Mon. Not. R. Astron. Soc. **000**, 000–000 (0000) Printed 19 June 2014 (MN LATEX style file v2.2)

GIZMO: A New Class of Accurate, Mesh-Free Hydrodynamic Simulation Methods

Philip F. Hopkins[1]*

[1]*Department of Astronomy, University of California Berkeley, Berkeley, CA 94720*

Submitted to MNRAS, June, 2012

ABSTRACT

We present and study two new Lagrangian numerical methods for solving the equations of hydrodynamics, designed to simultaneously solve many of the known disadvantages of both smoothed-particle hydrodynamics (SPH) and grid-based or adaptive mesh refinement (AMR) schemes. The new methods are based on a kernel discretization of the volume coupled to an arbitrarily high-order matrix gradient estimator and a Riemann solver acting over the volume "overlap." We implement and test a massively parallel, second-order version of the method with coupled self-gravity & cosmological integration, in the new code GIZMO: this simultaneously maintains exact mass, energy and momentum conservation; exhibits superior angular momentum conservation compared to all other methods we study; converges with second-order accuracy (without any need to increase neighbor numbers as in SPH methods); is Galilean-invariant; is numerically stable without "artificial diffusion" terms; and allows the fluid elements to move with the flow so resolution is automatically adaptive. We consider a wide range of test problems and systematically compare these methods to moving-mesh, stationary (non-moving) grid/AMR, and various SPH "flavors." The new methods are competitive with moving-mesh schemes on all problems we consider, and have several advantages, most notably dramatically improved angular momentum conservation in gravitating systems. We can accurately following hundreds of orbits of gaseous disks, while AMR, SPH, and moving mesh methods begin break down in a couple of orbits. The new methods have many advantages vs. SPH: proper convergence and elimination of known SPH errors, good capturing of fluid-mixing instabilities, dramatically reduced "particle noise" & numerical viscosity, more accurate sub-sonic flow evolution, & sharp shock-capturing. Advantages vs. non-moving meshes include: automatic adaptivity, dramatically reduced advection errors & numerical diffusion/overmixing, Galilean invariance of numerical errors, good angular momentum conservation and elimination of "grid alignment" effects, and accurate coupling to *N*-body gravity solvers. All of these are important for a wide range of astrophysical problems, especially those involving multiphase media, complicated fluid motion, self-gravity, and rotation.

Key words: methods: numerical — hydrodynamics — instabilities — turbulence — cosmology: theory

1 INTRODUCTION: THE CHALLENGE OF EXISTING COMPUTATIONAL METHODS

Smoothed particle hydrodynamics (SPH) is a method for solving the equations of hydrodynamics (in which Lagrangian discretized mass elements are followed; **??**) which has found widespread application in astrophysical simulations and a range of other fields as well (for recent reviews, see **???**).

The popularity of SPH owes to a number of properties: compared to many other methods, it is numerically very robust (stable), trivially allows the tracing of individual fluid elements (Lagrangian), automatically produces improved resolution in high-density regions without the need for any ad-hoc pre-specified "refinement" criteria (inherently adaptive), is Galilean-invariant, couples properly and conservatively to N-body gravity schemes, exactly solves the particle continuity equation,[1] and has excellent conservation properties. The latter character stems from the fact that – unlike Eulerian grid methods – the SPH equations of motion (EOM) can be rigorously and exactly derived from a discretized particle Lagrangian, in a manner that guarantees manifest and simultaneous conservation of energy, entropy, linear momentum, and angular momentum (**?**, henceforth S02).

However, there has been considerable discussion in the literature regarding the accuracy with which the most common SPH algorithms capture certain fluid mixing processes (particularly the Kelvin-Helmholtz instability; see e.g. **??????**). Comparison between SPH and Eulerian (grid) methods shows that while agreement is quite good for super-sonic flows, strong shock problems, and regimes with external forcing (e.g. gravity); "standard" SPH appears to suppress mixing in sub-sonic, thermal pressure-dominated regimes associated with contact discontinuities (**????**).[2] The reason is, in part, that in standard SPH the kernel-smoothed density enters the EOM, and so behaves incorrectly near contact discontinuities (introducing an artificial "surface tension"-like term) where the density is not differentiable.

A variety of "flavors" (alternative formulations of the EOM or kernel estimators) of SPH have been proposed which remedy this (see above and **????????**). These approaches share an essential common principle, namely recognizing that the pressure at contact discontinuities must be single-valued (effectively removing the surface tension term). Some of these show great promise. However, many (though not all) of these formulations either introduce

* E-mail:phopkins@caltech.edu

[1] This is the continuity equation for a *discretized* particle field. Exactly solving the continuity equation for a continuous fluid, of course, requires infinite resolution or infinite ability to distort the Lagrangian particle "shape."

[2] In fairness, we should emphasize that it has long been well-known that Eulerian grid codes, on the other hand, err on the side of *over*-mixing (especially when resolution is limited), and in fact this problem actually motivated some of the SPH work discussed above. This may, however, be remedied in moving-mesh approaches (though further study is needed; see e.g. **?**).

additional (potentially unphysical) dissipation terms and/or explicitly violate the manifest conservation and continuity solutions described above – perhaps the greatest advantages of SPH. This can lead to severe errors in problems with strong shocks or high-Mach number flows, limited resolution, or much larger gradients between phase boundaries (J. Read, private communication; see also the discussion in **???**). All of these regimes are inevitable in most astrophysically interesting problems.

Recently however, **?** (henceforth SM12) pointed out that the essential results of most of these flavors can be derived self-consistently in a manner that does properly conserve energy. The key insight is that the "problematic" inclusion of the density in the EOM (as opposed to some continuous property near contact discontinuities) arises because of the ultimately arbitrary choice of how to discretize the SPH *volume* element (typically chosen to be $\sim m_i/\rho_i$). Beginning with an alterative choice of volume element, one can in fact consistently derive a conservative EOM. They propose a specific form of the volume element involving internal energy and pressure, and show that this eliminates the surface tension term and resolves many problems of mixing near contact discontinuities.

In this paper, we develop this approach to provide a rigorous, conservative, Lagrangian basis for the formulation of alternative "flavors" of SPH, and show that this can robustly resolve certain issues in mixing. Although the EOM derived in SM12 conserves energy, it was derived from an ad-hoc discretization of the hydrodynamic equations, not the discrete particle Lagrangian. As such it cannot guarantee *simultaneous* conservation of energy and entropy (as well as momentum and angular momentum). And the EOM they derive is conservative only for constant SPH smoothing lengths (in time and space); to allow for adaptive smoothing (another major motivation for SPH), it is necessary to derive the "∇h" terms which account for their variations. This links the volume elements used for smoothing in a manner that necessitates a Lagrangian derivation. And their derivation depends on explicitly evolving the particle internal energy; there are a number of advantages to adopting entropy-based formulations of the SPH equations instead.

We show here that – allowing for a different initial choice of which thermodynamic volume variable is discretized – an entire extensible class of SPH algorithms can be derived from the discrete particle Lagrangian, and write a general EOM for these methods (Eq. **??**, our key result). We derive specific "pressure-energy" (Eq. **??**) and "pressure-entropy" (Eq. **??**) formulations of the EOM, motivated by the approaches above that endeavor to enforce single-valued SPH pressures near contact discontinuities. We consider these methods in a wide range of idealized and more complex test problems, and show that they *simultaneously* maintain manifest conservation while tremendously improving the treatment of contact discontinuities and fluid mixing processes.

2 A NEW NUMERICAL METHODOLOGY

Here we outline

3 REFERENCE METHODS: COMPARISON TO OTHER STANDARDS

Other methods

Finite Pointset Method (FPM; Kuhnert 1999, 2002); Finite Point Method (Onate 1996); Finite Particle Method (Liu 2005);

4 TEST PROBLEMS

In this section, we compare results from the different methods we have discussed in a number of pure hydrodynamic test problems. We will frequently compare both of our new proposed methods, both "traditional" and "modern" SPH, as well as moving mesh and regular grid codes. For the latter, we will compare with the published results from the AREPO code for moving mesh results (**?**), and the results of the publicly available high-accuracy and high-order fixed-grid MHD code ATHENA (**?**). Many of the tests are widely used (appearing for example in both of the grid code papers above and being standard elements in the ATHENA test suite), and enable comparison to many additional codes.

