Comparison between AMR and SPH

Sam Falle,
Department of Applied Mathematics,
University of Leeds.

David Hubber,
Department of Physics and Astronomy,
University of Leeds.

Simon Goodwin,
Department of Physics and Astronomy,
University of Sheffield.

Aaron Boley,
Department of Astronomy,
University of Florida.

Astrophysical Simulations

Most Astrophysical simulations use either

Eulerian grid codes with Adaptive Mesh Refinement (AMR)
or
Smoothed Partice Hydrodynamics (SPH)

Both methods attempt to solve the fluid equations, but in very different ways.

We would hope that they both give the same (correct) result when the numerical resolution is sufficiently high.

Theoretical arguments can tell us something, but the only way to be sure is to compare them on problems for which we have a good idea of the correct answer.

In this talk I will do this with the upwind AMR code MG and the SPH code SEREN.

SPH

Represent solution by N particles (finite elements) with masses m_i, positions $\mathbf{r}_i$.

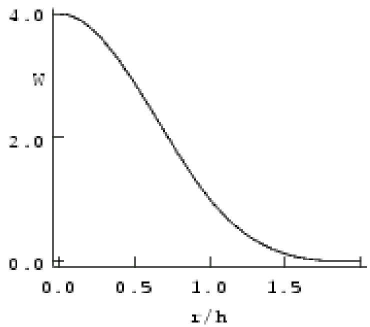
$$\rho(\mathbf{r}) = \sum_{i=1}^{N} m_i W(|\mathbf{r} - \mathbf{r_i}|, h_i) \quad \text{Density}$$

$$\mathbf{u}(\mathbf{r}) = \sum_{i=1}^{N} \frac{\mathbf{u}_i}{\rho_i} m_i W(|\mathbf{r} - \mathbf{r_i}|, h_i) \quad \text{Velocity}$$

$$I(\mathbf{r}) = \sum_{i=1}^{N} \frac{I_i}{\rho_i} m_i W(|\mathbf{r} - \mathbf{r_i}|, h_i) \quad \text{Internal Energy}$$

$W(r, h_i)$ is a spherically symmetric kernel function e.g. cubic spline

$$W(r,h) = \begin{cases} 4\left[1 - \frac{3}{2}\left(\frac{r}{h}\right)^2 + \left(\frac{3}{4}\frac{r}{h}\right)^3\right] & r < h \\ \left[2 - \left(\frac{r}{h}\right)\right]^3 & h < r < 2h \\ 0 & r > 2 \end{cases}$$



Equations

$$\frac{\mathrm{d}\rho_i}{\mathrm{d}t} = \sum_{j=1}^{N} m_j(\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W(|\mathbf{r}_i - \mathbf{r}_j|, h_j) \qquad \text{Continuity Equation}$$

$$\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}t} = \sum_{i=1}^{N} m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W(|\mathbf{r}_i - \mathbf{r}_j|, h_j) \qquad \text{Equation of Motion}$$

$$\frac{\mathrm{d}I_i}{\mathrm{d}t} = \frac{1}{2} \sum_{j=1}^{N} m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W(|\mathbf{r}_i - \mathbf{r}_j|, h_j) \qquad \text{Internal Energy}$$

Here ∇_i means differention w.r.t. $\mathbf{r}_i$.

Note the symmetric form $\Rightarrow$ conservation.

These are not the only possibilities, but they are the ones that are most commonly used.

Need to add artificial viscosity for shocks and artificial thermal conduction for shear layers (see later).

If h_i varies, then need to be more careful to ensure conservation.

Gravity usually done with a multi-pole expansion (tree code) , but could project density onto a grid and use Poisson equation (Particle-mesh code).

Adaptive Mesh Refinement (AMR)

Basic idea of AMR is to refine mesh where solution varies rapidly.

Hierarchical AMR

Hierarchy of grids.

Quadrilateral grid in 2D, hexahedral in 3D i.e. not triangles and tetrahedra.

Ingredients

Criterion for mesh refinement – ideally based on error estimates.

Procedure for coarse-fine grid boundaries.

Different timesteps for different grids (desirable, but not essential).

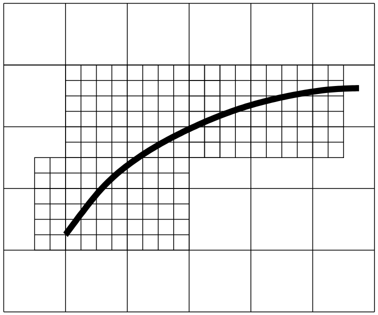
Upwind code (very desirable, but not absolutely essential)

Solution computed on all grids to give error estimate (desirable).

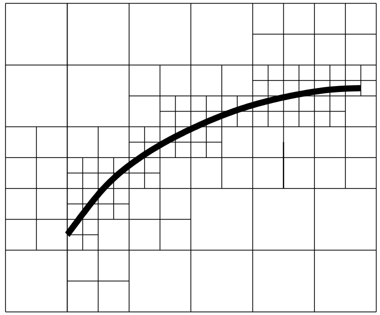
Use MG code, which is upwind, conservative and has different timesteps on different grids. Error estimator uses solution on different grids.

About 20 times faster than FLASH.

Patches versus Cell-by-Cell



Patches



Cell-by-Cell

Patches: AMRITA, ATHENA, CHOMBO, CLAWPACK, ENZO, FLASH, ORION.

Cell-by-cell: RAMESES, KHOKHLOV, MG.

Efficiency (D dimensional calculation)

	Uniform grid	AMR
Memory cost	$\propto (1/\Delta x)^D$	$\propto (1/\Delta x)^{D-1}$
CPU time	$\propto 1/\Delta x)^{D+1}$	$\propto (1/\Delta x)^D$

if regions requiring high resolution are sheets. Even better for filaments or compact objects.

Gravity

Solve the Poisson equation with a full approximation multigrid with a very coarse level 0 grid, combined with Gauss-Seidel relaxation.

Create a "shadow" grid of cells that only contain the potential – speed up memory access since Gauss-Seidel involves little work per cell.

