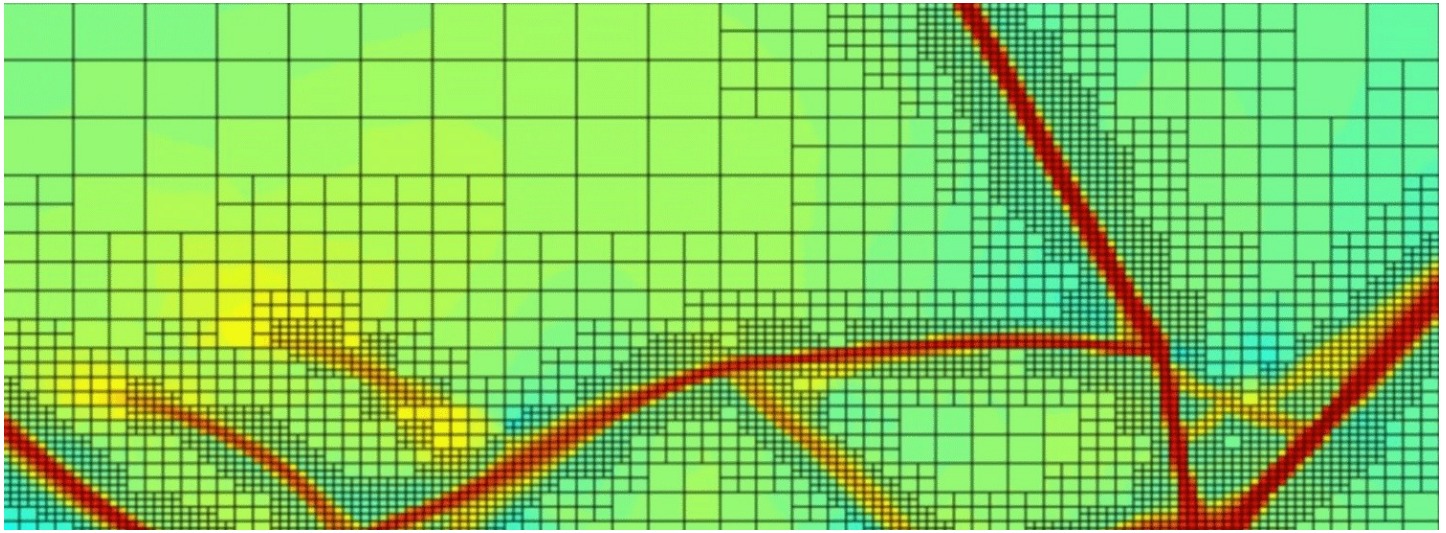
1. refinement algorithms are damn complicated

2. refinement algorithms can generate spurious effects, especially at higher frequency and at boundaries

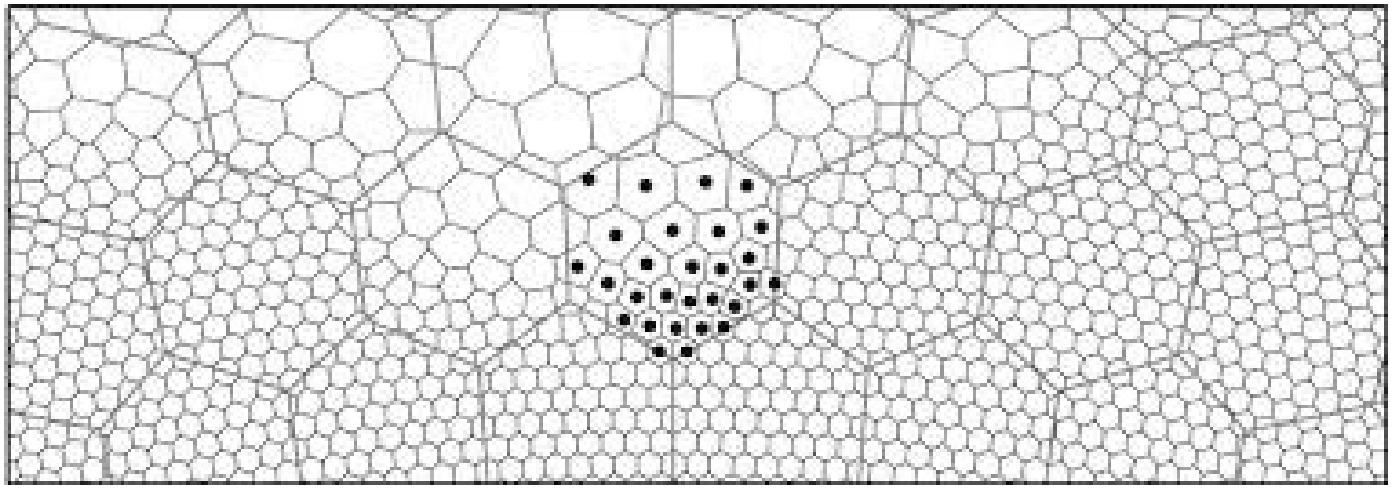
2. ADAPTIVE MESH REFINEMENT:

MAIN REFINEMENT STRATEGIES

- REGULAR (Berger & Oliger 1984; Berger 1986; Berger & Colella 1989):
Computational volume is divided into cubic elements (cells)



- IRREGULAR (Loehner & Baum 1991):
Computational volume is divided into arbitrary-shape cells



2. ADAPTIVE MESH REFINEMENT:

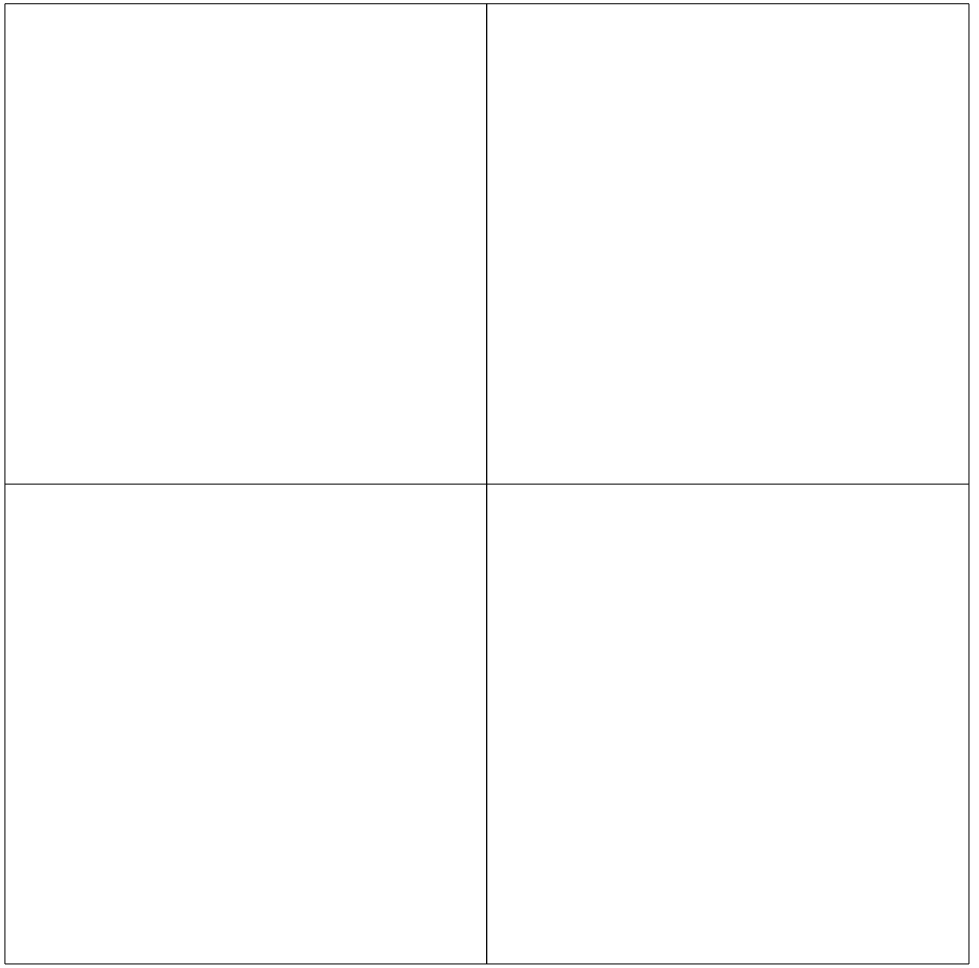
ONE EXAMPLE OF REGULAR AMR: Fully Threaded Tree (FTT)
Khokhlov 1997, Kravtsov+ 1997

similar to a TREE

-root node (COARSE GRID) splits into 8 cubic cells

-cells splits into 8 daughter cells (REFINED GRID) if requested by refinement criterion

-and so on



Coarse grid: 2x2x2

2. ADAPTIVE MESH REFINEMENT:

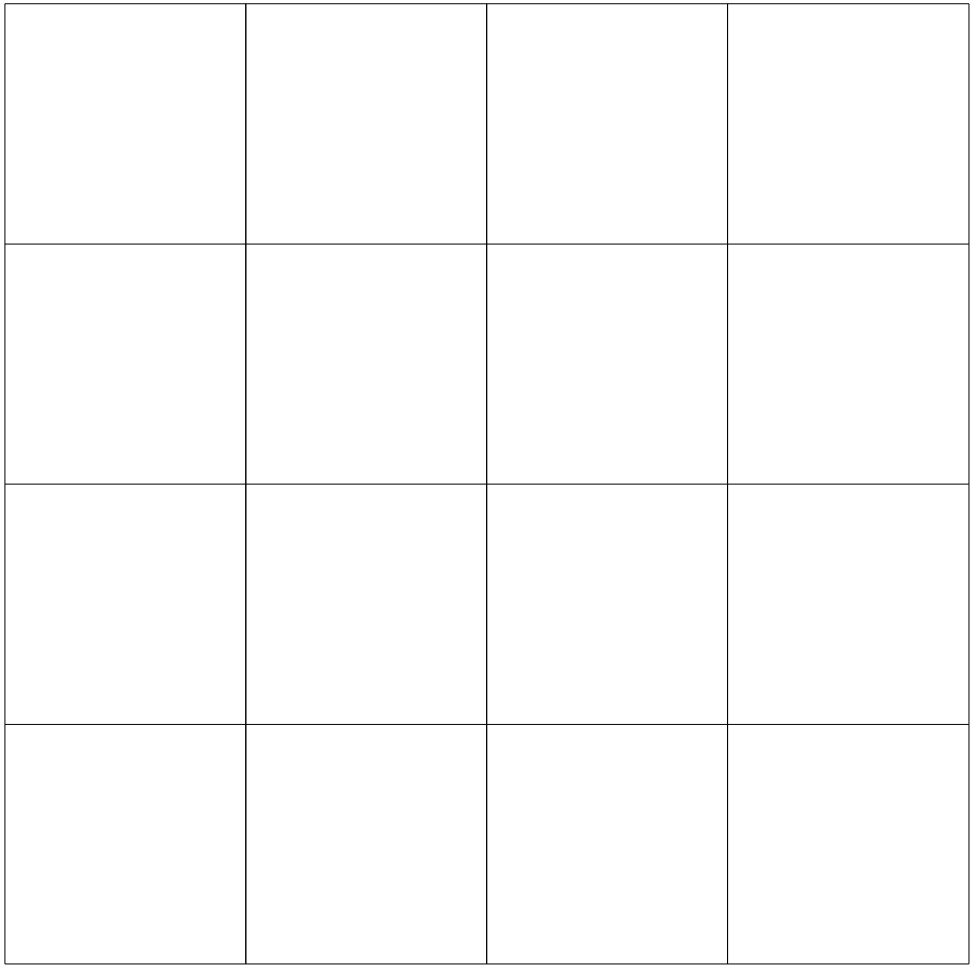
ONE EXAMPLE OF REGULAR AMR: Fully Threaded Tree (FTT)
Khokhlov 1997, Kravtsov+ 1997

similar to a TREE

-root node (COARSE GRID) splits into 8 cubic cells

-cells splits into 8 daughter cells (REFINED GRID) if requested by refinement criterion