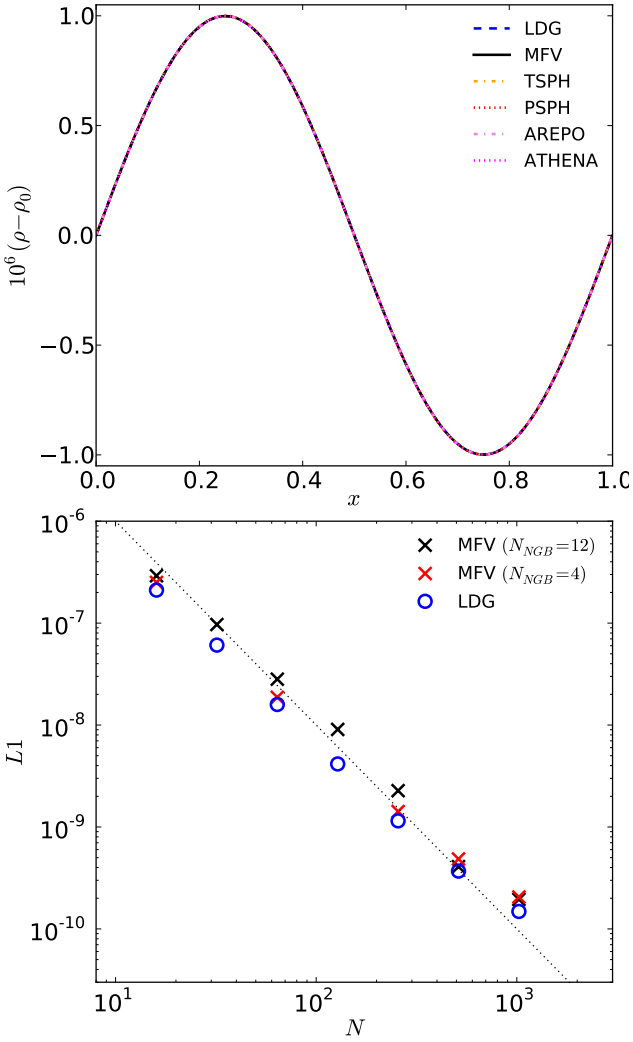

Figure 1. Linear traveling soundwave test problem (§ 4.1.1). *Top:* Soundwave evolved one period; we solve the problem with each of the methods shown, corresponding to our new lagrangian method (LDG), new meshless finite volume method (MFV), "traditional" SPH (TSPH), "modern" pressure-SPH (PSPH), a moving mesh (AREPO), and a fixed grid (ATHENA). All the codes give indistinuishable results from the analytic solution. *Bottom:* L1 error norm as a function of particle number N. Dotted line shows the ideal second-order ($L1 \propto N^{-2}$) scaling. Both new methods are second-order. The neighbor number $N_{\rm NGB}$ is fixed while N is varied; we plot results for two choices of $N_{\rm NGB}$ for the MFV method. Convergence does not require increased $N_{\rm NGB}$. We also compare

4.1 Smooth Equilibrium Tests

First we consider tests which should reflect equilibrium or steady-state configurations. Some of these turn out to be the most demanding tests of certain methods!

Table 1. Summary of Some Popular Numerical Hydrodynamics Methods

Method Name	Order	Conservative? (Mass/Energy /Momentum)	Conserves Angular Momentum	Numerical Dissipation	Long-Time Integration Stability?	Number of Neighbors	Known Difficulties
Smoothed-Particle Hydro. (SPH)							
"Traditional" SPH (GADGET, GASOLINE, TSPH)	0	✓	up to AV	artificial viscosity (AV)	✓	~ 32	fluid mixing, noise, E0 errors
"Modern" SPH (P-SPH, SPHS, PHANTOM, SPHGal)	0	✓	up to AV	AV+conduction +switches	✓	$\sim 128-442$	excess diffusion, E0 errors
"Corrected" SPH (rpSPH, Integral-SPH, Morris96 SPH, Moving-Least-Squares SPH)	0-1	×	×	artificial viscosity	×	~ 32	errors grow exponentially, "self-acceleration"
"Godunov" SPH (GSPH, GSPH-I02, Cha03 SPH)	0	✓	up to gradient errors	Riemann solver	✓	~ 300	stability, expense, E0 errors
Finite-Difference Methods							
Gridded/Lattice Finite Difference (ZEUS, PENCIL)	2-3	×	×	artificial viscosity	×	$\sim 8-128$	instability, non-conservation,
Lagrangian Finite Difference (PHURBAS, FPM)						~ 60	advection errors
Finite-Volume Godunov Methods							
Static Grids (ATHENA, PLUTO)	2-3	✓	×	Riemann solver	✓	~ 8 (geometric) $\sim 8-125$ (stencil)	over-mixing, ang. mom., non-Galilean invariance (NGI)
Adaptive-Mesh Refinement (AMR) (ENZO, RAMSES, FLASH)	2-3	✓	×	Riemann solver	✓	$\sim 8-48$ $\sim 24-216$	over-mixing, ang. mom., NGI, refinement criteria
Moving-Mesh Methods (AREPO, TESS)	2	✓	×	Riemann solver	✓	$\sim 18-30$	mesh tangling, ang. mom., refinement criteria
New Methods In This Paper							
Lagrangian Discontinuous Galerkin & Meshless Finite-Volume (LDG, MFV)	2	✓	exact in linear flows	Riemann solver	✓	~ 32	? (TBD)

A crude description of various numerical methods which are referenced throughout the text. Note that this list is necessarily incomplete, and specific sub-versions of many codes listed have been developed which do not match the exact descriptions listed. They are only meant to broadly categorize methods and outline certain basic properties.
(1) Method Name: Methods are grouped into broad categories. For each we give more specific sub-categories, with a few examples of commonly-used codes this category is intended to describe.
(2) Order: Order of consistency of the method, for smooth flows (zero means the method cannot reproduce a constant). "Corrected" SPH is first-order in the pressure force equation, but zeroth-order otherwise. Those with 2-3 listed depend on whether PPM methods are used for reconstruction (they are not 3rd order in all respects). Note that all the high-order methods become 1st-order at discontinuities.
(3) Conservative: States whether the method manifestly conserves mass, energy, and linear momentum (✓), or is only conservative up to integration accuracy (×).
(4) Angular Momentum: Describes the *local* angular momentum (AM) conservation properties. No method which is numerically stable exactly conserves local AM (even if *global* AM is conserved). Either the method has no AM conservation (×), or conserves AM up to certain errors, such as the artificial viscosity and gradient errors in SPH.
(5) Numerical Dissipation: Source of numerical dissipation in e.g. shocks. Either this comes from an up-wind Riemann solver scheme, or artificial viscosity/conductivity/hyperdiffusion terms.
(6) Integration Stability: States whether the method has long-term integration stability (i.e. do errors grow unstably).
(7) Number of Neighbors: Typical number of neighbors between which hydrodynamic interactions must be computed. For meshless methods this is the number in the kernel. For mesh methods this can be either the number of faces (geometric) when a low-order method is used or a larger number representing the stencil for higher-order methods.
(8) Known Difficulties: Short summary of some known problems/errors common to the method. An incomplete and non-representative list!

4.1.1 Linear Traveling Soundwave: Convergence Testing

We begin by considering a simple linear one-dimensional soundwave.[3] This is problem is analytically trivial; however, since virtually all schemes are first-order for discontinuities such as shocks, smooth linear problems with known analytic solutions are the only way to measure and quantitatively test the accuracy and formal convergence rate of numerical algorithms. Following **?**, we initialize a periodic domain of unit length, with a polytropic $\gamma = 5/3$ gas with unit mean density and sound speed (so pressure $P = 3/5$). We then add to this a traveling soundwave with small amplitude $\delta\rho/\rho = 10^{-6}$ (to avoid any non-linear effects) with unit wavelength. After the wave has propagated one wavelength, it should have returned exactly to its initial condition.

Fig. 1 shows the results for each code after one period. Unsurprisingly, all the methods are able to accurately follow the soundwave. After one wave propagation period, we define the L1 error norm as

$$\mathrm{L1} = \frac{1}{N} \sum_{i} |\rho_i - \rho(x_i)| \quad (1)$$

where N is the number of particles, ρ_i is the numerical solution for cell i, and $\rho(x_i)$ is the analytic solution (identical to the initial conditions for this problem). Fig 1 shows the error norm as a function of the particle number: for both the LDG and MFV methods, the results show second-order convergence (as expected for a smooth problem and a second-order accurate method). The LDG shows slightly smaller errors but the difference is not large. Note that the number of neighbors in the kernel is kept *fixed* as N is increased: convergence does not require higher-N. For all kernel-based methods, we use $N_{\rm NGB} = 4$ neighbors in one dimension unless otherwise specified, for this and all other 1D tests; however for the LDG and MFV methods the results are insensitive to this choice as long as it is not too small, $<= 2$). We show this explicitly by comparing the L1 norm for $N_{\rm NGB} = 12$ for the MFV method (the LDG result is similar). If anything, the L1 norm is slightly *larger* for high $N_{\rm NGB}$; this is because of the additional numerical diffusion from solving the Reimann problem across more neighbors. The L1 norm becomes slightly larger again for $N_{\rm NGB} < 4$, because there are not enough particles in the stencil, but in all cases it exhibits second-order convergence.