Typically gravity increases cost by

factor	3	Gas dynamics
factor	2	MHD

Need to calculate potential at boundary. At the moment done by direct sum on a coarse grid. This is a bit crude: a multi-pole expansion (tree-code) would be better. Easy to do with AMR grid.

In fact a multi-pole expansion for gravity might be best : easier and faster parallelisation, no worries about coarse/fine boundaries.

Advantages of SPH over AMR

Lagrangian

Galilean invariant $\Rightarrow$ no numerical mass diffusion. This can be a problem for grid codes. Also SPH is grid-free i.e. no preferred directions. But absence of mixing can be a problem.

Large dynamic range in density

More particles where density is large. AMR needs a lot of levels to handle large density contrasts.

No domain

Just follow particles. Grid code needs a domain, which could turn out to be too small.

Conservative

Conserves mass, momentum, energy and angular momentum to rounding if all particles have same timestep.

Note Grid codes conserve mass, momentum and energy to rounding, but not angular momentum.

However, it is usually impractical to use the same timestep for all particles. In that case SPH does not conserve momentum and energy (unlike AMR codes).

Also multi-pole expansion for gravity does not conserve momentum.

Advantages of AMR over SPH

Low convergence rate for SPH.

SPH converges like h^2 if noise $< h^2 \Rightarrow h \equiv$ mesh spacing in a second order grid code.

Let N_n be no of particles in sphere radius $2h$.

If particles are on a lattice noise $\propto 1/N_n \log N_n$.

If random, then noise $\propto 1/\sqrt{N_n}$.

Require $N_n \propto 1/h^2$ (lattice), $1/h^4$ (random) as $h \rightarrow 0$.

$\Rightarrow$ cost per timestep per smoothing volume $\propto 1/h^4$ (lattice) or $\propto 1/h^8$ (random).

For grid code cost per timestep per cell is constant.

In smooth flow error $\propto h^2$ and for 3D get

Uniform grid code	Error $\propto \dfrac{1}{\mathrm{cost}^{1/2}}$.
SPH lattice	Error $\propto \dfrac{1}{\mathrm{cost}^{1/4}}$.
SPH random	Error $\propto \dfrac{1}{\mathrm{cost}^{1/6}}$.

If there are shocks, error $\propto h$ and for 3D get

Uniform grid code	$\text{Error} \propto \dfrac{1}{\text{cost}^{1/4}}.$
SPH lattice	$\text{Error} \propto \dfrac{1}{\text{cost}^{1/6}}.$
SPH random	$\text{Error} \propto \dfrac{1}{\text{cost}^{1/8}}.$

Note most people vary h so that $N_n \simeq$ constant $\simeq 50$. This does not ensure convergence. Very difficult to increase $N_n \rightarrow$ clumping instability!

SPH has limited adaptivity

SPH only adapts to density $\Rightarrow$ poor resolution in low density regions. AMR code can adapt to minimize error.

SPH has inferior shock resolution

SPH needs artificial viscosity for shocks, which causes problems and leads to poor shock-capturing. Upwind grid codes are really good at this.

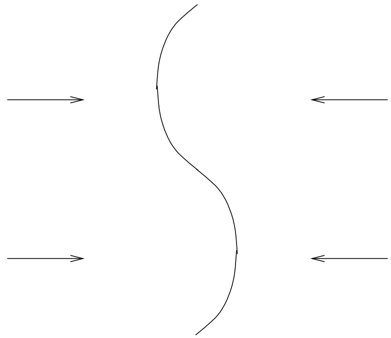
SPH has trouble with magnetic fields

So far impossible to do magnetic fields properly with SPH: there are grid codes that have no difficulty with this!

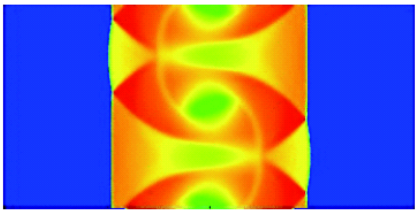
There are other horrors in SPH (see later)

Non-linear Thin Shell Instability

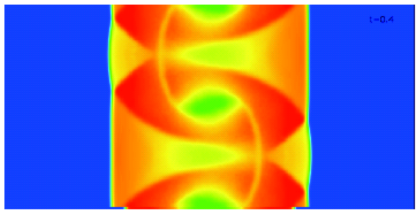
Instability of two colliding flows with a
periodic interface in the
initial conditions (Vishniac 1994).



Uniform Grid MG (800x400)



SPH (240,000 particles)



Density

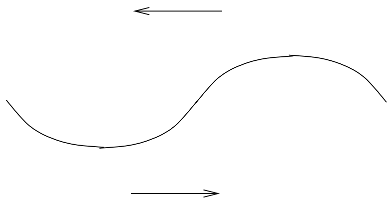
Excellent agreement!

However, very high resolution in both cases and SPH used constant h.

Kelvin-Helmholtz Instability

Instability of a shear layer.

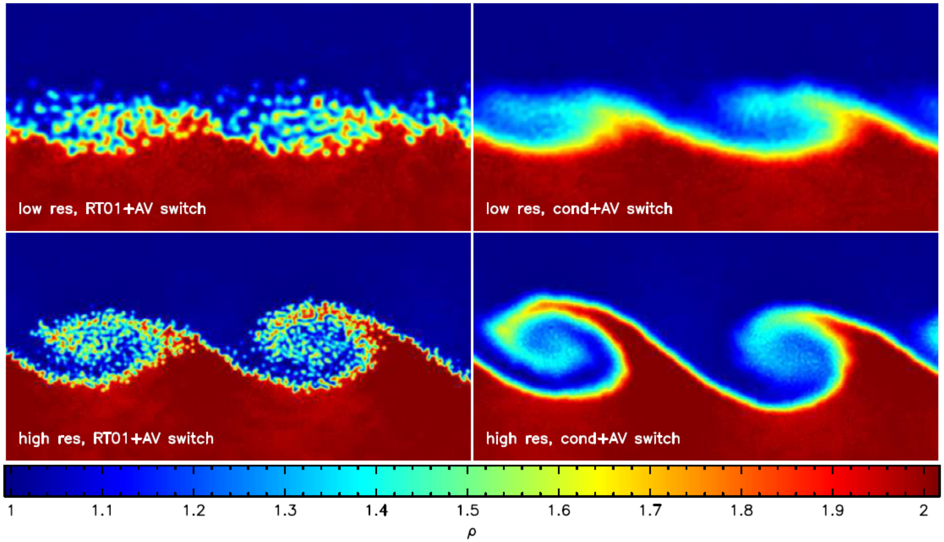
Usual to assume that the contact line has an initial low amplitude sinusoidal perturbation.