-and so on



2nd level: $2^2 \times 2^2 \times 2^2$

2. ADAPTIVE MESH REFINEMENT:

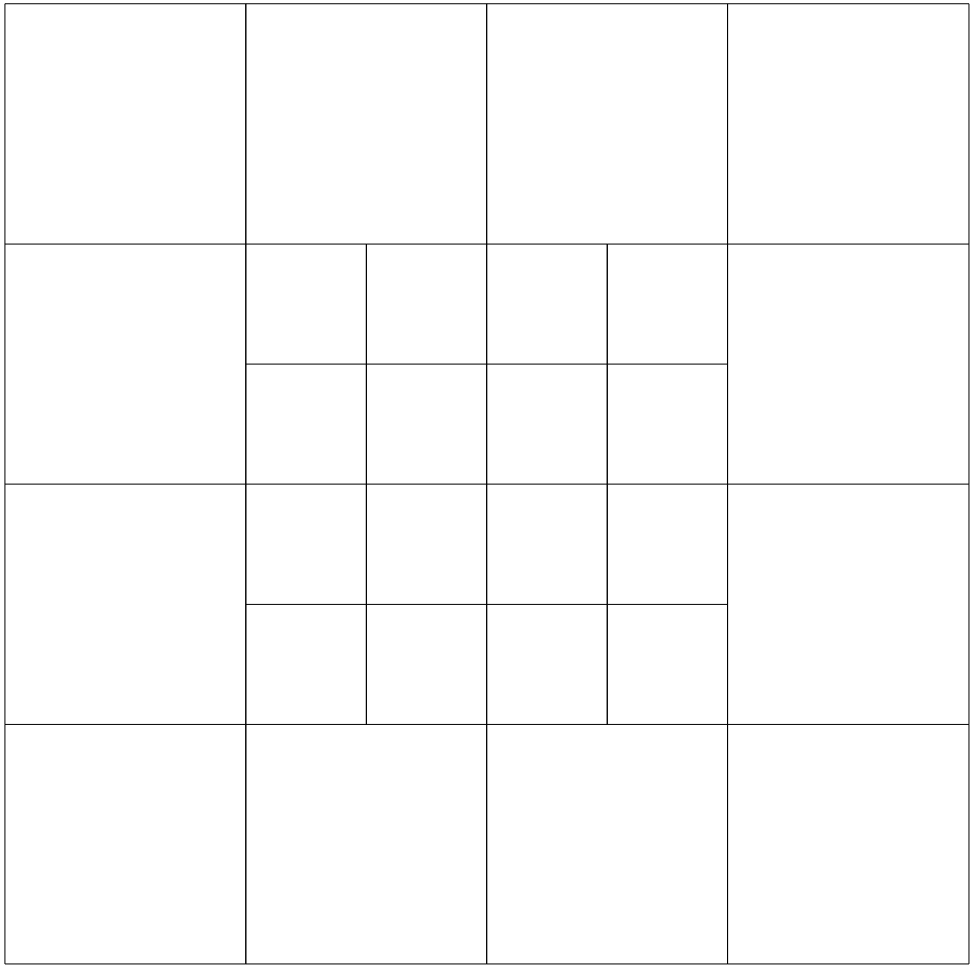
ONE EXAMPLE OF REGULAR AMR: Fully Threaded Tree (FTT)
Khokhlov 1997, Kravtsov+ 1997

similar to a TREE

-root node (COARSE GRID) splits into 8 cubic cells

-cells splits into 8 daughter cells (REFINED GRID) if requested by refinement criterion

-and so on



3rd level: $2^3 \times 2^3 \times 2^3$

2. ADAPTIVE MESH REFINEMENT:

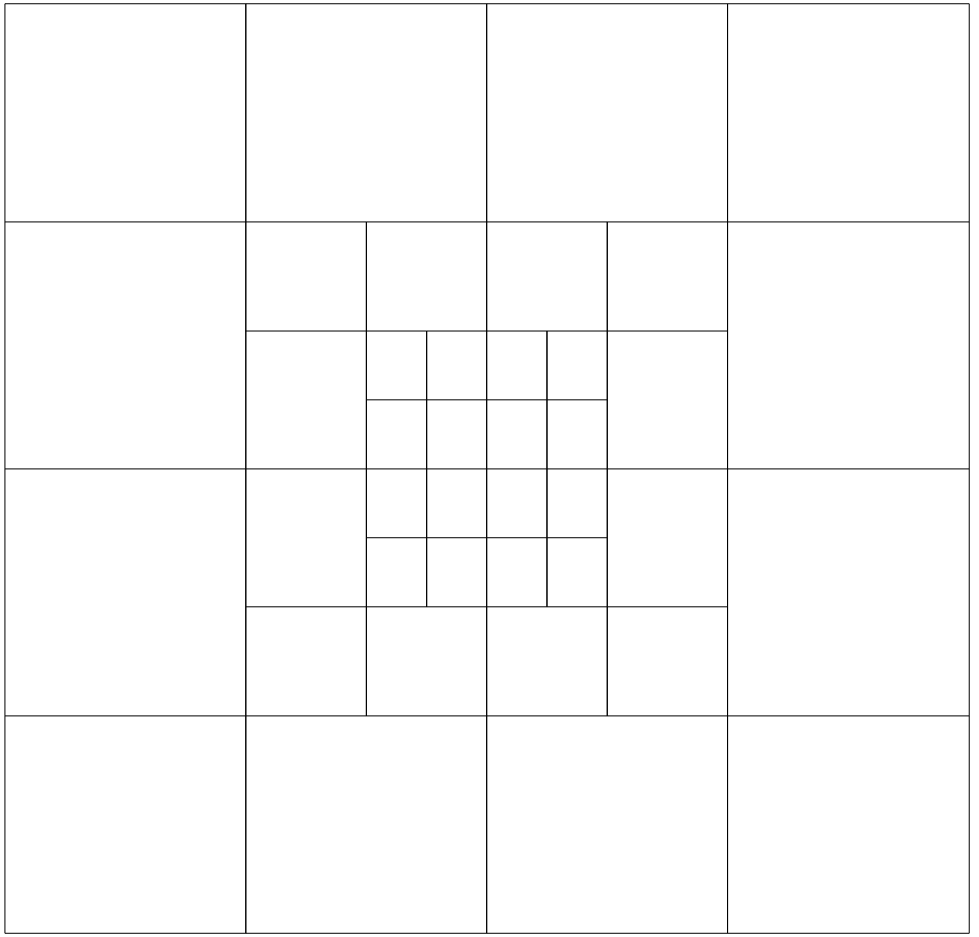
ONE EXAMPLE OF REGULAR AMR: Fully Threaded Tree (FTT)
Khokhlov 1997, Kravtsov+ 1997

similar to a TREE

-root node (COARSE GRID) splits into 8 cubic cells

-cells splits into 8 daughter cells (REFINED GRID) if requested by refinement criterion

-and so on



4th level: $2^4 \times 2^4 \times 2^4$

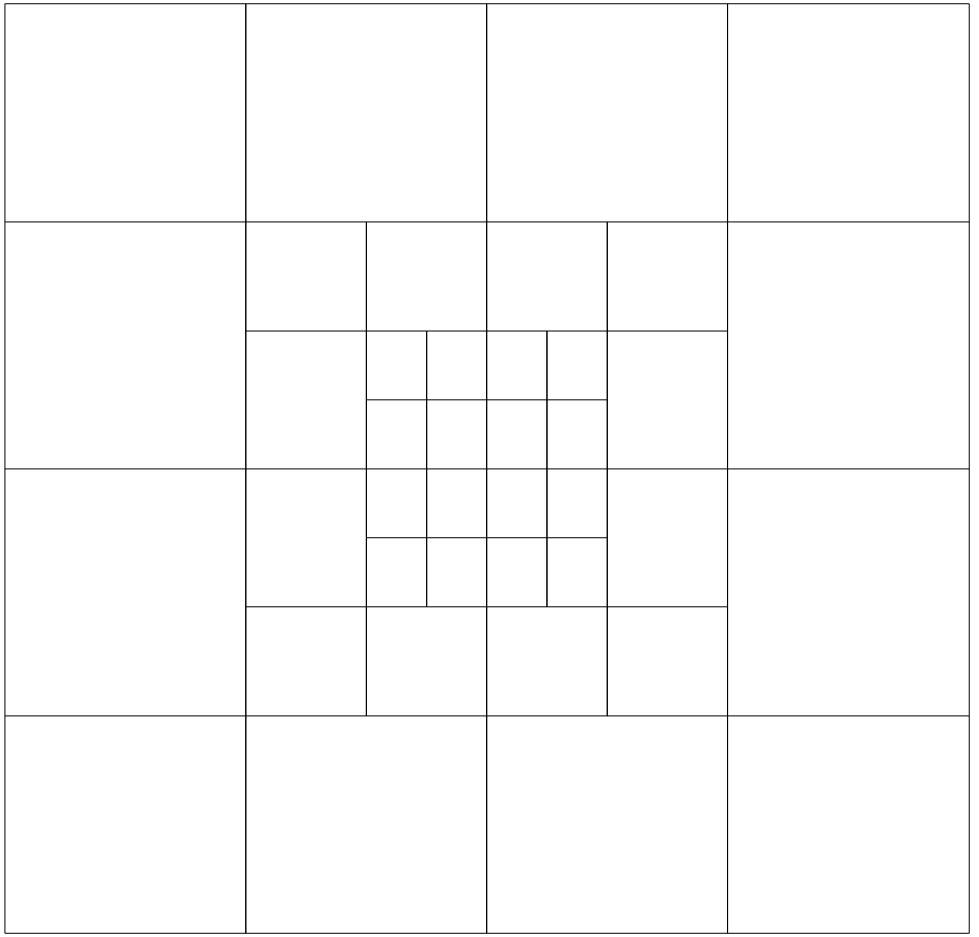
2. ADAPTIVE MESH REFINEMENT:

ONE EXAMPLE OF REGULAR AMR: Fully Threaded Tree (FTT)
Khokhlov 1997, Kravtsov+ 1997

similar to a TREE

REFINED GRIDs can be smaller than the coarse grid and non-connected

but FTT: there cannot be more than 1 level difference between neighboring cells



4th level: $2^4 \times 2^4 \times 2^4$

2. ADAPTIVE MESH REFINEMENT:

WHICH ARE THE REFINEMENT CRITERIA???????