We have evolved the wave to ~ 1000 periods in the LDG and MFV methods, and see no visible diffusion at the resolution plotted (as expected).

Repeating this test in 2D and 3D gives similar results for all codes.

4.1.2 The Square Test: Advection & Surface Tension Errors

We next consider the "square" test common for recent SPH studies (**????**). We initialize a two-dimensional fluid in a periodic box of length $L = 1$ and uniform pressure $P = 2.5$, $\gamma = 1.4$, and density $\rho = 4$ within a central square of side-length $L = 1/2$ and $\rho = 1$ outside. The entire fluid is given a uniform and abitrary large initial velocity $v_x = 142.3$, $v_y = -31.4$. We use 64^2 particles. Fig. 2 shows the initial condition and the resulting system evolved to a time $t = 10$, centered on the central square position at that time. The square should be in hydrostatic equilibrium, with the final state exactly identical to the initial state.

The LDG and MFV methods perform essentially perfectly here: in fact, it is straightforward to show that they solve this particular test problem exactly (to machine accuracy). The same is true of moving mesh codes, provided that the moving mesh also uses a gradient estimator which is exact for linear gradients and advects cells with the bulk fluid velocity.

It is well known (see the references above) that "traditional" SPH (all SPH methods where the density is kernel-smoothed but entropy or internal energy is particle-carried) have an error term which behaves as a physical surface tension: a repulsive force appears on either side of the contact discontinuity, opening the gap between the central square and outer medium which then deforms the square to minimize the surface area of the contact discontinuity (eventually becoming a circle). This is the same as the error which generates the "pressure blip" in the shocktube test below. We see exactly this behavior here. Perhaps most disturbing, the error *does not converge away* (it is zeroth-order). The pressure-entropy case minimizes this error (see **??**); however, there is still a "rounding" of the corners and substantial noise around the edges of the square. This owes to two factors: (1) the zeroth-order consistency (E0) error in SPH means that even when every particle has an exactly identical pressure, there are still net forces on the particles, especially when there is an asymmetry in the particle distribution as occurs near the contact discontinuity; (2) the artificial conduction terms in the modern SPH diffuse the contact discontinuity even when there is perfect stability.

If the square is not moving, this problem is trivial for grid codes. However, if the square has any motion relative to the grid (and not perfectly aligned), then large advection errors appear. In ATHENA, each time the square moves its own length, it is both diffused and distorted (the magnitude of the distortion comparable to that in SPH "per crossing"). Here we have used the second-order integration method to match the other codes; if we use a higher order PPM method we see some improvement but the qualitative behavior is the same, if we use a first-order method the square cannot be reliably advected even a single unit length. It is also worth noting that in the grid code, the Courant criterion must include the relative gas-grid motion: thus these errors appear despite the fact that the timesteps in the grid code are a factor of ~ 1000 smaller than in all the other methods. And we stress that AMR methods cannot help here, without overall increasing the resolution (in which case they will still be less accurate than an LDG or MFV run at the same resolution), since the diffusion is uniform around the boundaries – in fact running this test with RAMSES or ENZO, we actually see *more* diffusion if we refine at the contact discontinuity, because (as is well-known) the AMR scheme effectively becomes lower-order along refinement boundaries.

Note that in the 1D analog of this problem (advecting a constant-pressure, constant-velocity 1D contact discontinuity), LDG, MFV, and moving-mesh methods perform similarly well, and SPH has no problems (a pressure 'blip' is present, but the surface tension-like instability only appears in higher dimensions). But it is well-known that non-moving grid codes will still excessively diffuse the discontinuity (even though the motion is necessarily grid-aligned; see **?**). In the 3D analog (advecting a cube), the results and differences between codes are essentially identical to the 2D test here (SPH deforms it into a sphere, fixed-grid codes diffuse along all edges, moving-mesh, LDG, and MFV codes are machine-accurate).

[3] See `http://www.astro.princeton.edu/~jstone/Athena/tests/linear-waves/linear-waves.html`

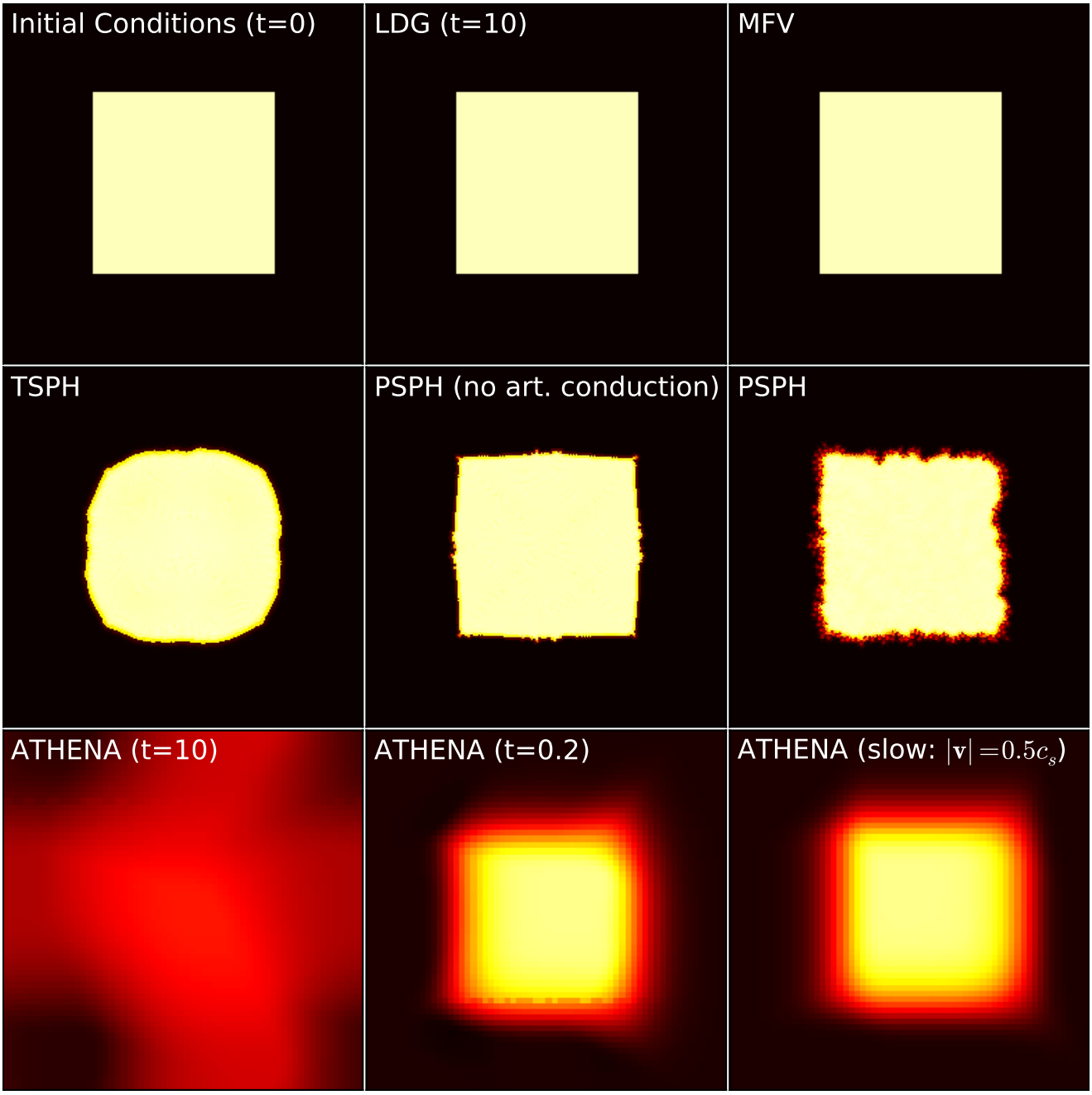

Figure 2. Hydrostatic square advection test (§ 4.1.2). *Top Left:* Initial condition (red shows density $\rho = 4$, blue $\rho = 1$), a high-density square in hydrostatic equilibrium, with all fluid moving at constant velocity. *Top Center:* LDG solution at $t = 10$; this reproduces the correct solution (identical to the IC) to machine precision. *Top Right:* MFV solution. This is also exact to machine precision. AREPO should do the same. *Middle Left:* TSPH: Advection is handled well, but the known "surface tension" error forces the square gradually into a circle. *Middle Center:* PSPH, using the "traditional" SPH artificial viscosity and no artificial conductivity: this removes the surface tension but particle asymmetry around the contact discontinuity still produces spurious forces. *Middle Right:* "standard" PSPH: artificial conductivity produces excessive (and noisy) diffusion around the discontinuity. *Bottom Left:* ATHENA: advection errors completely destroy the square, despite the Courant condition forcing ~ 1000 times smaller timesteps in this case. *Bottom Center:* ATHENA result at time $t = 0.2$, showing the magnitude of distortion after the square moves a few times its size. *Bottom Right:* ATHENA result for a slower sub-sonic (Mach number $= 0.5$) advection at $t = 10$ (the square has traveled much less distance).