This does not lead to a pure mode: should really use the linear eigenfunction.

Standard SPH has problems with K-H instability if the two fluids have different densities (e.g. Agertz et al. 2007).

Price (2008) shows that one can fix this by adding artificial thermal conductivity.
But this is based on pressure gradient.



density contrast of 2

Get the horrid blobs when smoothing length is varied.

Can also use alternative forms of the SPH equations (e.g. Read et al. 2009, Cha et al. 2010).

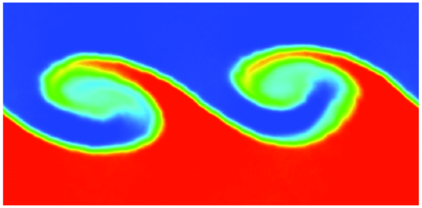
Will look at the same problem (i.e. density contrast of 2).

For SPH will use artificial conductivity and fixed smoothing length.

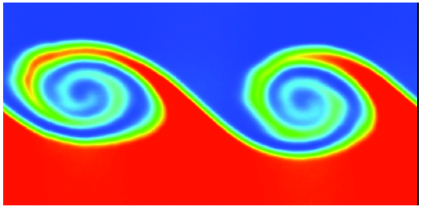
SPH Kernel

Can get better results with a higher order kernel.

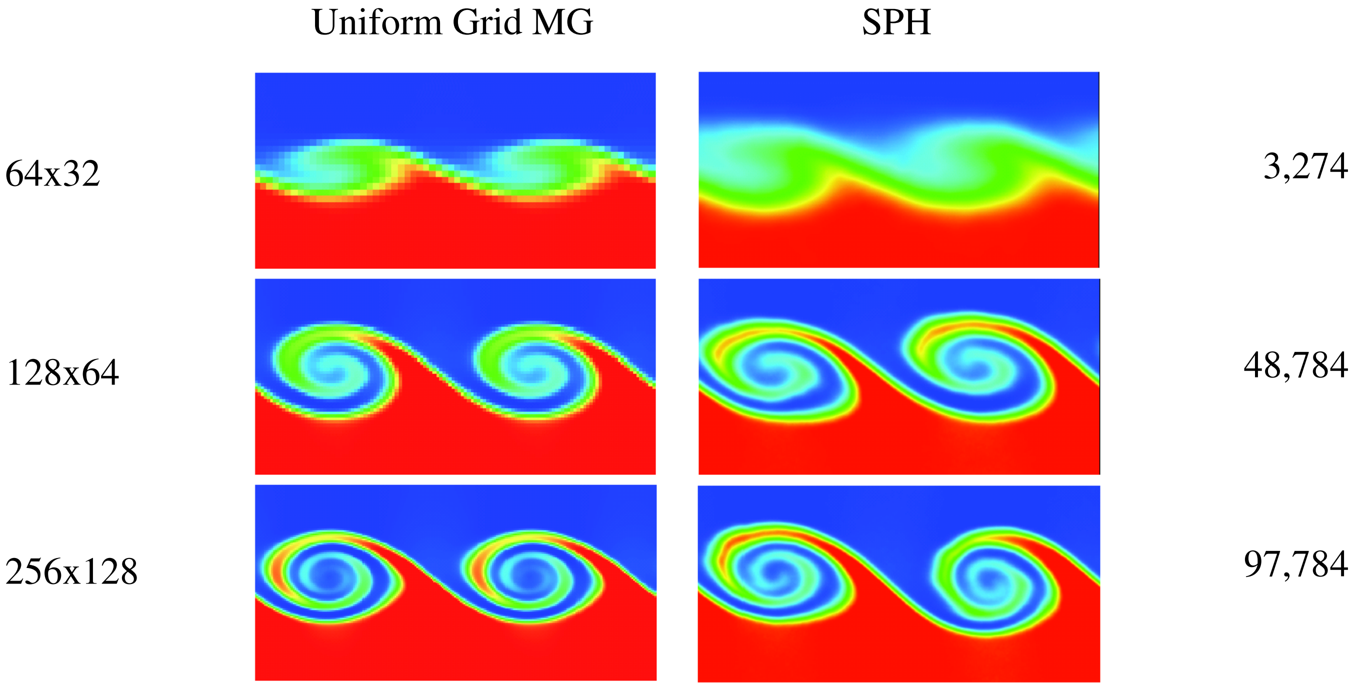
Cubic Kernel



Quintic Kernel



Same number of particles in each case.



Reasonable agreement, but SPH much slower (factor 20).
This would be $\simeq 50$ in 3D (if $N_n = 50$).

Static $n = 1$ Polytrope

As a first gravity test look at a static $n = 1$ polytrope at various resolutions.

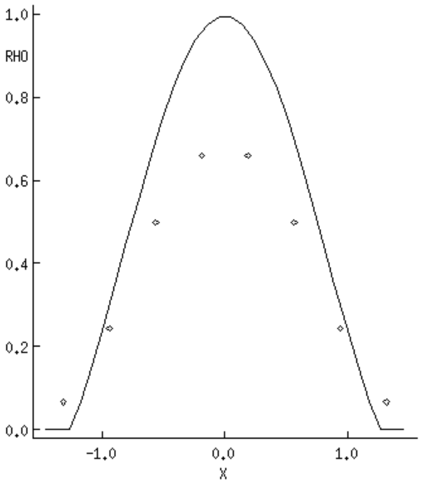
Calculations done in 3D.