For example:

1- a new level l is created if the baryons mass in a cell exceeds

mrefine(ilevel) * mscale

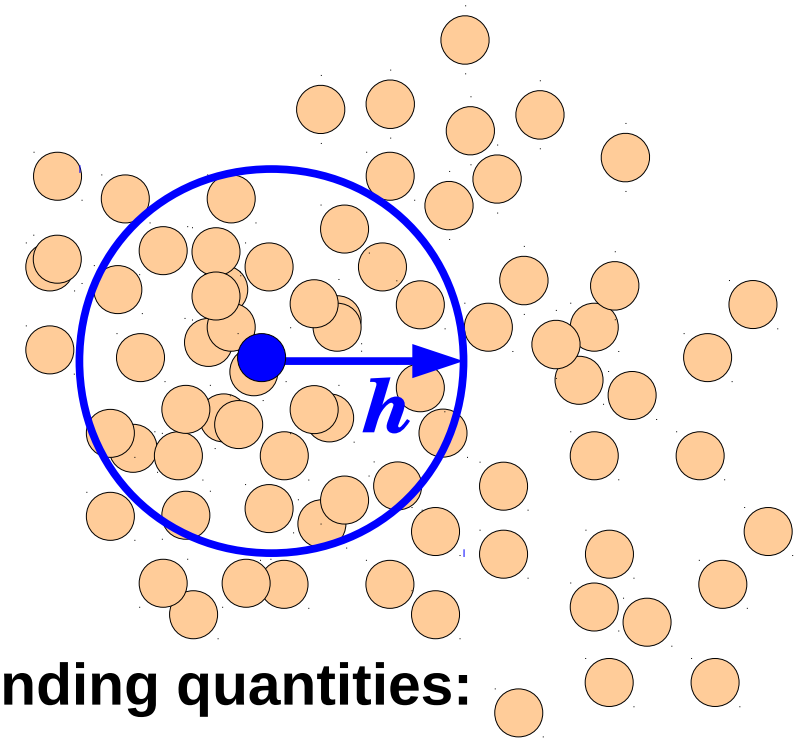
where *mscale* is the mass scale of the simulation and *mrefine* is a multiple of mass scale that depends on the i-level

2- Jeans refinement strategy: each level is refined if the cell size exceeds the local Jeans length divided by *jeans_refine(ilevel)*

3. Smoothed particle hydro-dynamics

Modifies Leapfrog method:

1. define smoothing length *h*:
distance at which a particle has a number N of neighbors (N=32,64,..) similar to softening length but for gas particle properties smoothed on a Kernel



2. calculate Kernel function and corresponding quantities:

$$\langle A(\mathbf{r})\rangle = \int W(\mathbf{r}-\mathbf{r}', h)\, A(\mathbf{r}')d^3r$$

***W*(r, h) = kernel function, *A*(r) can be *ρ*, *P*, *e*, etc**

→ **discretized into** $A_i = \sum_{j=1}^{N} m_j\, W(\mathbf{r}_{ij};\, h_i)\, A_i$

For example density $\rho_i = \sum_{j=1}^{N} m_j\, W(\mathbf{r}_{ij};\, h_i)$

NB: each quantity is defined with respect to the chosen number of neighbors

3. Smoothed particle hydro-dynamics

Modifies Leapfrog method:

3. calculate new dynamical equations (accounting for Euler equations):

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i$$

$$\frac{d\mathbf{v}_i}{dt} = -\frac{\nabla \mathbf{P}_i}{\rho_i} + \mathbf{a}_i^{\rm visc} - \nabla\phi_i$$

P and ρ both smoothed on the Kernel function

Leapfrog has no viscosity but fluid has viscosity: ARTIFICIAL VISCOSITY TERM

Acceleration: same calculation as in no-gas simulations

4. scheme is Lepfrog with additional terms beyond gravity

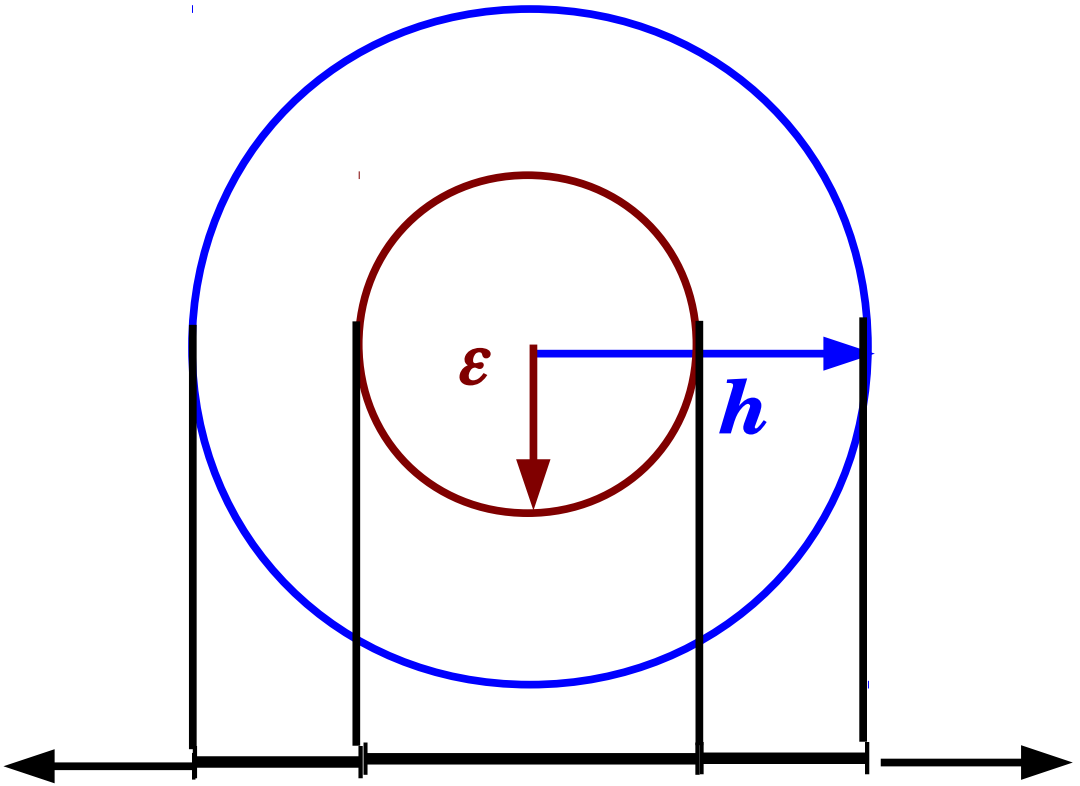
3. Smoothed particle hydro-dynamics

Which is the best choice for SMOOTHING LENGTH?

Smoothing length must be ~ softening length (Bate & Burkert 1995)

If $h < \varepsilon$ → pressure stronger than gravity → spurious expansion

If $h > \varepsilon$ → gravity stronger than pressure → spurious collapse



Neither P nor gravity are smoothed: Both are well described

Only P is smoothed Gravity dominates

Both P and Gravity are smoothed: they balance

Only P is smoothed Gravity dominates

Neither P nor gravity are smoothed: Both are well described

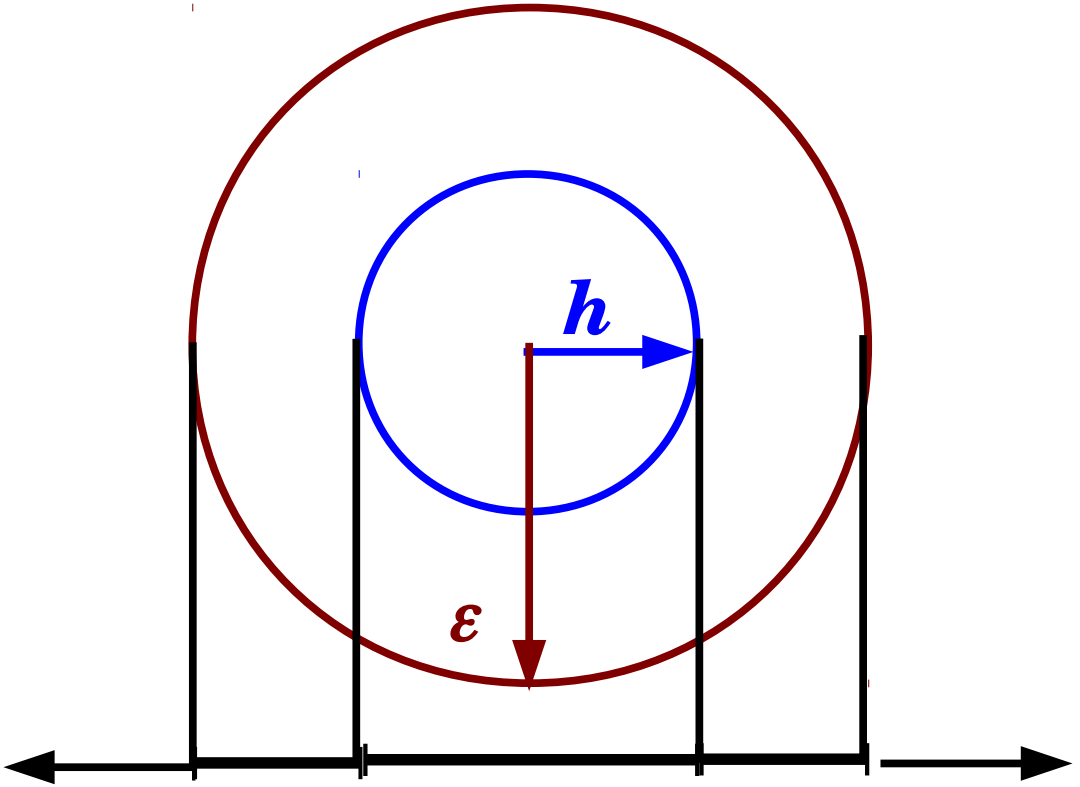
3. Smoothed particle hydro-dynamics

Which is the best choice for SMOOTHING LENGTH?