4.1.3 The Gresho Vortex: Sub-Sonic Turbulence & Angular Momentum

We next consider the triangular vortex of **?**. A two-dimensional gas with uniform $\rho = 1$ is initialized in a periodic domain $0 < x < 1$, $0 < y < 1$, with zero radial velocity, pressure

$$P(R) = \begin{cases} 5 + 12.5\,R^2 & (0 \le R < 0.2) \\ 9 + 12.5\,R^2 - 20\,R + 4\ln(5\,R) & (0.2 \le R < 0.4) \\ 3 + 4\ln 2 & (R \ge 0.4) \end{cases} \quad (2)$$

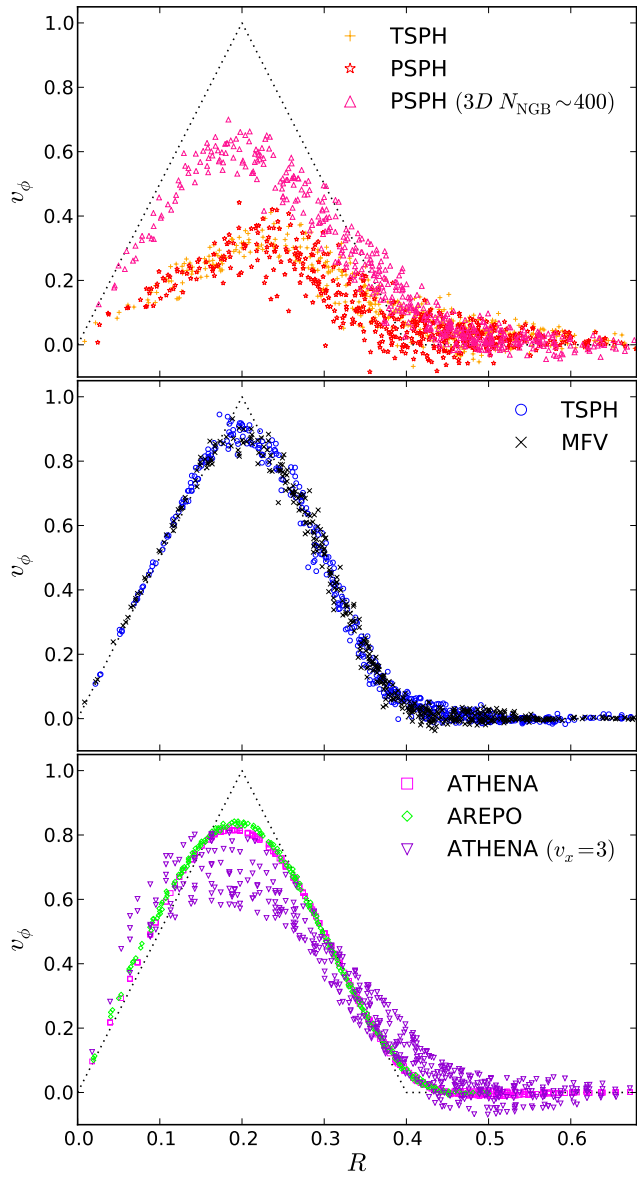

Figure 3. The Gresho vortex (§ 4.1.3). The code should preserve a steady-state hydrodynamic vortex following the analytic solution (dotted line); we plot the azimuthal velocity versus radius for each code method at $t=3$, or ~ 1 complete vortex orbit, at 40^2 resolution (each point is one particle/cell; for clarity we plot a random subset of all cells). *Top:* SPH methods. This is known to be a very challenging test for SPH, and even the most sophisticated SPH methods generate large noise (from "remeshing" and the E0 error) and steadily degrade the vortex. Increasing the kernel neighbor number helps, but convergence is slow: we compare a test run with a higher-order Wendland kernel and the 3D equivalent of 400 neighbor particles (vs. standard 32). *Middle:* LDG and MFV methods. The two are very similar. Some (much smaller) noise is generated but the peak is preserved. *Bottom:* Moving-mesh and fixed-grid methods. These give very similar results when the vortex has zero mean velocity; both give much less noise (because the volume partition is exact, not second-order), though they dissipate the peak slightly more than LDG/MFV. We compare, however, the results if the vortex is moving (add a uniform velocity $v_x=3$); here advection errors lead to much larger noise in the fixed-grid solution (while the moving-mesh, LDG, MFV, and SPH results are machine-accuracy invariant to such boosts).

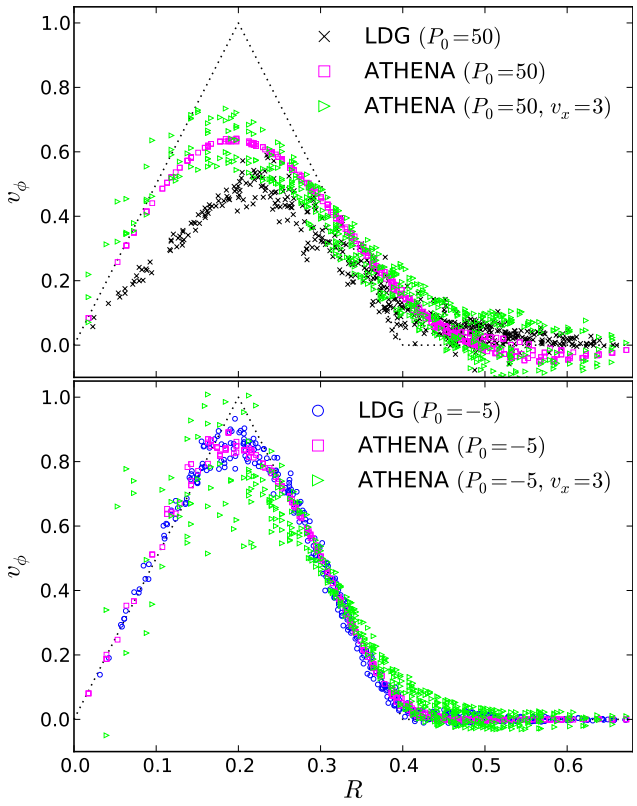

Figure 4. Gresho vortex as Fig. 3, but varying the Mach number of the vortex. We compare the LDG and fixed-grid methods at $t=3$ (with and without a moving vortex, for the fixed-grid results). MFV is very similar to LDG in all cases so is not shown. *Top:* A highly sub-sonic (rms Mach number ~ 0.06) vortex (background pressure increased by $P_0=50$). The accuracy of all methods degrades, but the effect is more severe in the LDG/MFV methods. *Bottom:* A trans-sonic (rms Mach ~ 0.6; $P_0=-5$) vortex. All solutions improve, except the noise in grid methods (especially for a moving vortex) gets larger. In all cases, both SPH methods (even with $N_{\rm NGB}=400$) perform significantly worse than any other method.

and azimuthal velocity

$$v_\phi(R) = \begin{cases} 5R & (0 \le R < 0.2) \\ 2-5R & (0.2 \le R < 0.4) \\ 0 & (R \ge 0.4) \end{cases} \quad (3)$$

where $R^2=x^2+y^2$. This represents a steady-state equilibrium vortex. We initialize the vortex with 64^2 elements. Fig. 3 shows the results, evolved to time $t=3$, or about 2.4 orbits of the vortex "peak" (1.2 orbits of the outermost vortex edge). There is no "1D vortex" analogue of this problem; but we discuss the 3D analogue (the "vortex tube") below.

It is well-known that SPH has serious difficulties with this test (in fact, in most SPH tests in the literature, the vortex is not evolved beyond $t=1$). The shear motion of particles leads to a constant re-mapping of the effective particle "volume"; since particle masses and energies are locally carried this leads to noise in the pressure field, hence ultimately in the velocity field. The velocity noise is damped by artificial viscosity, diffusing the vortex. We confirm this: with both TSPH and PSPH, the results are very noisy, and the damping of the peak velocity is severe, as is the velocity diffusion out to larger radii (beyond the original vortex). Improved artificial viscosity switches do not do much to change this. Various authors have pointed out that increasing the kernel neighbor num-

ber does help here; **?** (see Fig. 5-6 there) and **?** (Fig. 9-10 therein) advocate going to $N_{\rm NGB} > 400$ neighbors in 3D. We have in fact repeated this test using the Wendland C^6 kernel with $N_{\rm NGB} = 500$ or triangular kernel with $N_{\rm NGB} = 442$ (to do so we repeated this with the 3D analogue of the test, which also helps to reduce noise). This does help, but not very much; as shown by both groups, the $L1$ norm decreases only as $\sim N_{\rm NGB}^{-0.5}$. In fact, the performance with both SPH methods even with $N_{\rm NGB} \sim 500$ is still significantly worse than any other method we consider. And the computational expense involved is large: depending on the kernel, the "effective resolution" goes down as something like $N_{\rm NGB}^{-1/2}$, so the CPU cost of the SPH computation for equivalent resolution scales something like $\sim N_{\rm NGB}^{3/2}$ – i.e. this improvement entails a factor > 50 CPU cost in the SPH loops over the standard ~ 32 neighbors! And both these authors verify that, because this problem is significantly affected by the E0 error in SPH, convergence with total particle number, even at high $N_{\rm NGB}$, is slow ($\sim N_{1D}^{-0.6}$).