Expect the structure to relax to a numerical equilibrium which is different from the exact solution.

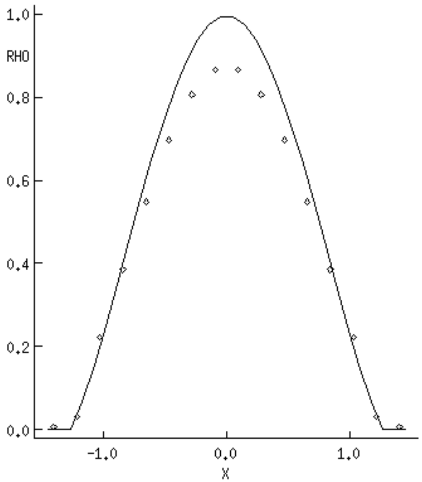
Agreement should get better with increasing resolution.

Uniform Grid MG

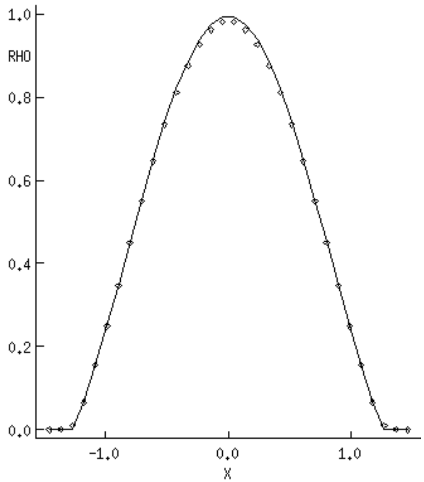
4 x 4 x 4



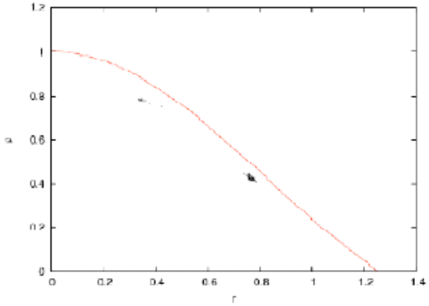
8 x 8 x 8



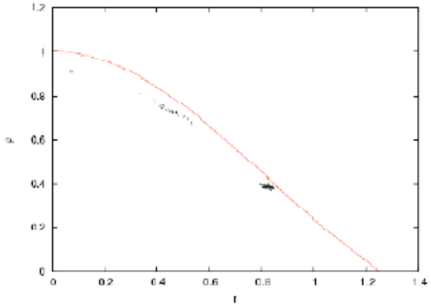
16 x 16 x 16



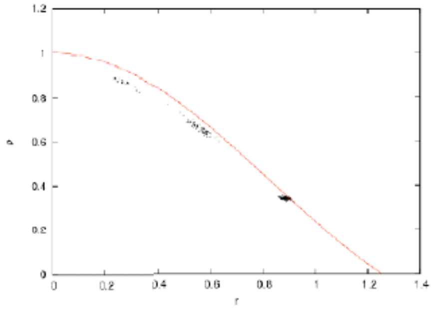
SPH



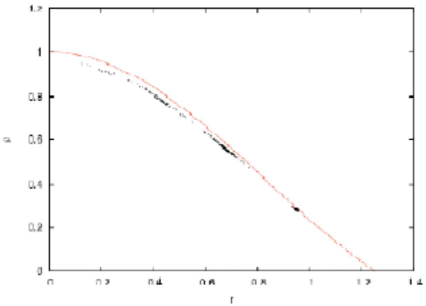
N = 60



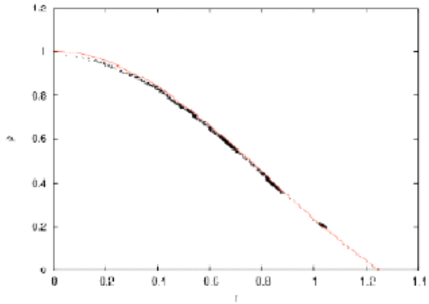
N = 100



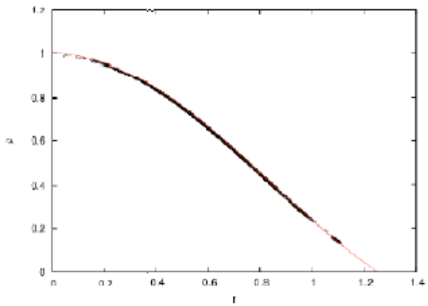
N = 200



N = 400



N = 1600



N = 6400

Moving Gravitationally Bound Objects

Some AMR codes have trouble with moving gravitationally bound objects.

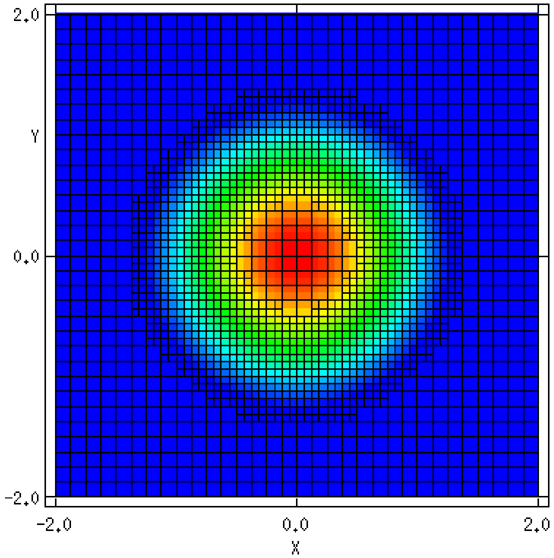
Edgar et al. 2005 showed that with FLASH a uniformly translating object loses momentum.

Gawryszczak et al. 2005 showed that orbiting objects lose angular momentum with FLASH.