Smoothing length must be ~ softening length (Bate & Burkert 1995)

If $h < \varepsilon$ → pressure stronger than gravity → spurious expansion

If $h > \varepsilon$ → gravity stronger than pressure → spurious collapse



Neither P nor gravity are smoothed: Both are well described

Only Gravity is smoothed Pressure dominates

Both P and Gravity are smoothed: they balance

Only Gravity is smoothed Pressure dominates

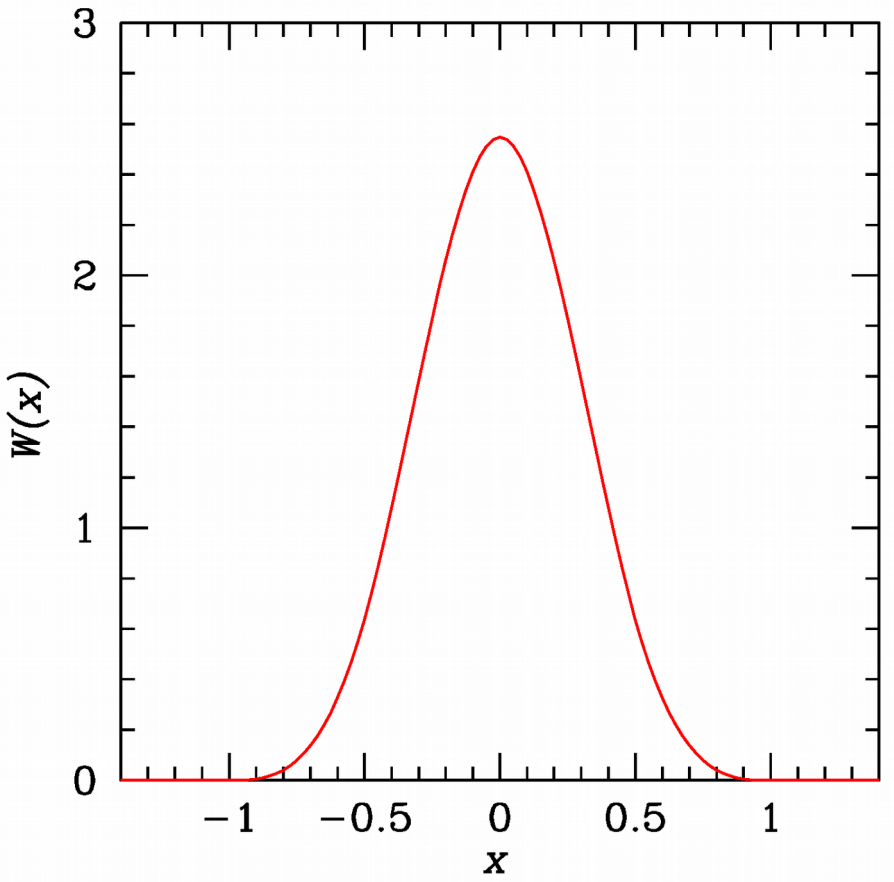
Neither P nor gravity are smoothed: Both are well described

3. Smoothed particle hydro-dynamics

Kernel functions?

- must be normalized to unity
- must be simple
- high order interpolation
  for accuracy
- spherical symmetry
  (for angular momentum
  conservation)

e.g. CUBIC SPLINE



$$W(x) = \frac{8}{\pi} \begin{cases} 1 - 6\,x^2 + 6\,|x|^3 & \text{if } 0 \le |x| \le 1/2 \\ 2\,(1 - |x|)^3 & \text{if } 1/2 < |x| \le 1 \\ 0 & \text{if } |x| > 1 \end{cases}$$

4. Some good reasons for choosing SPH / AMR

SPH PRO:
- simple
- refines automatically

SPH CONS:
- numerical viscosity is dangerous
- does not solve (strong) shocks

→ **Use it if no shocks and interested mainly in dynamics**

AMR PRO:
-no numerical viscosity
-solves shocks pretty well

AMR CONS:
-difficult to use
-requires lot of memory
-refinement criteria are dangerous:
can induce spurious effects

→ **Use it if shocks and interested mainly in fluid equations**

5. EXAMPLE CODE FOR SPH: gasoline

PARAMETER FILE (first part as in lecture 5, then add gas):

```
bPeriodic = 0    # periodic boundaries: use only for cosmo
bParaRead = 0    # read input file in parallel (usually safe)
bParaWrite = 0    # write outputs in parallel

nSteps = 20      #number of time steps
iStartStep = 0   # starting step number of simulation (for output numbering)
dDelta  = 0.01 # length of base timestep ("Rung 0") in Nbody units

achInFile  = ./cond_init_DM.std #input file
achOutName = ./out  #output file

dTheta  = 0.7  # opening angle for gravity (standard value)

iCheckInterval  = 1      # do a checkpoint every N timesteps
lOutInterval    = 1      # print an output every N timesteps
dExtraStore     = 3      # extra memory storage per node

bKDK          = 1        # kick-drift-kick, leave like this
iMaxRung      = 15       # max number of "rungs", i.e. sub-timestep
```

5. EXAMPLE CODE FOR SPH: gasoline

PARAMETER FILE (first part as in lecture 5, then add gas):

```
bStandard      = 1        # snapshots in standard binary
bDoGravity     = 1        # ok

dMsolUnit      = 2.225e5 # mass scale in Msun (for Nbody units)
dKpcUnit       = 1.0 # length scale in kpc (for Nbody units)
                         #→ with this choice timescale is 1 Gyr


bVDetails=1              # write verbose stdout
```

5. EXAMPLE CODE FOR SPH: gasoline

PARAMETER FILE (first part as in lecture 5, then add gas):

```
dEta              = 0.15      # time step criterion for gas
dEtaCourant       = 0.3       # accuracy of SPH

bViscosityLimiter =1      # limit artificial velocity with the Balsara
                          #switch (never remove!)
bBulkViscosity =0         # add bulk viscosity (0 means no)
bDoDensity     =1         # calculate density
nSmooth        =64        # number of neighbors to calculate kernel


bGasCooling     = 0      # cooling (0/1 means without\with cooling)
dMeanMolWeight  =2.46 # molecular weight (here for molecular cloud)
dConstGamma     =1.001 # gamma of equation of state
dCoolingTmin    = 10.0  # minimum temperature of gas
```

5. EXAMPLE CODE FOR SPH: gasoline

Analysis tool:

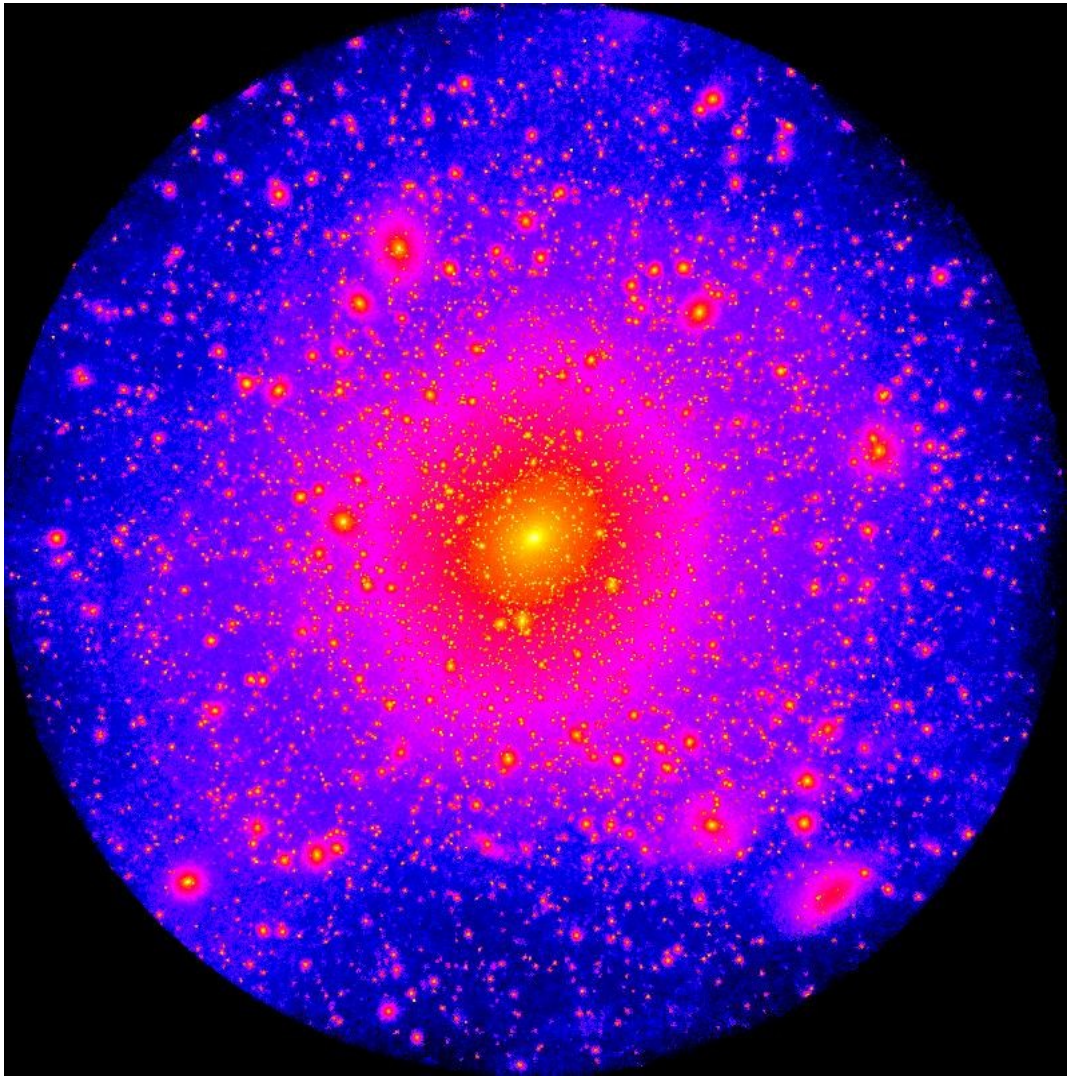
TIPSY
http://www-hpcc.astro.washington.edu/tools/tipsy/tipsy.html

Installation:
follow readme on the website

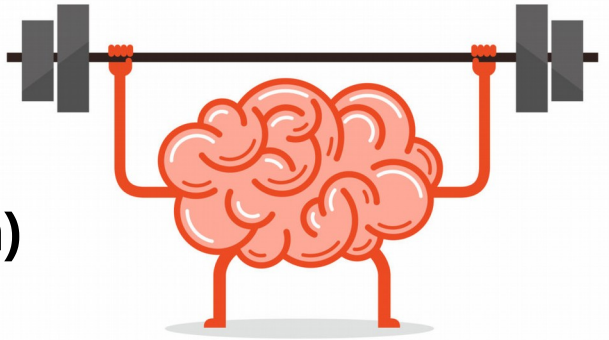
Usage:
Type tipsy on the terminal

openb namefile
loads 1
xall
boxstat 0 all

and so on..