The LDG and MFV methods show a tremendous improvement relative to SPH, despite using the simple cubic spline kernel with fixed $N_{\rm NGB} = 32$ in 3D ($N_{\rm NGB} = 16$ in 2D, following **?**). The solution is less noisy than the SPH equivalent with $N_{\rm NGB} \gtrsim 500$, and the vortex has decayed much less rapidly. However there is still significant noise. We also find (not surprisingly) that the degree of vortex decay is very sensitive to our choice of slope-limiter: using a more conservative limit on monotonicity (see § **??**) leads to a smoother solution but stronger damping of the vortex peak.

By comparison, the results from ATHENA show almost no noise, because of the exact volume partition meaning there is no "remapping" error. There is more decay at this time compared to the LDG and MFV methods, but we find at later times the vortex is better preserved. However, while fixed-grid methods do very well in the basic version of this test, at least two simple modifications of the problem dramatically reduce their accuracy (while having no effect on the other methods we consider). The first is if we consider a 3D version of the problem, where the vortex is initialized as a cylinder with the same dependence of v and P on R and infinite (periodic) in z, then rotate the problem geometry so it is not exactly aligned with the cartesian grid axes. This creates significant errors which quickly damp the angular momentum until the vortex is realigned with the local grid (then, this realignment will slightly offset the vortices at different heights in the cylinder, which will interfere with each other via numerical viscosity until the structure is dissipated). We will consider such errors in the next test problem (§ **??**). The second modification is to simply set the problem in bulk motion. **?** consider this case in more detail, and show that the errors at fixed resolution then grow rapidly with the bulk motion: for a bulk motion with $v_{\rm bulk} \gtrsim v_{\rm vortex}$ (where $v_{\rm vortex} = 1$ is the peak vortex velocity here), the the noise in the fixed-grid solution becomes substantially larger than any of the other methods we consider; once $v_{\rm bulk}$ reaches $\gtrsim 3\,v_{\rm vortex}$, the noise becomes comparable to the actual solution. **show such a test ???**. We verify this here – for modest bulk flow velocities relative to the grid (any velocity comparable to the vortex rotation velocity itself), the noise "blows up," becoming worse than our LDG and MFV results (though still superior to the SPH results, until we reach vortex velocities $\gtrsim 30-50$).[4]

Of course, in stationary grid codes this noise owing to misalignment or bulk-motion of vortices can be reduced by increasing the resolution, and will eventually converge away. However this means that at fixed resolution, their accuracy can be severely reduced, or equivalently that their "effective resolution" will be much lower for certain problems. By comparison, all the other methods we consider are manifestly invariant to both rotations of the vortex and bulk motions. So, for a mis-aligned vortex with bulk motion of $\sim v_{\rm vortex}$, for example, we require a resolution of $\sim 256^2$ to achieve similar accuracy to a 64^2 simulation with the LDG or MFV methods. And since the whole volume is affected, AMR does not improve things.

This is a serious concern for realistic simulations with stationary grids, where the vortex position and motion cannot be exactly known "ahead of time." Consider, for example, simulations of super-sonic turbulence. If we desire to resolve a modest Reynolds number of ~ 100, then since the super-sonic cascade $|\mathbf{v}^2(\lambda)|^{1/2} \propto \lambda^{1/2}$ (where λ is a parameter reflecting spatial scale; see **?**), we expect the smallest "resolved" eddies to be randomly advecting through the box with bulk motions set by the largest eddies, a factor $100^{3/8} \sim 6$ larger than their internal eddy velocities. If we "boost" the Gresho problem by this multiplier, we find we require a resolution $\sim 32^2 - 64^2$ across the eddy for its structure to survive to $t = 3$, and $\sim 256^2$ for it to have comparable accuracy to a non-boosted 32^2 simulation: so our grid-scale should actually be $\sim 32 - 256$ (depending on the desired accuracy) times smaller in linear size than the smallest eddy we wish to resolve! This is more demanding than what is usually estimated based on examining the shape of the turbulent power spectrum, by a factor of a few (which should actually not be surprising, since here we are not just requiring the second moments be reasonable, which can be accomplished via noise, but that the eddy structure is reasonable). Because the errors grow with boost velocity, the resolution requirement grows super-linearly with the desired Reynolds number in stationary-grid simulations.

The best compromise in this particular test problem appears to come from moving mesh methods. These give similarly accurate and smooth results to the second-order stationary grid methods with no bulk velocity, but are invariant to bulk motions of the vortex and to rotations. The advantage over the MFV and LDG methods here is the exact volume partition, which drastically reduces (although does not completely eliminate) the "remapping noise."

All of these results, however, are sensitive to the Mach number of the vortex. Note that, mathematically, we can add any constant P_0 to the pressure everywhere in Eq. 2 and the dynamics should be identical. Numerically, however, none of the methods is invariant with respect to this choice. In *all* cases, lowering (raising) the background pressure ($P_0 < 0 / P_0 > 0$) leads to better (worse) conservation of the vortex; this is because small integration errors in the pressure gradients will launch spurious velocities that have magnitudes which scale with the sound speed. The minimum physical pressure for this problem, $P_0 = -5$, corresponds to a vortex with Mach number $\mathcal{M}(R = 0.2) \approx 1.1$ at the vortex "peak" (rms $\langle \mathcal{M}^2 \rangle^{1/2} \approx 0.6$ over the vortex). The standard choice of **?** above ($P_0 = 0$) corresponds to $\mathcal{M}(R = 0.2) \approx 1/3$ ($\langle \mathcal{M}^2 \rangle^{1/2} \approx 0.2$). We also consider a much higher $P_0 = 50$, or $\mathcal{M}(R = 0.2) \approx 0.1$ ($\langle \mathcal{M}^2 \rangle^{1/2} \approx 0.06$).

In all cases, the qualitative differences between the methods are similar. With $P_0 = -5$, there is some improvement across the board, but the results are similar to those shown for the "standard" choice in all cases (some surprising noise appears in the ATHENA solution, whose origin is unclear; but it is not enough to change our conclusions). However, the meshless methods (SPH, LDG, and

[4] We have verified that what matters for these errors in stationary-grid codes is the ratio of bulk velocity to vortex velocity, not the sounds speed or pressure. The errors which scale with the sound speed (discussed below) are almost entirely separable.

MFV) are much more sensitive to large P_0 than the stationary grid methods. It appears that the errors from the implicit "remeshing" grow super-linearly with sound speed. As a result, in SPH the vortex is completely "wiped out" by $t=3$ for $\langle\mathcal{M}^2\rangle^{1/2}\gtrsim 0.2-0.3$; for LDG and MFV methods we see some vortex survive to $t=3$ down to $\langle\mathcal{M}^2\rangle^{1/2}\gtrsim 0.03-0.05$; but with stationary grids we can reach $\langle\mathcal{M}^2\rangle^{1/2}\gtrsim 0.001$ (perhaps even lower with a higher-order PPM method). As above, we can always improve this with increasing resolution, but for SPH the convergence is very slow (sub-linear), and even for the LDG and MFV the convergence is closer to linear than second-order (also seen for AREPO; see **?**, Fig. 29). This is a serious concern for simulations of sub-sonic turbulence. The limitations of SPH in this regime are well-known (see e.g. **?**); we confirm those results here. But while our LDG and MFV methods offer a tremendous improvement relative to SPH, and can converge, this test suggests they lose accuracy rapidly relative to stationary grid codes once the Mach numbers fall to ~ 0.01 (numerical noise starts to swamp "real" turbulent effects at reasonable resolution of the smaller eddies in a realistic simulation). For highly sub-sonic problems, then, the lack of a re-meshing error suggests stationary grid codes offer a significant advantage. It remains to be seen how moving-mesh codes compare in this limit, since there still is a non-zero remeshing error (which depends on the "mesh normalization" procedure), but the volume partition is exact. Unfortunately we do not have access to runs with AREPO in this regime, so this remains to be studied in future work.