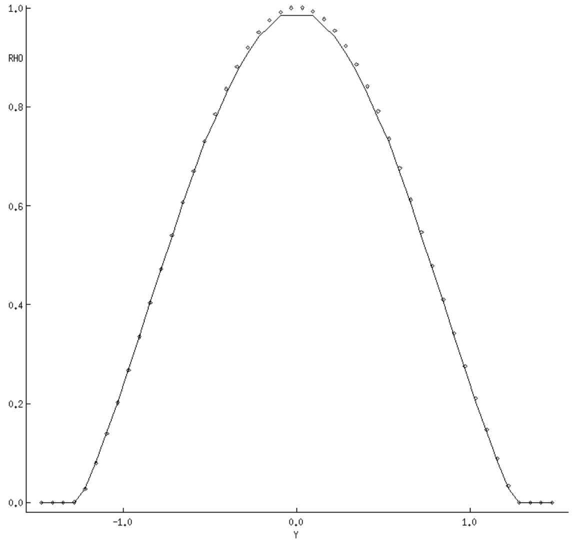
Tasker et al. (2008) found large errors in the central density for a translating King model with FLASH. ENZO PPM did better, but parallel gravity solver is dubious .

Translating $n = 1$ Polytropes

Take the same $n = 1$ polytrope and move it at speed 1 (Mach 0.71) through 5 radii.



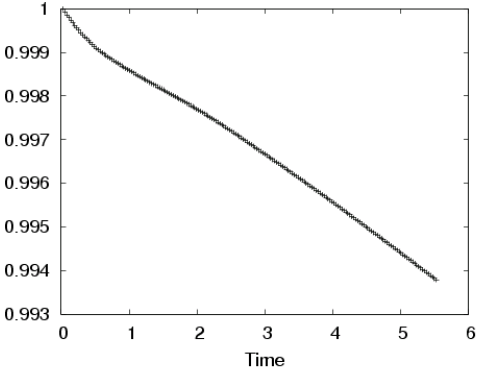
Initial Grid



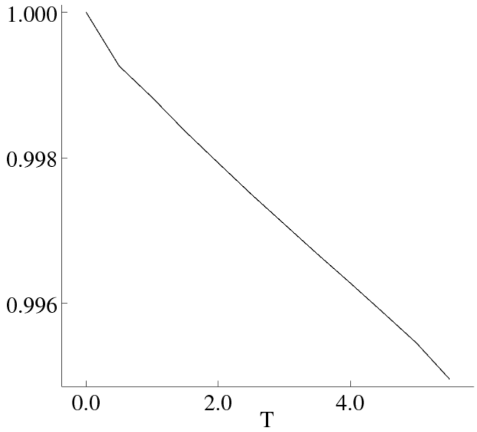
Initial and final density

Momentum decreases 0.5% during run.
RAMSES gives error of 0.63%.

Translating $n = 1$ Polytrope Momentum

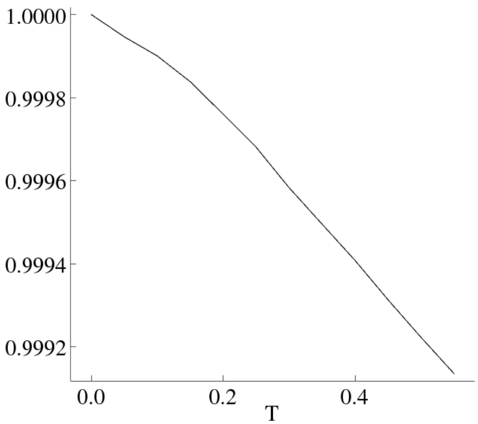


Ramses



MG

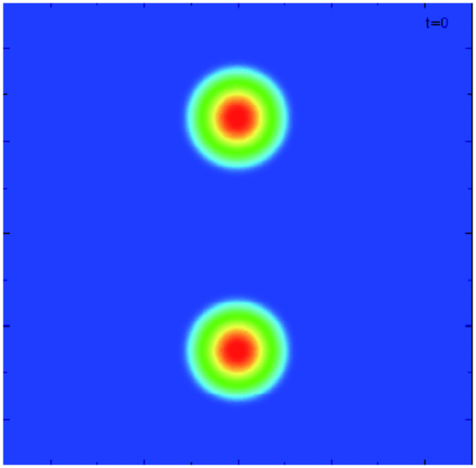
MG Velocity = 10



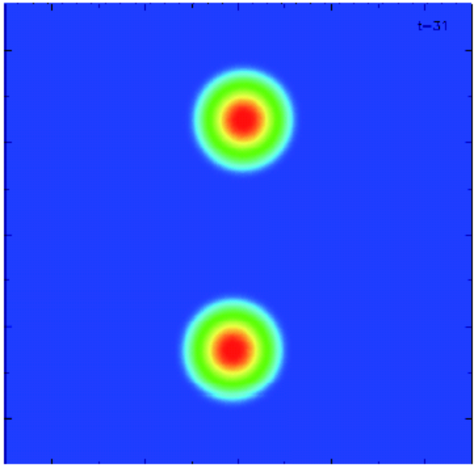
Orbiting $n = 1$ Polytropes

Take two $n = 1$ polytropes in orbit. Mach no $= 0.354$.

SPH



t = 0

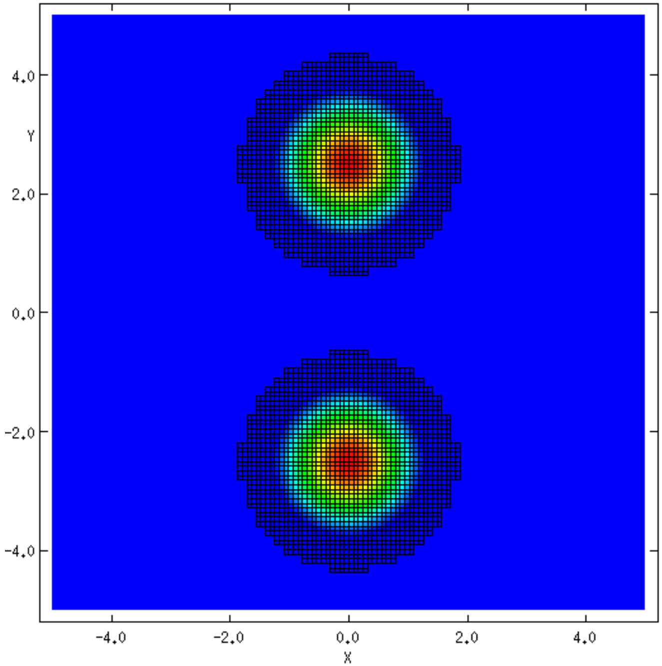


t = 30.8 (1 period)