Exercise # 15:



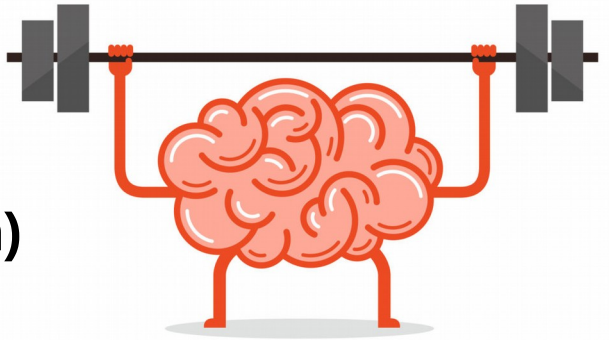
RUN A BLOB OF GAS WITH UNIFORM DENSITY
(velocity field is Gaussian, with a power spectrum)
10^5 gas particles for 5 Myr

mpirun -np 2 ./gasoline ./parameterfile_gas.par

visualize with tipsy

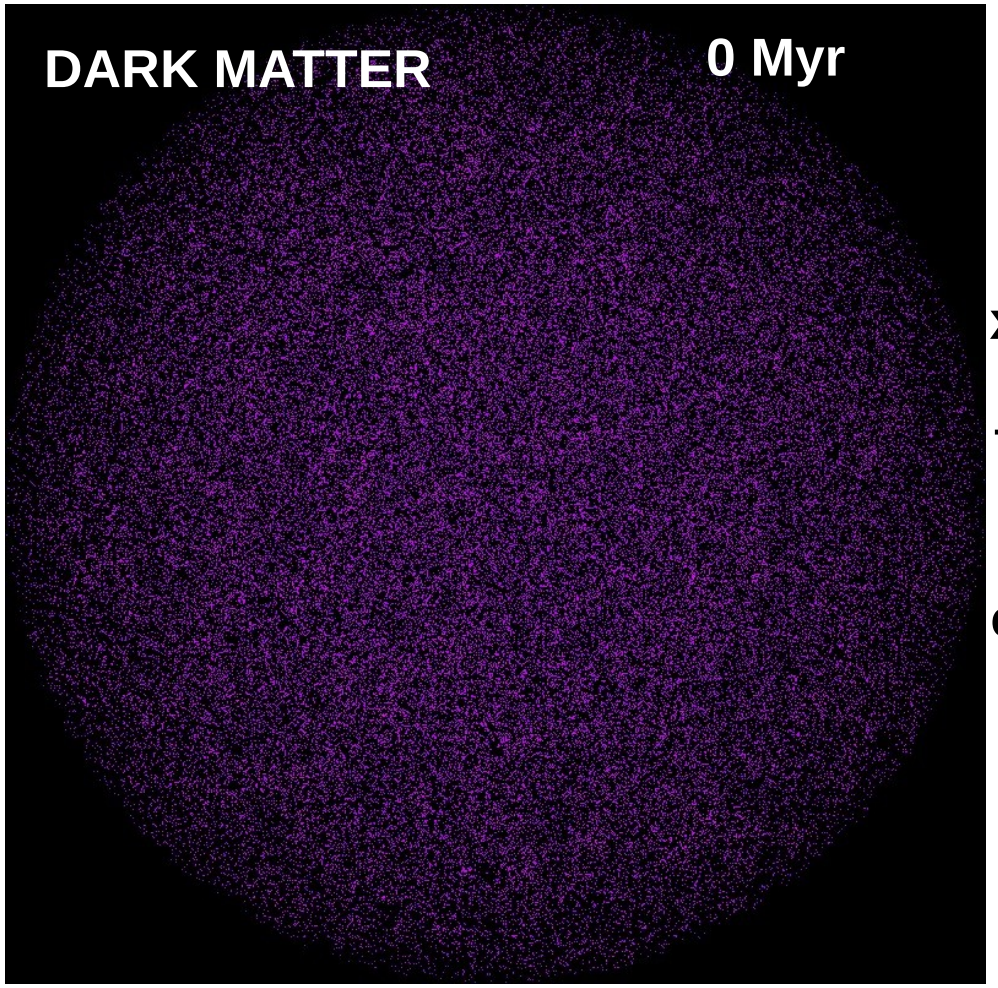
```
openb out.000020
loads 1
zall                                  # xall or yall depending on projection
boxstat 0 all
viewgas logrho 2 6                    # to see projected density in code units,
                                          log scale
hard movie out.20                     # create an image in tipsy image format
shell convert out.20 out.20.jpg  # convert image to jpg
shell rm out.20                       #remove tipsy image
```

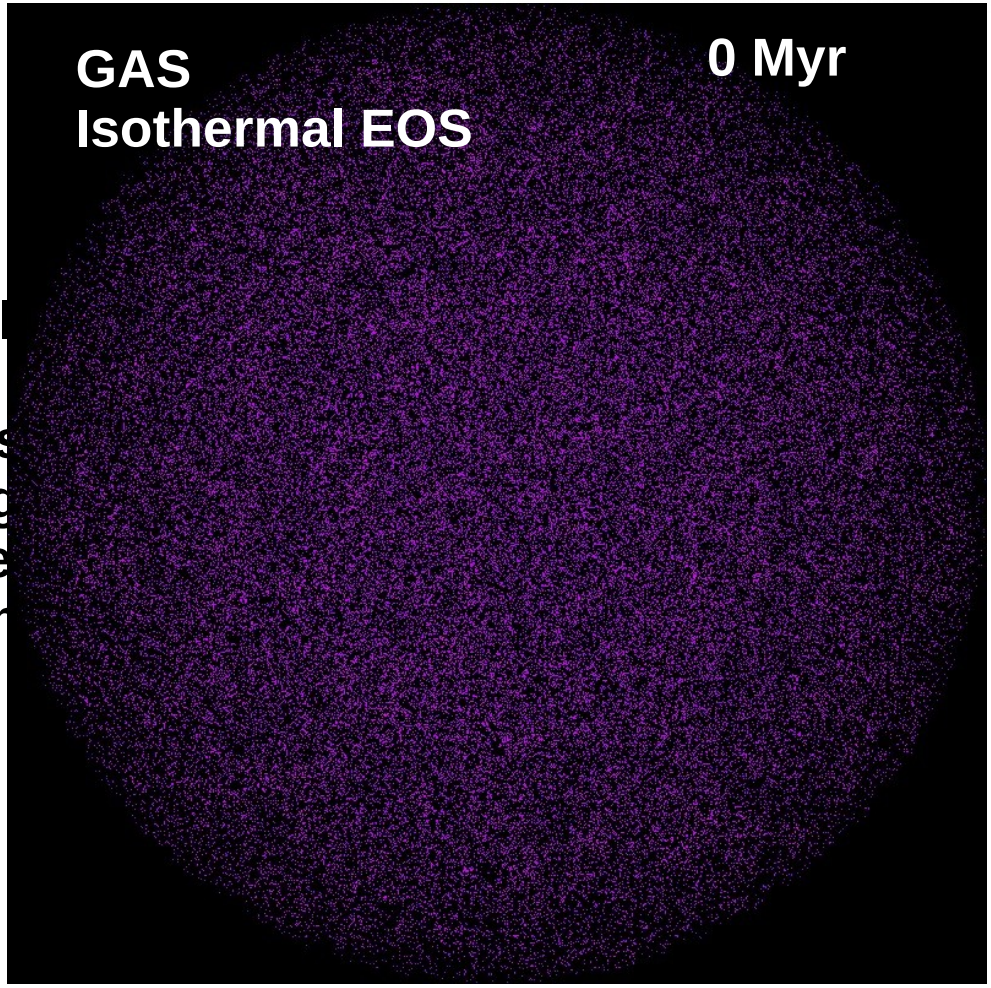
Exercise # 15:



RUN A BLOB OF GAS WITH UNIFORM DENSITY
(velocity field is Gaussian, with a power spectrum)
10^5 gas particles for 5 Myr