4.1.4 Keplerian Disks: Angular Momentum Conservation, "Grid Alignment," & Stability of Cold Orbits

We now consider a cold Keplerian disk test problem. This is a critical problem for understanding the ability of codes to conserve angular momentum and follow gas in gravitational orbits for any length of time. Disks, in particular, are ubiquitous in astrophysics, and unlike the vortex problem above are dominated by gravitational acceleration rather than pressure forces (with the rotation velocity often super or trans-sonic); this focuses on that regime.

We initialize a razor thin $\gamma=5/3$ disk with constant surface density $\Sigma=dM/dA=1$, vanishingly small pressure, and all gas on circular orbits about the origin subject to a Keplerian external acceleration $\ddot{\mathbf{r}}=-\hat{r}GM_0/|\mathbf{r}|^2$ with $G=M_0=1$ (and no self-gravity) and $\mathbf{r}$ the vector from the origin. We focus on gas around the radius $|\mathbf{r}|\approx 1$. [5] We use an "effective" resolution of 256^2 (i.e. for the particle methods the initial particles are evenly spaced such that the square domain from $-2<x<2$, $-2<y<2$ has 256^2 particles; then particles are removed inside $r<0.5$ and outside $r>2$; for the grid methods this is the same as a 320^2 grid in 2D across the $-2.5<x<2.5$ domain, or 320^3 in 3D).

Note that for the non-mesh (particle-based) codes, the fully three dimensional version of this problem is manifestly identical to a 2D problem where we initialize the gas in the $x-y$ plane with $\rho=1$. In other words the code and solution are invariant to rotations of the disk in any direction. However for any structured-grid code (fixed grid codes like ATHENA but also AMR codes, and even irregular-grid but non-moving or gridded-particle codes like PHURBAS), there is a difference if the disk is not moving exactly in the same plane as the grid cells (i.e. if we rotate the disk out of the $x-y$ plane so it is not perfectly grid-aligned). So we show the result for both cases.

Here, our LDG and MFV methods perform exceptionally well. Noise arises in the particle density and pressure distribution, as in the Gresho problem, but it has very weak effects on the dynamics. Total angular momentum and local orbits are well-conserved at the ~ 10 orbits we have followed.[6] Of all the methods we study, these appear to exhibit the lowest numerical viscosity in this specific problem (not necessarily in all problems!).

The MFV method generates some very small angular momentum errors, because there is a non-zero mass flux between particles; whenever this is not aligned with the line connecting the particle centers-of-mass, there can be weak violations. These begin to affect the disk evolution at $\sim 30-50$ orbits; hence we see in Fig. 5 that the very inner edge of the disk has experienced some angular momentum evolution. The LDG method has no mass flux, hence identically zero advection errors in angular momentum; the only way it can dissipate angular momentum is via numerical viscosity. The combination of the Reimann solver and accurate gradient estimator make this very low. Hence the angular momentum evolution is nearly perfect. In Fig. 6, we show the evolution to time $t=600$, or ~ 160 orbits of the inner disk, and see the angular momentum conservation is still nearly perfect! In fact, we have integrated as far as ~ 1000 orbits, and found that we cannot tell the difference in angular momentum conservation between our LDG method, applied to gas in this test problem, and integration of pure collisionless test particles!

On the other hand, we see a rapid and catastrophic breakup of the disk within ~ 2 orbits in our TSPH test. This is a well-known result (**???**), and occurs because of the physical viscous instability (**?**), except that the disk is supposed to be inviscid! The problem is the "standard" SPH artificial viscosity produces far too much shear viscosity.[7]

Our PSPH method uses an improved artificial viscosity switch proposed by **?**; this uses a least-squares matrix-base gradient esti-

[5] Since the problem is self-similar (we can pick any radii over which to examine the gas), we initialize gas only from radii $0.5<|\mathbf{r}|<2$, and solve the problem in an open box of side-length $L=5$ ($-2.5<x<2.5$) centered on the origin. For convenience to prevent very slow timesteps if gas reaches the center (which it should not), we actually soften the acceleration according to $\ddot{\mathbf{r}}=-\hat{r}GM_0/(|\mathbf{r}|^2+(0.25)^2)$. For particle-based codes we can simply have a hard "edge" to the initial gas disk with vacuum outside; for grid codes we must initialize a finite density, so we initialize a smooth density gradient such that the density drops off within $\Delta r\approx 0.1$ of the "edge." The details of this are unimportant to our conclusions, however.

[6] As shown by **??**, although the exact angular momentum conservation properties of the LDG and MFV methods are unclear in the general case, they do exactly conserve angular momentum if the gas distribution is first-order (i.e. there are no second-order terms in the expansion of gas properties). Higher-order terms are generated by the noise here, and by the numerical viscosity of the method, but these do not grow rapidly. In practice, we find that the errors for SPH and fixed-grid codes are dominated by a combination of numerical viscosity and advection errors – not the formal angular momentum conservation of the code. The LDG and MFV methods do the best job of *simultaneously* minimizing these errors, hence their good behavior in this test.

[7] Specifically, our TSPH example uses the "standard" **?** artificial viscosity with a **?** switch. This attempts to suppress numerical shear viscosities, but only does so by a modest amount, and is very noisy because it is based on the "standard" SPH gradient estimator that has large zeroth and first-order errors (especially in shear flows). **?** study several variations of SPH in this problem (there a Keplerian ring; see their Fig. 8), and confirm that the both the "standard" SPH artificial viscosity and the time-dependent viscosity method of **?**, with or without the **?** switch (e.g. methods in PHANTOM, GASOLINE, GADGET-2, and many other codes), undergo catastrophic fragmentation within $\lesssim 2-3$ orbits.