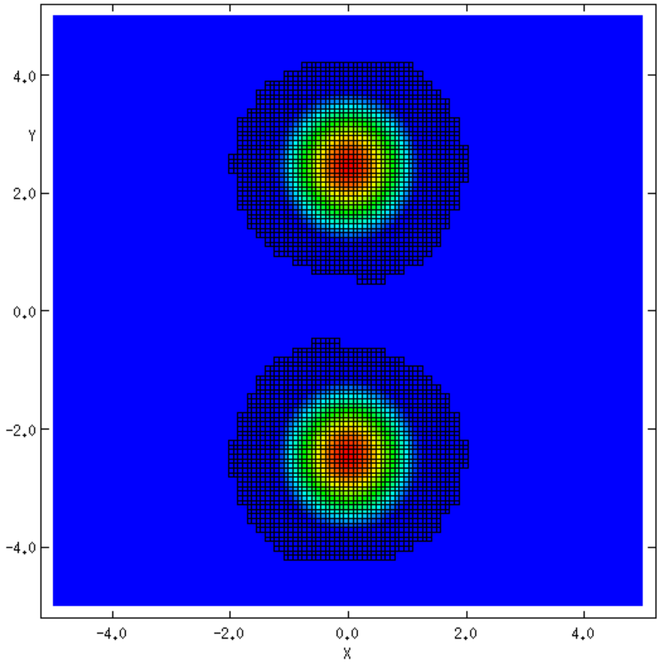
Angular momentum changes by 0.01% (depends on integration scheme).

Binary separation changes by -1$\%$ (error + tidal).

MG



$T = 0$



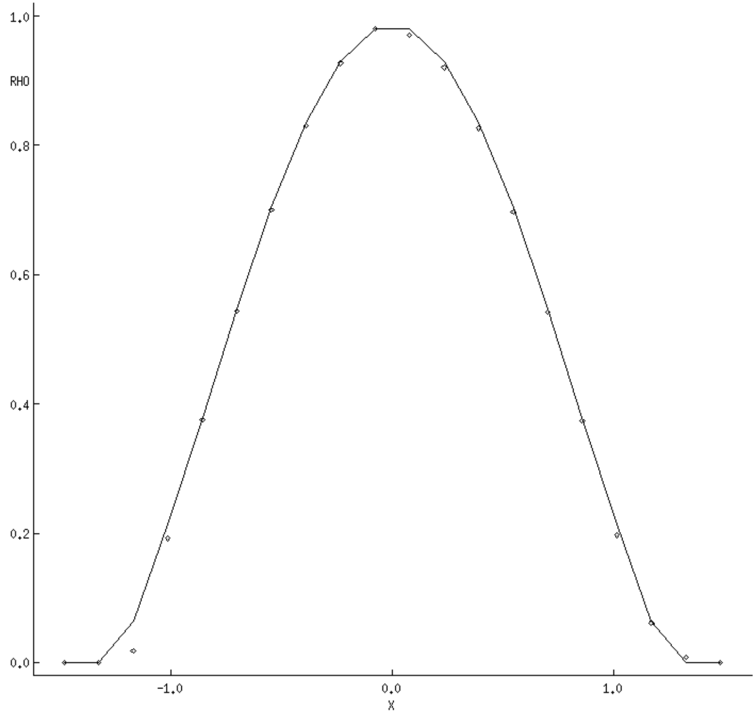
$T = 31$

Angular momentum changes by -0.26%.

Binary separation changes by -2.86% (error + tidal).

Error not due to boundary potential.

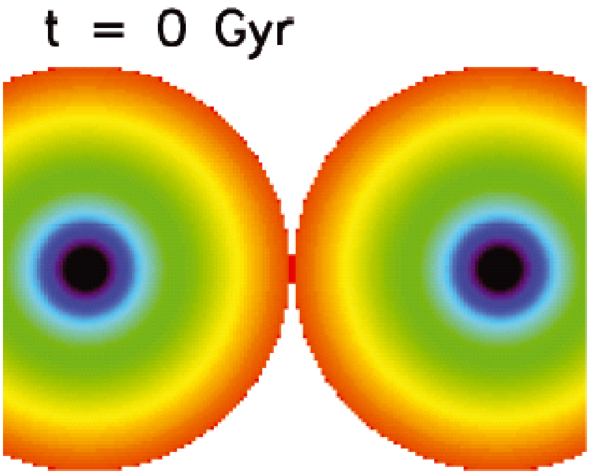
MG density profile after one orbit



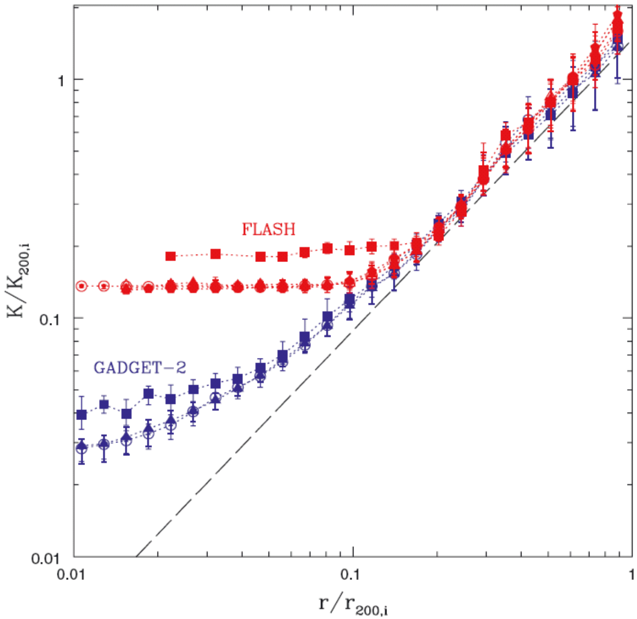
Cluster Entropy Problem

Frenk et al. 1999 found a serious discrepancy between the entropy distributions produced by grid codes and SPH during the formation of a galaxy cluster.