mpirun -np 2 ./gasoline ./newpar_gas.par

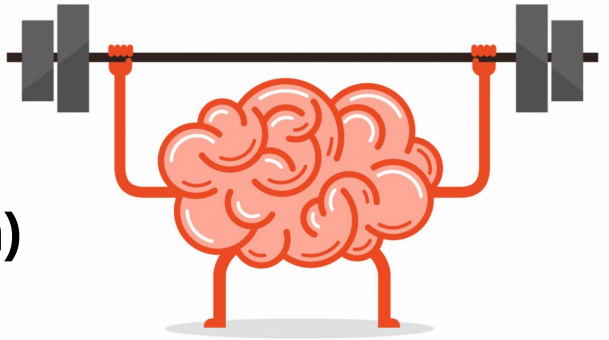




xall

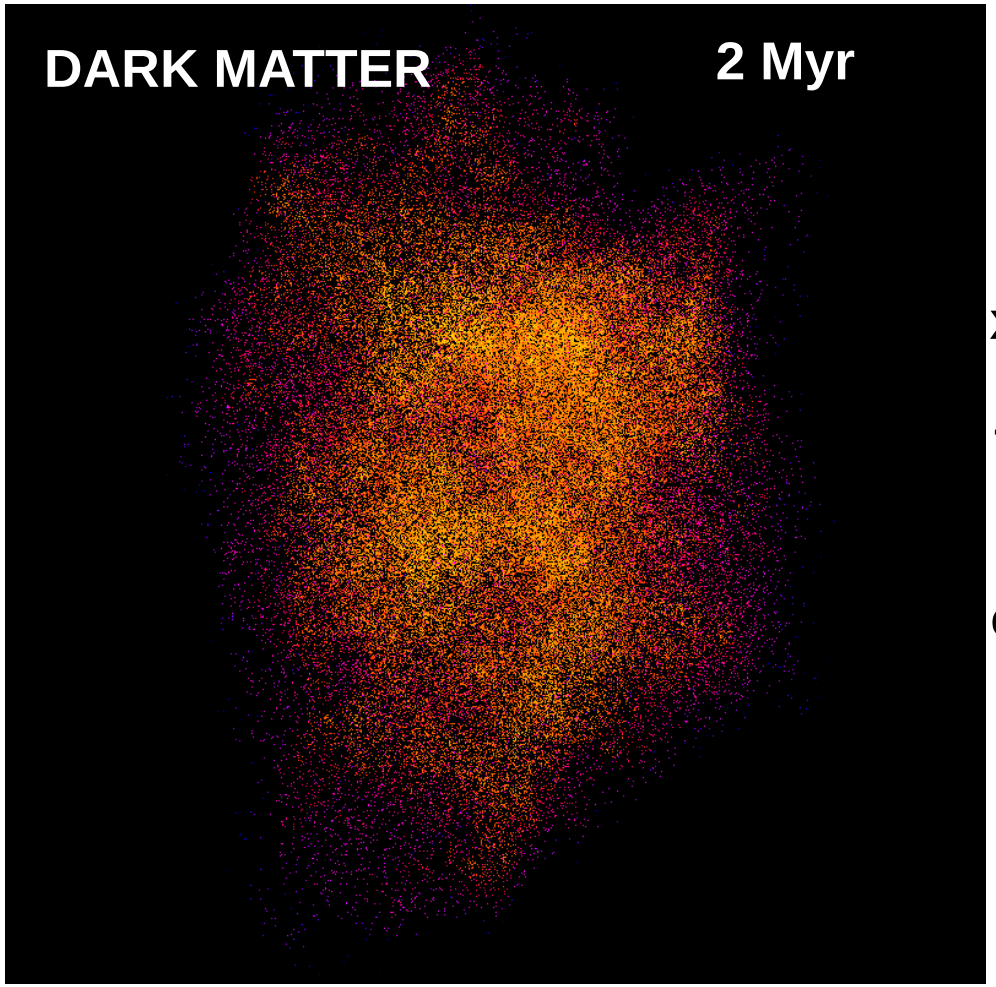
to s
log
cre
cor

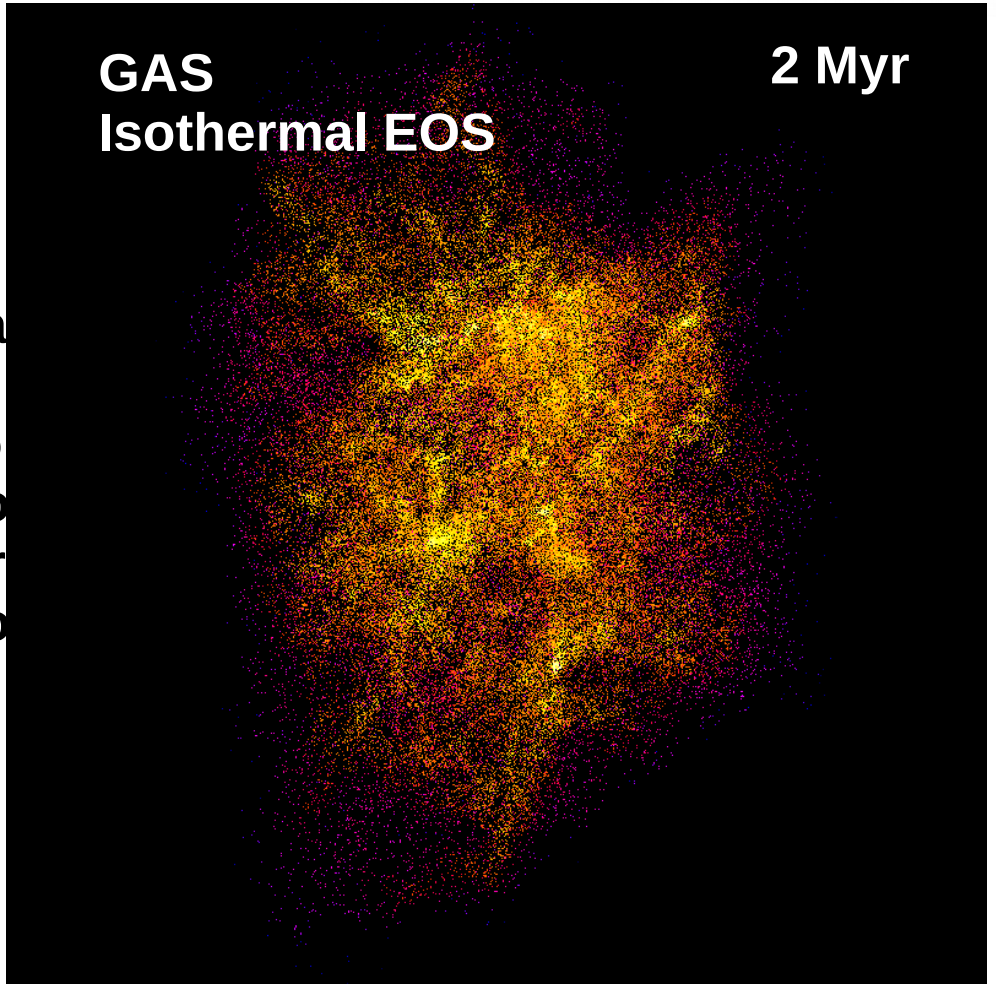
Exercise # 15:



RUN A BLOB OF GAS WITH UNIFORM DENSITY
(velocity field is Gaussian, with a power spectrum)
10^5 gas particles for 5 Myr

mpirun -np 2 ./gasoline ./newpar_gas.par





xa

to
lo
cr
co

5. EXAMPLE CODE FOR AMR: RAMSES

TO DOWNLOAD:

git clone https://bitbucket.org/rteyssie/ramses

Creates directory ramses/
with subdirectories
branches/ **particular versions of the code**
trunk/ **main version of the code (amr/ bin/ doc/ hydro/ namelist/ patch/ pm/ poisson/ utils/)**

TO COMPILE:

cd ramses/trunk/ramses/bin

find a Makefile and a Makefile.rt
(standard Makefile and radiative transfer version)

into Makefile set
NDIM = 1 (or 2 or 3) number of dimensions
NVAR = NDIM+2 (2 is necessary to allocate space for density and pressure)
NPRE = 8 (for double precision / 4 for single precision)

5. EXAMPLE CODE FOR AMR: RAMSES

Then choose compiler among the options (MPI, no MPI, different fortran)
For example

F90=gfortran
FFLAGS=-Mpreprocess -DWITHOUTMPI –DNDIM=$(NDIM) –DSOLVER=$(SOLVER)

make

creates executable (ramses1d or ramses2d or ramses3d)

<u>TO RUN:</u>
Simple test:
bin/ramses1d namelist/tube1d.nml

Other test parameter files are in namelist/

5. EXAMPLE CODE FOR AMR: RAMSES

PARAMETER FILE:

```
&RUN_PARAMS
pic=.true.                  #Particle (in cell) solver
poisson=.true.              #Poisson solver
hydro=.true.                #with hydro
nrestart=0                  #restart files
nremap=2                    #redo load-balance every #timesteps
ncontrol=2                  #plot control lines on screen every #timesteps
nsubcycle=2,2,2             #number of sub-timesteps per level
/

&OUTPUT_PARAMS
noutput=8                   #number of outputs
tout=0.00025,0.00050,0.00075,0.00100,0.00125,0.00150,0.00175,0.00200
                            #time of outputs (code units)
/

&INIT_PARAMS
filetype='grafic'           #file format
initfile(1)='./IC64/'       #ICs coarse grid
Initfile(2)='./IC128/'      #ICs next level
/
```

5. EXAMPLE CODE FOR AMR: RAMSES

PARAMETER FILE:

```
&AMR_PARAMS
levelmin=6              #level coarse grid
levelmax=10             #max refinement level
ngridmax=1677726        #max number of grids per refinement level per CPU
npartmax=30500          #max number of particles
boxlen=0.10             #physical length of box in code units
/

&POISSON_PARAMS
gravity_type=0          #0 means self-gravity
epsilon=1.0d-4          #accuracy Poisson solver
/

&HYDRO_PARAMS
gamma=1.666667          #adiabatic gamma
courant_factor=0.75     #courant parameter (Godunov's scheme)
smallr=1.0d-10          #Minimum density to prevent floating
                        exceptions.
smallc=1.0d-10          #Minimum sound speed to prevent floating
                        exceptions.
/
```

5. EXAMPLE CODE FOR AMR: RAMSES

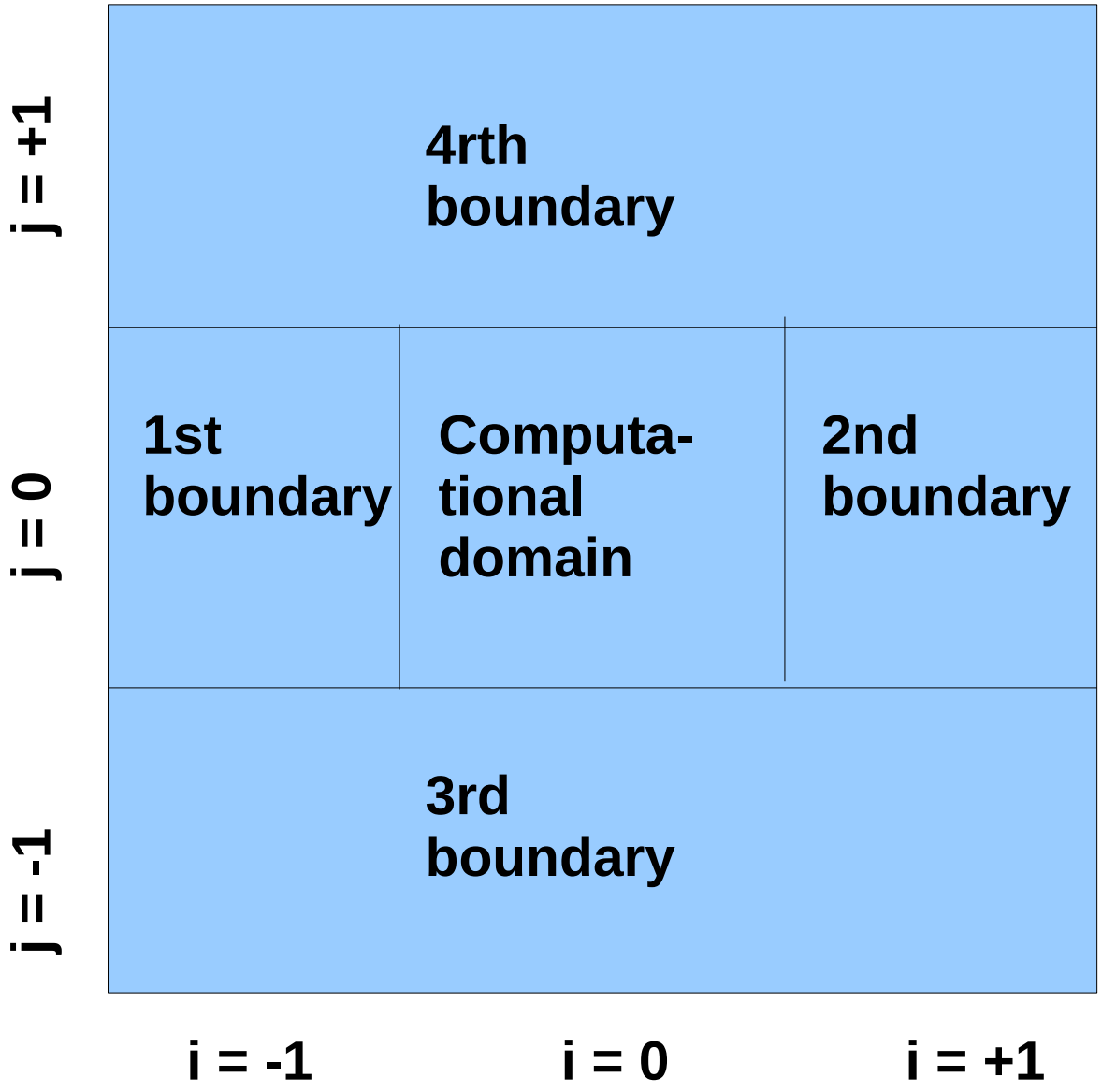
PARAMETER FILE:

```
&PHYSICS_PARAMS
cooling=.true.          #switch on cooling (see next lecture)
/

&REFINE_PARAMS
jeans_refine=4.0        #Jeans refinement criterion
/

&BOUNDARY_PARAMS
boundary=4                          #boundary conditions (2dim)
bound_type= 2, 2,  2,  2            #type of boundary (0 is periodic 1 is
                                    #reflexive, 2 is outflows, 3 is inflow)
ibound_min=-1, 1, -1, -1
ibound_max=-1, 1,  1,  1
jbound_min= 0, 0, -1,  1
jbound_max= 0, 0, -1,  1

/
```

5. EXAMPLE CODE FOR AMR: RAMSES

PARAMETER FILE:

```
ibound_min=-1, 1, -1, -1
ibound_max=-1, 1,  1,  1
jbound_min= 0, 0, -1,  1
jbound_max= 0, 0, -1,  1

/
```



5. EXAMPLE CODE FOR AMR: RAMSES

ANALYSIS:

Files are binaries and difficult to read

→ download PYMSES PACKAGE

http://irfu.cea.fr/Projets/PYMSES/

5. EXAMPLE CODE FOR AMR: RAMSES

EXAMPLE of RUN:

RUN A BLOB OF GAS WITH UNIFORM DENSITY
(velocity field is Gaussian, with a power spectrum)
10^5 gas particles for 4 Myr

mpirun -np 4 ./ramses3d ./parameterfile.nml

or

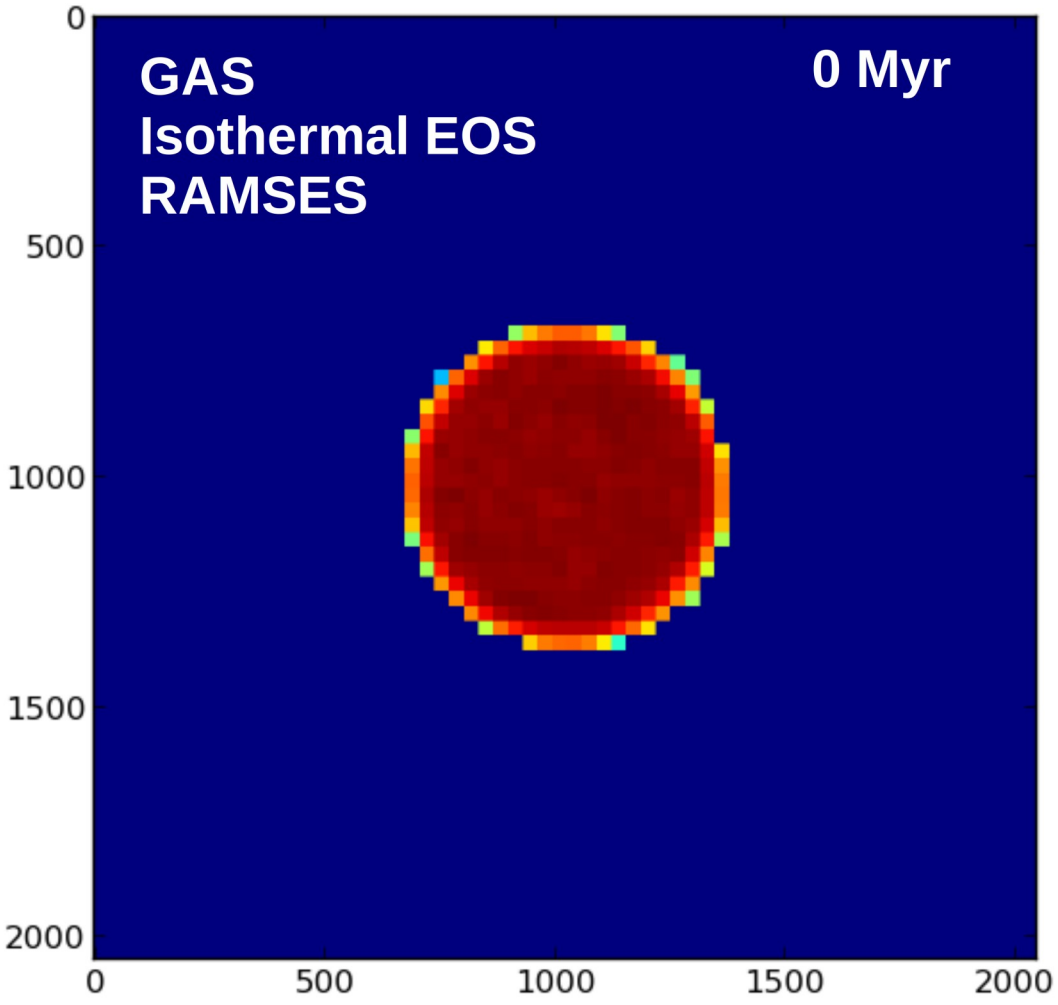
gfortran ./ramses3d ./parameterfile.nml

visualize with pymses, template examplemap.py

5. EXAMPLE CODE FOR AMR: RAMSES

EXAMPLE of RUN:

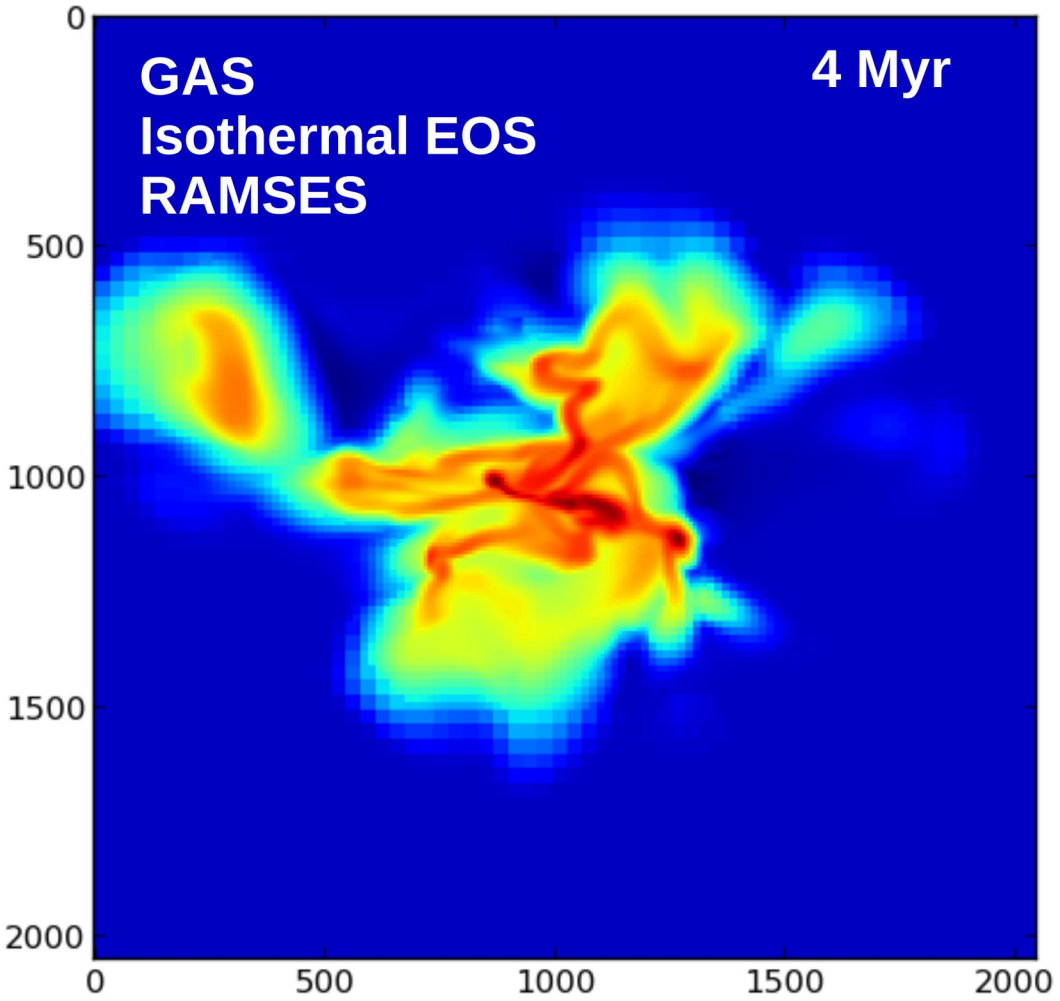
RUN A BLOB OF GAS WITH UNIFORM DENSITY
(velocity field is Gaussian, with a power spectrum)
10^5 gas particles for 4 Myr



5. EXAMPLE CODE FOR AMR: RAMSES

EXAMPLE of RUN:

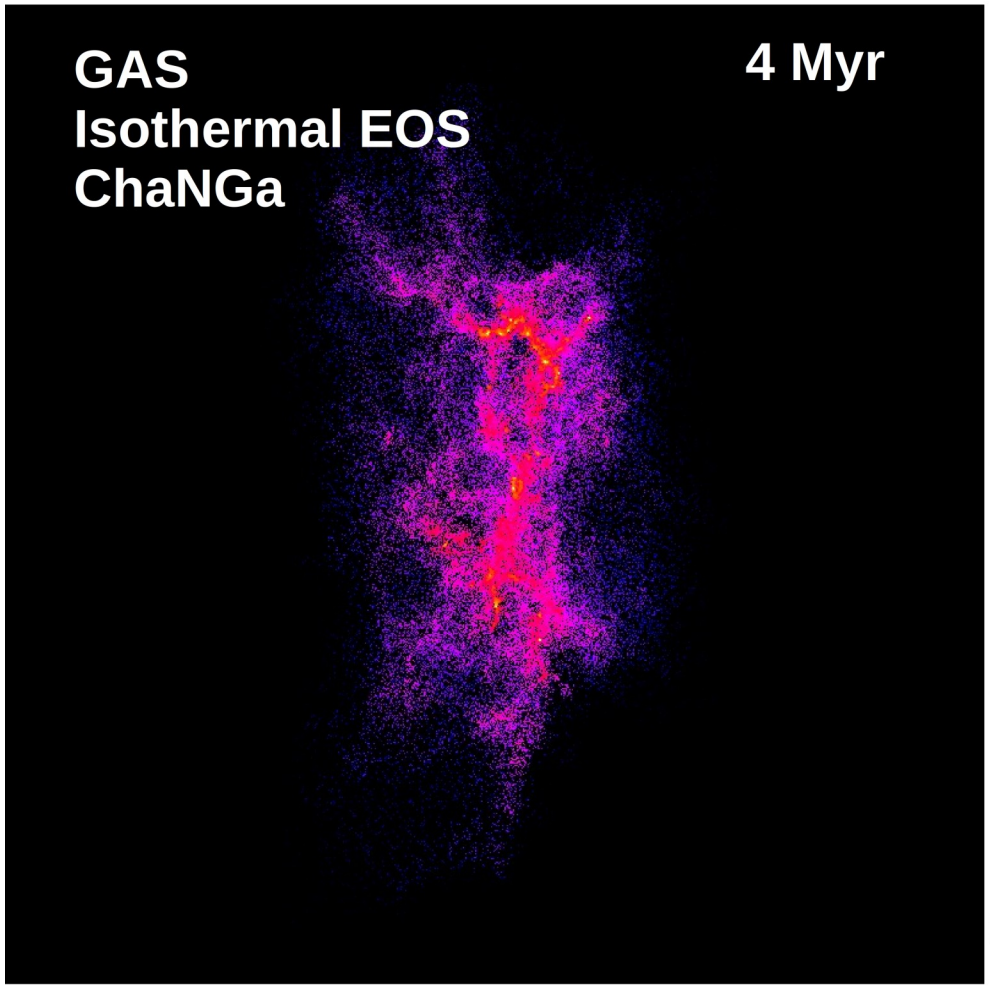
RUN A BLOB OF GAS WITH UNIFORM DENSITY
(velocity field is Gaussian, with a power spectrum)
10^5 gas particles for 4 Myr