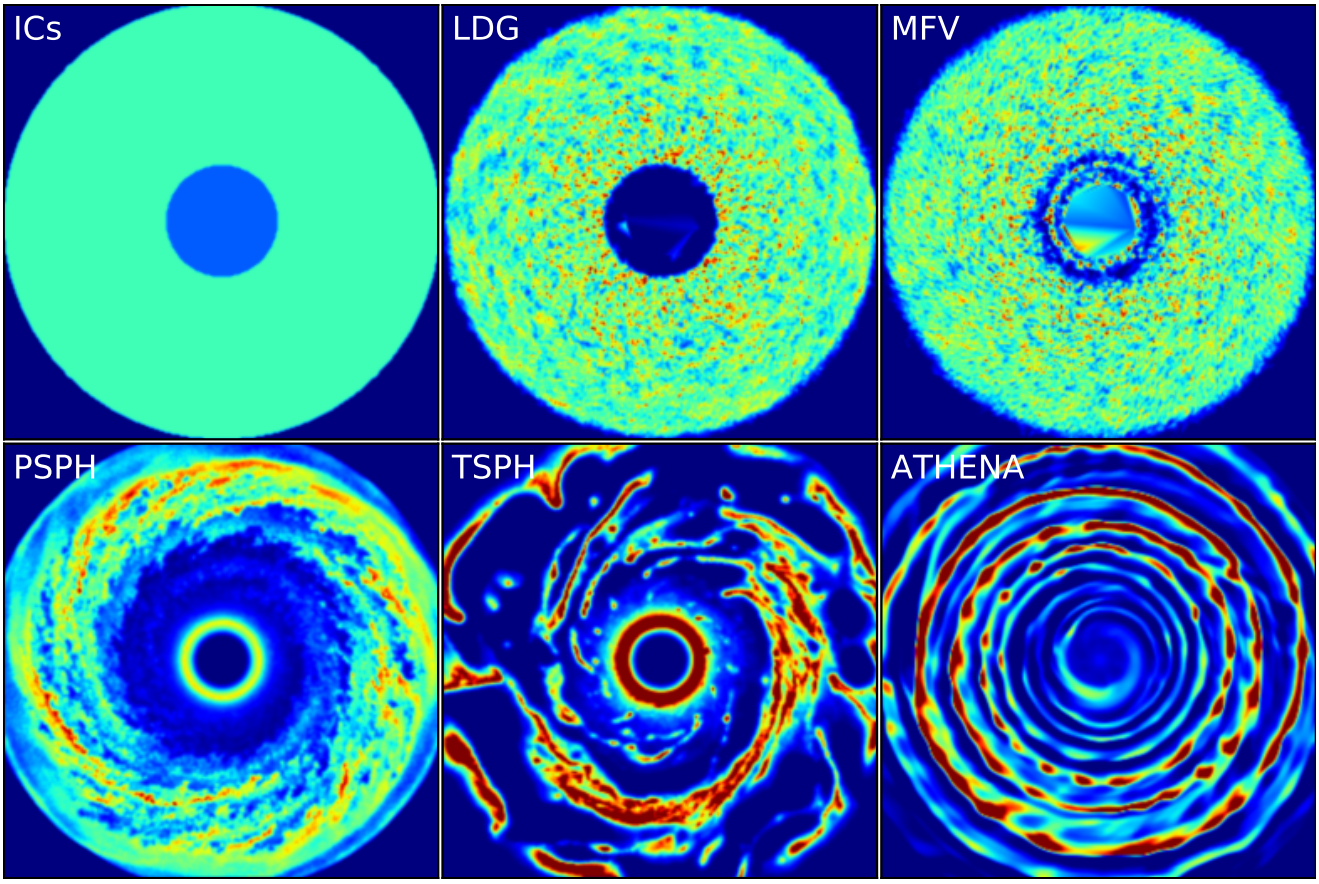

Figure 5. Keplerian disk problem (§ **??**). *Top Left:* Initial Conditions. The gas is initialized with constant surface density from $r = 0.5 - 2.0$, on circular orbits, with vanishing pressure, subject to an analytic Keplerian potential (without self-gravity), with effective 256^3 resolution. This should remain in equilibrium indefinitely, but numerical viscosity and advection errors steadily degrade the disk and transport angular momentum. We show the surface density evolved in each method to $t = 120$ (~ 10 orbits). *Top Middle:* LDG: the disk preservation is excellent (there is a small amount of noise in the density field, as in the Gresho test, but this does not degrade the orbits). We can continue to evolve the system for ??? orbits before the disk degrades. *Top Right:* MFV: mass fluxes lead to slightly less noise in the disk density, but a small amount of angular momentum transfer which begins to degrade the inner disk at $\gtrsim 30$ orbits. *Bottom Left:* PSPH: Using a high-order artificial viscosity switch, shear viscosities are sufficiently suppressed to allow good evolution to ~ 5 orbits, but the degredation is significant. *Bottom Center:* TSPH: Using "traditional" SPH artificial viscosity with a **?** switch leads to far too much shear viscosity, and the disk undergoes the viscous instability and disrupts within $\sim 1 - 2$ orbits. *Bottom Right:* Stationary meshes: numerical viscosity is low but advection errors of circular orbits through a Cartesian mesh are significant and disrupt the disk in ~ 1 orbit.

mator (similar to our LDG and MFV methods), which is zeroth-order accurate. This dramatically improves the results, allowing semi-stable evolution to ~ 5 orbits; however, we still see the viscous instability appear. The artificial viscosities are still excessively large in shear flows, and the method still has zeroth and first-order errors in the hydrodynamic forces together with first-order errors in the velocity gradient estimator.[8]

Moreover, as noted in **?**, all SPH artificial viscosity methods also produce excessively high numerical viscosities and disk breakup if the disk has modest internal turbulence (enough to set a scale height $h/R \gtrsim 0.1$), because then the artificial viscosity is "triggered" in the turbulent compressions, but cannot be "removed" instantly.[9] Once any artificial viscosity appears, the viscous instability grows rapidly. **?** suppress this with an additional, stronger switch that leads to instantaneous post-shock viscosity decay. We have experimented with this, and find it helps here but does not eliminate the viscous instability, and it leads to significantly larger particle noise in all the shock problems we consider below. Of course, we can evolve this problem perfectly with SPH if we simply disable artificial viscosity entirely, but then the method is disastrously unstable in real problems!

In grid methods, the numerical viscosity is much lower. However, as shown in § 4.1.2, advection errors in non-moving grids

[8] **?**, in their similar test problem (Fig. 8 therein), find that their method works well to ~ 5 orbits, which we confirm, but we should note several differences between the test problem there and here. They use an effectively higher resolution (equivalent to 128^2 in our setup) and use a carefully chosen initial particle distribution following **?** which minimizes the artificial viscosity noise, both of which delay breakup. They also set the minimum artificial viscosity in their method to zero, which gives good results on this test but we find leads to significant particle disorder and potentially catastrophic particle-interpenetration (where particles "move through" each other) in poorly-resolved shocks (very common in real problems). We find that the numerical parameters required for stable evolution in all other test problems shown here lead to somewhat faster breakup than the "ideal" parameters for this test problem alone.

[9] The standard prescription for "damping" artificial viscosity in SPH, in a supersonic disk, operates more slowly than the local dynamical time, hence the viscous instability can grow.

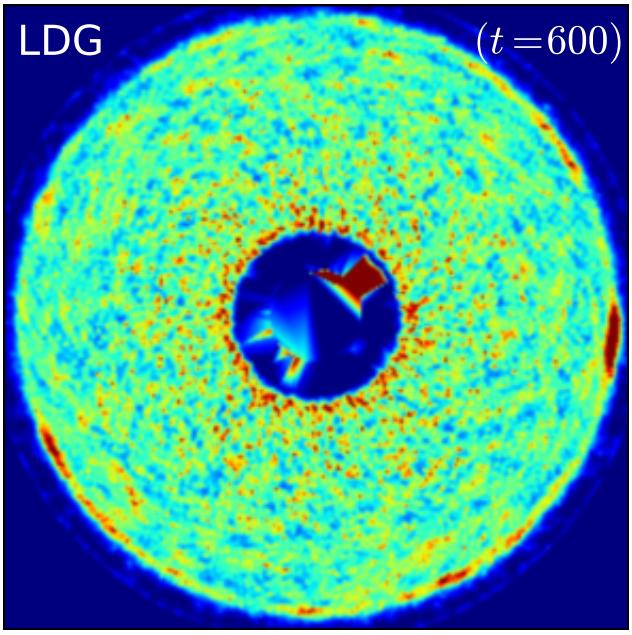

Figure 6. Keplerian disk as Fig. 5, at time $t = 600$ (not a typo)! The inner disk has executed > 160 orbits, at this time, without decaying or disrupting.

are serious. We find (as have many others before) that these very quickly diffuse the disk, spreading the mass around and seriously distorting the shape of the disk before completely destroying its structure within ~ 2 orbits. The inner parts lose angular momentum until they form a hot, hydrostatic center, and the outer parts are flung out. Total and local angular momentum are poorly conserved even over ~ 1 orbit (significantly more poorly than any other method we consider, including TSPH). This is well-known, and can be improved by going to higher resolution and higher-order methods, but even then the improvement is comparatively slow and the same qualitative effects occur. The problem is that the motion requires constant advection with the grid faces almost never aligned with the flow, in a circular trajectory which is not accurately approximated by second-order methods. Since the errors affect the whole disk volume simultaneously, going to AMR methods does not help.

These issues are even more severe if we rotate the disk relative to the axes (i.e. embed it in three dimensions, but tilt it out of the $x-y$ plane). The LDG, MFV, and SPH methods reproduce themselves to machine accuracy independent of such tilt. But in the structured grid codes, the advection errors above are compounded (by another mis-aligned axis). Moreover, the grid-alignment effect leads to an effective "numerical torque" which forces the orbits to align with the nearest coordinate axis (eventually pushing the disk back into the $x-y$ plane); this generates a large-scale warp in the disk on just ~ 1 orbital timescale. Such grid alignment effects are well-known (e.g. **?**). For example, (**?**) study cosmological simulations of galaxies in AMR and find that the galaxy spin axes are strongly aligned with the grid axes by low redshift, even at "effective" resolutions of $\sim 128^2 - 512^2$ in the disk plane (particle numbers in the disk up to 5×10^5); **?** demonstrate similar grid alignment and disk destruction effects in AMR simulations of stellar evolution and binary orbits up to AMR resolutions of $\sim 1024^3$. A variety of coordinate "patch" schemes or hybrid advection schemes have been designed to reduce these errors, but these all rely on some prior knowledge of the computational geometry. For the problem here, of course, we would obtain the most accurate results by using a pre-defined cylindrical coordinate system translated and rotated to center on and align with the disk. But while useful for some idealized problems, we specifically wish to study the more general case, since there are a huge range of problems (e.g. turbulence, galaxy formation and evolution, stellar evolution with convection and rotation, binary mergers, accretion from eccentric or gravitationally unstable disks, asymmetric SNe explosions) where the flow geometry cannot be completely determined ahead of time, or adaptive meshes must be used, so rotation cannot be perfectly grid-aligned.

Using moving meshes helps reduce the angular momentum errors from advection in fixed-grid (AMR) codes. However, we find that with AREPO, the disk goes unstable and the angular momentum evolution is still corrupted within a couple of orbits! This is actually inevitable in moving-mesh codes, and has been discussed in e.g. **??**. In a shearing disk, if the cells adapt in a Lagrangian manner, then they are inevitably deformed into a highly sheared/irregular shape. As soon as they become non-spherical (or more accurately fail to be exactly radially symmetric about their own cell center of mass), then moving a cell (even in isolation for a single timestep) no longer conserves angular momentum (indeed, the angular momentum of an irregular cell cannot be defined exactly but only to the same order of integration accuracy as the local velocity gradient estimator). Mass advection between cells leads to additional angular momentum errors. If some regularization/remeshing procedure is used to keep the cell shapes "regular" (as is necessary in *any* moving-mesh code used for real problems), then the remeshing means the cells cannot move entirely with the fluid and the gas must be advected over the cells. Since the coordinate system used for this in general problems is Cartesian, this introduces the same errors that we saw with fixed grid/AMR methods! There are various ways to improve the situation, but like with AMR codes these inherently depend on knowing the problem geometry ahead of time; for example, **?** design a moving grid which is a series of cylindrical shells free to rotate independently about a shared axis. But as noted above, this severely limits the range of applicability of the method.