Mitchell et al. 2009 found the same thing with FLASH and GADGET-2 for a simpler problem of the collision of two clusters. Not much simpler: it has dark matter. Collision velocity is circular velocity at $4\times$ core radius.

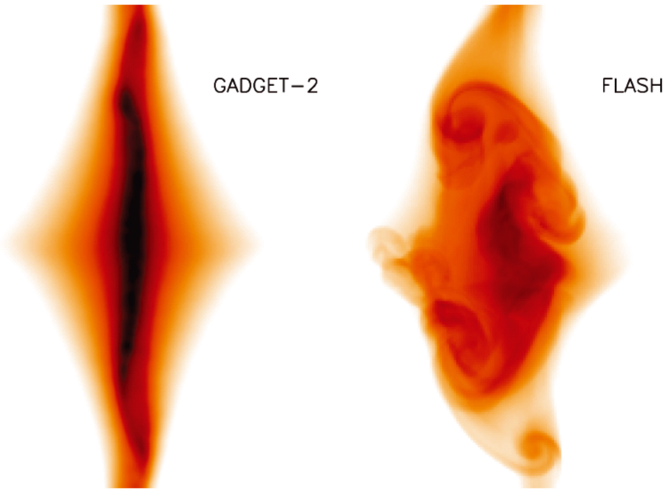


Initial Conditions (p/ρ^γ)



Final p/ρ^γ (averaged over spherical shells)

Concludes that it is due to lack of mixing in SPH (the Kelvin-Helmholtz problem).



Logarithmic projected p/ρ^{γ} at $t = 2.3$ Gyr (just after collision).

Plausible: mixing would tend to make entropy uniform.

But many other factors and FLASH did not do too well on the translating King cluster.

Really Simple Problem

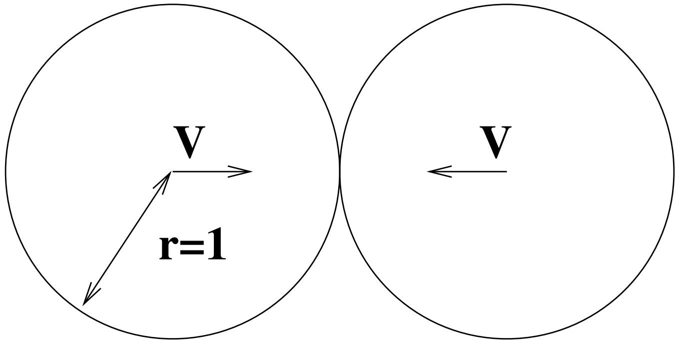
Look at collision of two clouds in a fixed spherically symmetric potential.

Uniform cloud density $= 1$, pressure 10^{-3}.

Potential due to $\rho(r) = \dfrac{0.9}{r(1+r)^2}$

Bizarre, but same as Mitchell et al.

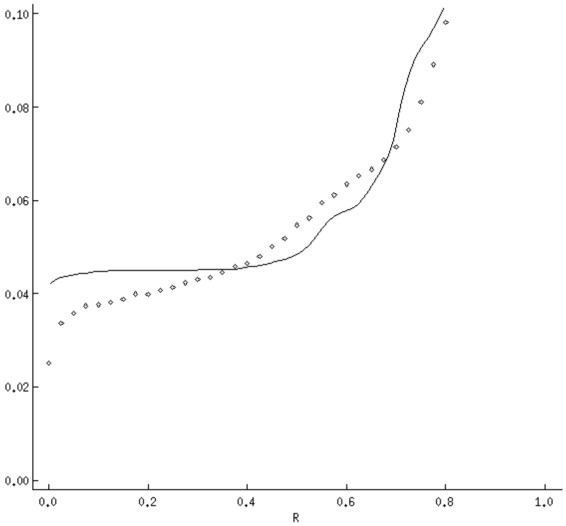
$V = 0.2 \times$ escape speed at $r = 1$.



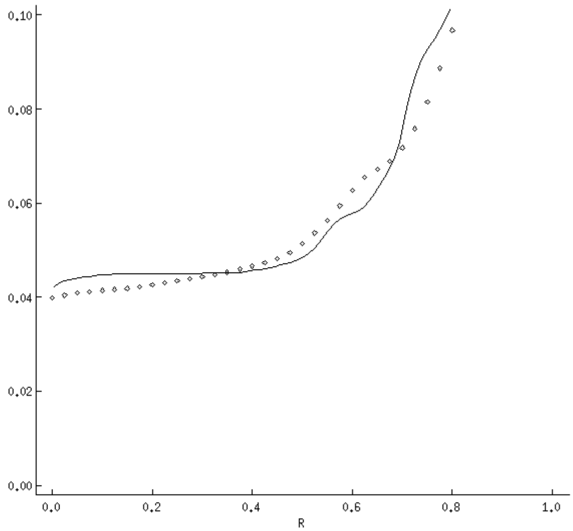
2D so that we can easily use very high resolution.

p/ρ^γ averaged over spherical shells

MG – line, SPH – markers



SPH1



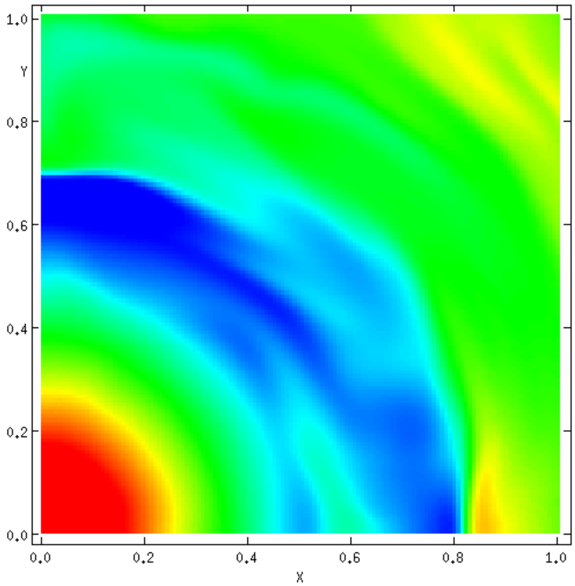
SPH2

SPH1: Standard SPH with cubic kernel.