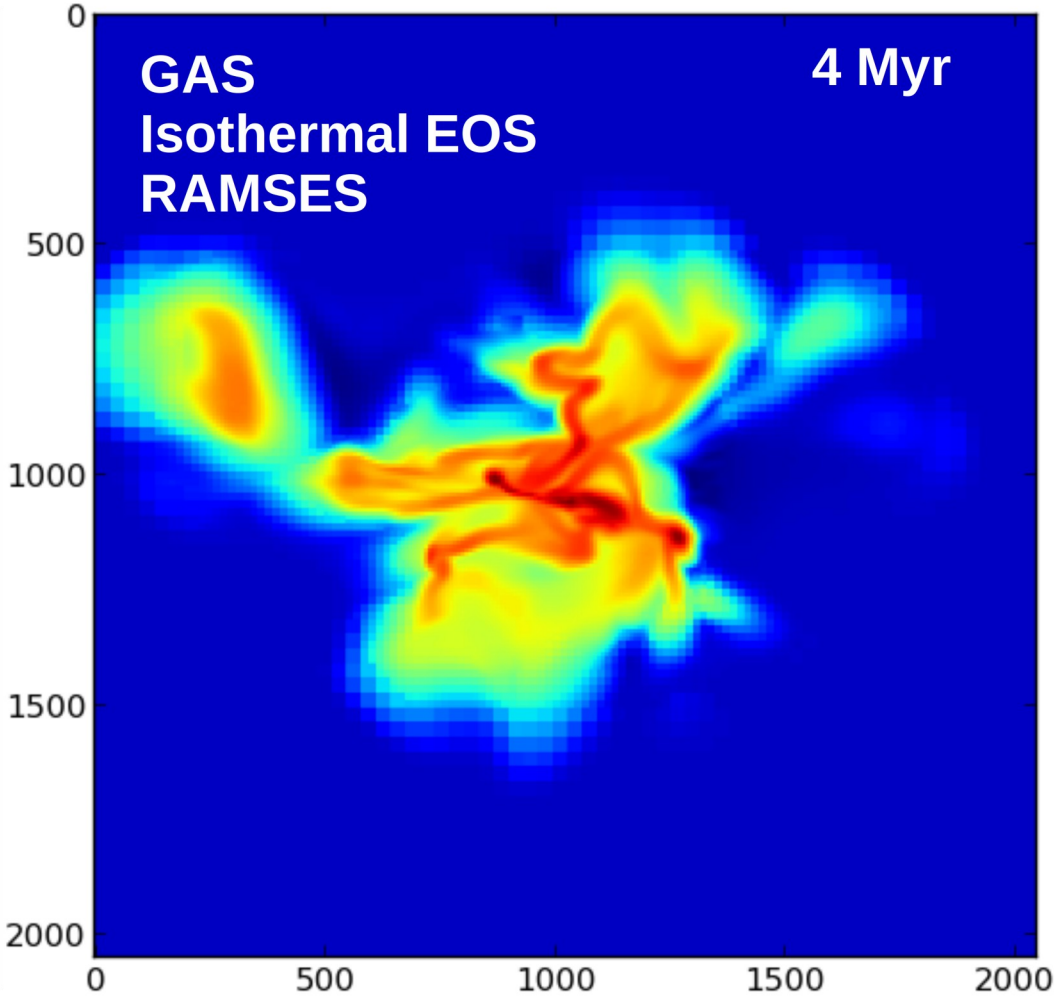


5. EXAMPLE CODE FOR AMR: RAMSES

EXAMPLE of RUN:

RUN A BLOB OF GAS WITH UNIFORM DENSITY
(velocity field is Gaussian, with a power spectrum)
10^5 gas particles for 4 Myr





5. EXAMPLE CODE FOR AMR: RAMSES

EXAMPLE of RUN:

RUN A BLOB OF GAS WITH UNIFORM DENSITY
(velocity field is Gaussian, with a power spectrum)
10^5 gas particles for 4 Myr

Huge difference wrt ChaNGa
REASONS:

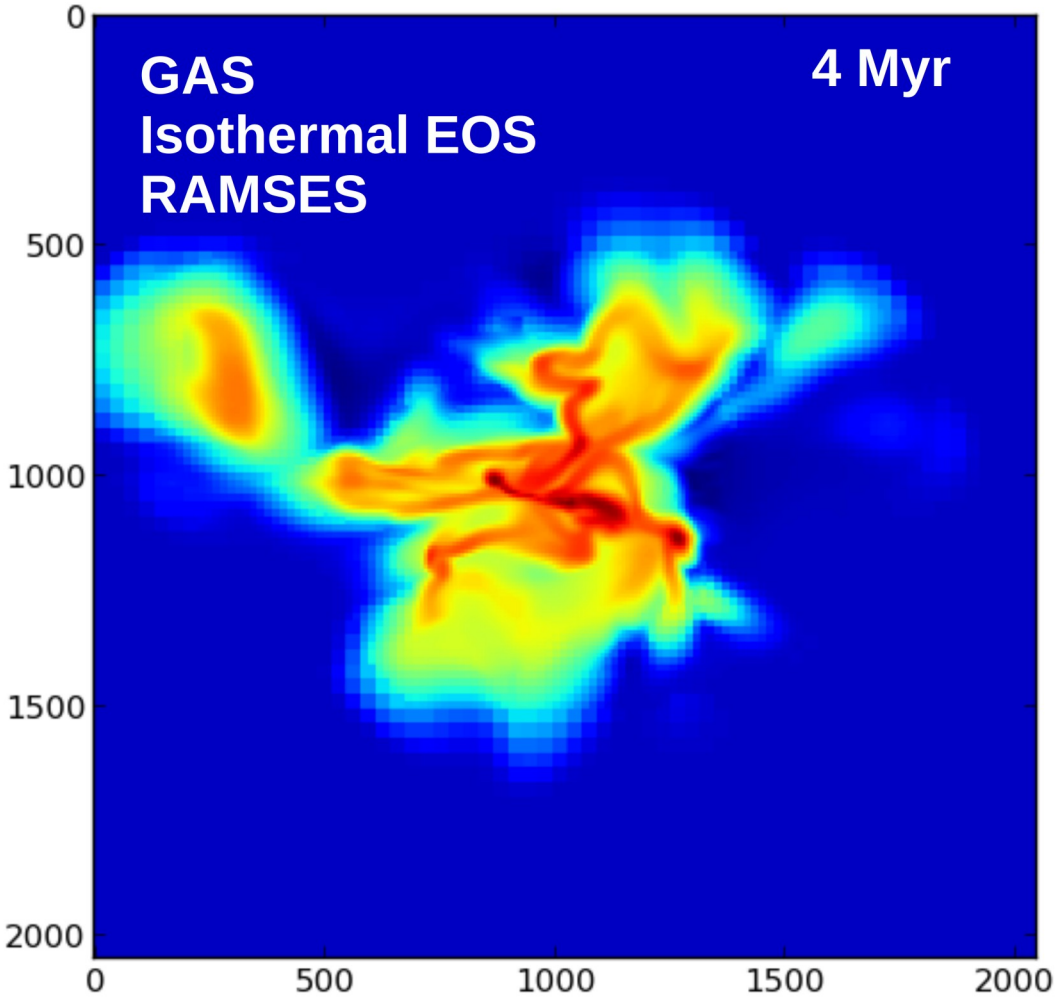
-low res (10^5 ptc, 64^3 grid)

-intrinsic differences:

*** AMR has horror vacui**

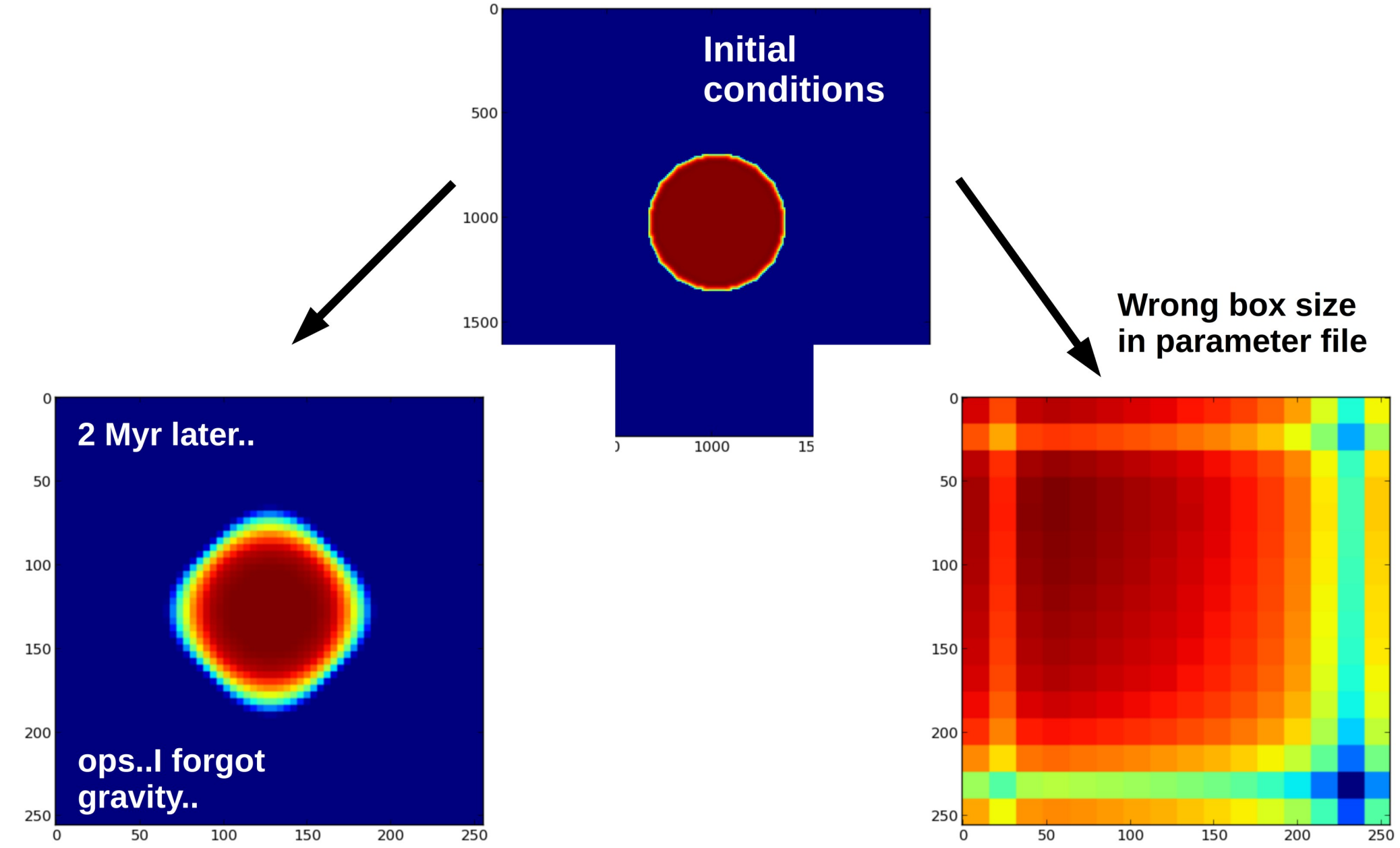
*** viscosity treatment**

*** find out yourself!**



5. EXAMPLE CODE FOR AMR: RAMSES

IF YOU MAKE SOMETHING WRONG WITH AMR CODES YOU IMMEDIATELY FIND OUT...



5. EXAMPLE CODE FOR AMR: RAMSES

IF YOU MAKE SOMETHING WRONG WITH AMR CODES YOU IMMEDIATELY FIND OUT...