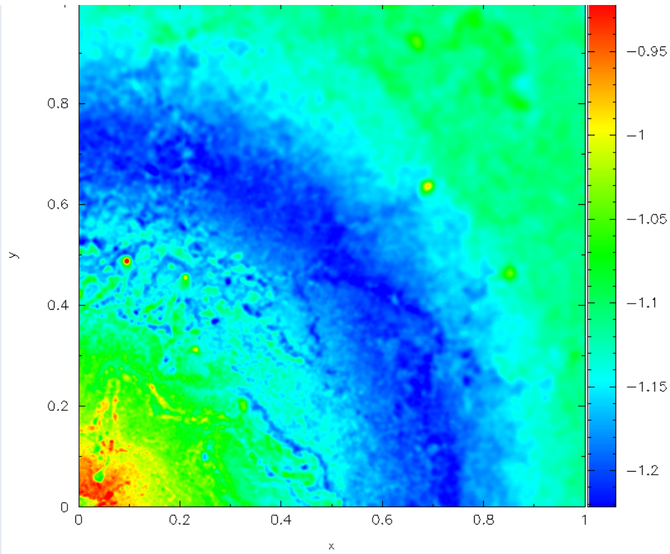
SPH2: Riemann SPH, quintic kernel, thermal conductivity.

Total entropy in $0 < r < 1$ almost identical for all three calculations $\Rightarrow$ mixing.

Temperature

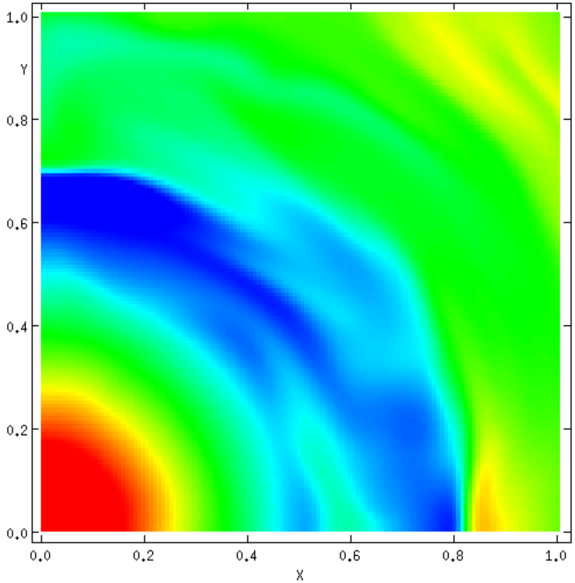


MG

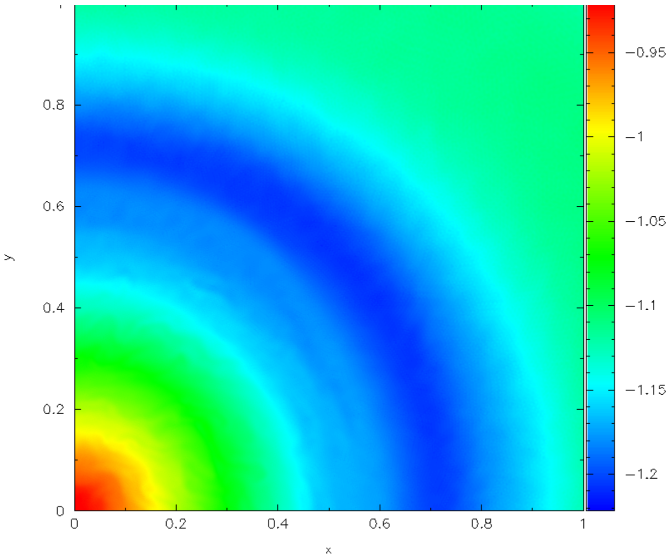


SPH1

Temperature

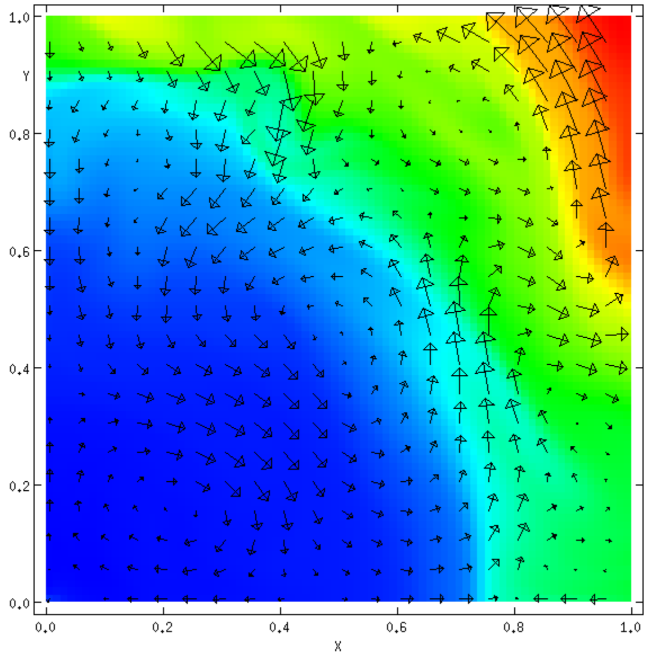


MG



SPH2

Vortices



Velocity and p/ρ^{γ} at t = 100

What's the correct answer?

Standard SPH gets this problem wrong, grid code is better for the problem as posed.

But this is not the correct answer for the cluster problem.

Actually flow is turbulent and no Euler code models this correctly.

Turbulence would make the entropy uniform if thermal conductivity is small compared to viscosity.

But the opposite is true for high temperature gas. So would expect temperature to be uniform.

Amusing twist

Codes agree much better for a sensible potential e.g. uniform sphere.