Finally, we note that again there is no 1D analogue of this problem, but if we were to repeat our experiments for a 3D analogue (a cylindrically symmetric rotating tube) we would reach all the same conclusions. The purely geometrical disk thickness is not important; "thickness" matters only in the sense of the relative importance of pressure support versus angular momentum. In the limit of a "thicker disk" meaning a more pressure-dominated disk, the problem becomes progressively more hydrostatic and therefore "easier" for all methods considered here.

4.2 Shock and Non-Linear Jump Tests

We now consider several tests which probe the opposite regime: strong shocks.

4.2.1 Sod Shock Tube: A Basic Reimann Problem

check details of ICs.. still a bit suspicious of some behaviors here as artifacts of bad ICs; although snap 0's look good. could be slope-limiter effects?

We begin by considering one of the many simple Riemann problems used in standard code tests. We simulate a one-dimensional Sod shock tube with a left state ($x < 0$) described by $P_1 = 1$, $\rho_1 = 1$, $v_1 = 0$ and right state ($x \geq 0$) with $P_2 = 0.1795$, $\rho_2 = 0.25$, $v_2 = 0$, and $\gamma = 1.4$. These parameters are used in many code tests (**?????**). We intentionally consider a "low" resolution test, in which we place an initial 100 particles in the range

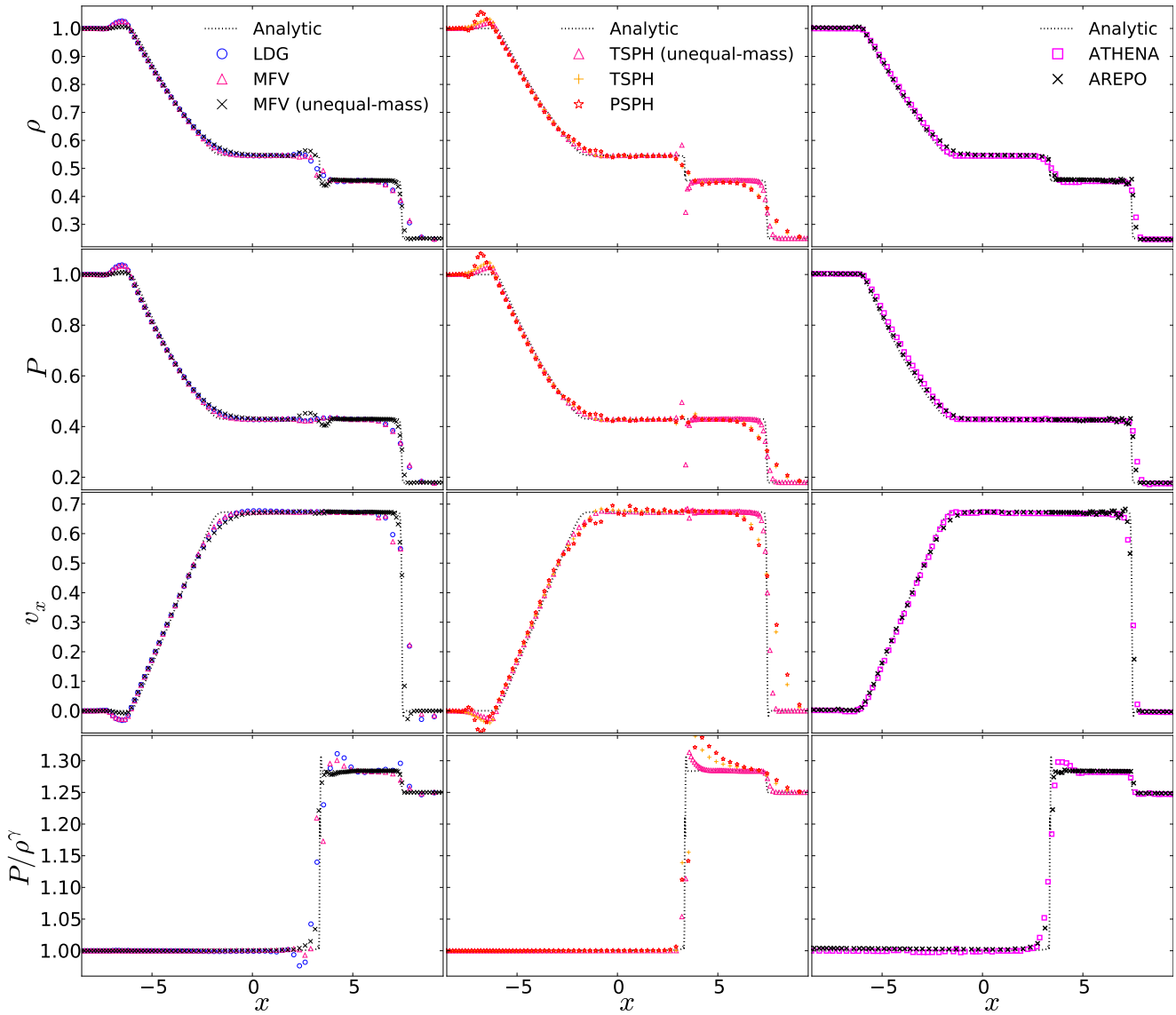

Figure 7. One-dimensional Sod shock tube § 4.2.1. From top to bottom, we plot density ρ, pressure P, velocity v_x, and entropy P/ρ^γ. We show the analytic solution (dotted) for each, compared to different methods (all with 100 resolution elements): all perform reasonably well with subtle differences. *Left:* LDG and MFV methods: both are very similar, with a small "bump" at the rarefaction and contact discontinuity due to the slope-limiter which rapidly converges away at higher resolution. "Default" cases shown assume equal-particle masses; in the "unequal-mass" case the ICs feature a factor $= 4$ jump in particle mass at the contact discontinuity (hence sharper resolution in the low-density shock). *Middle:* SPH methods: The "bumps" are larger, especially using PSPH, shocks are more smoothed, there is some velocity "ringing" in the rarefaction, and there is the known "pressure blip" around the contact discontinuity which does not converge away. *Right:* Moving-mesh (AREPO) and fixed-mesh (ATHENA) methods: these have the sharpest shock-capturing; but still feature weak "bumps" (ATHENA) or post-shock oscillations (AREPO).

$-10 < x < 10$ (spacing $\Delta x \approx 0.01$, 0.2, respectively). We plot results at $t = 5.0$.

All calculations here capture the shock and jump conditions reasonably, but there are differences. For all the non-mesh methods, it makes a difference whether we initialize the problem with equal-mass particles, or with the initial discontinuity corresponding to a particle mass jump (in which case the particle masses change discontinuously by a factor ~ 4 at the contact discontinuity). At the front at $x \sim -6$, all methods produce a small 'bump' in the density and corresponding dip in v_x; this is minimized in the grid codes and the unequal-mass particle MFV model: both the MFV method and unequal initial particle spacing minimize this relative to the TSPH and LDF methods. The 'bump' is amplified by the PSPH method. In PSPH there is also added noise where the pressure becomes flat ($x \sim 0$).

not so sure about this.... should the blip be there with unequal mass? At the contact discontinuity ($x \sim 3$), LDG and MFV methods with equal-mass particles behave well; the large particle-mass discontinuity in the unequal-mass case requires (because particle *volumes* are kernel-determined and vary smoothly) a 'blip' in the density, which then appears in the pressure. In SPH, however, a comparable blip appears even with equal-mass particles, and it is much more severe with unequal-mass particles; most importantly, the 'blip' converges away in the new methods (a modest-resolution MFV run with just 500 particles is indistinguishable from the dotted line shown), while the SPH blip never converges away (it gets

narrower but higher-amplitude at higher resolution). Fixed-grid and SPH methods also produce an "entropy overshoot" on the right side of the contact discontinuity; this is particularly strong in the equal-mass SPH examples. We should note that on this problem, non-conservative SPH methods (see Table ???) produce disastrous errors (easily order-of-magnitude deviations from the real solution, often with comparable particle-to-particle scatter) behind the shock front **??? should we show this? appendix?**

At the shock ($x \sim 8$), the methods are similar when the resolution is similar. The "equal particle mass" case feature a broader shock, but only because the mass choice means the spatial resolution is lower in this region: the number of particles needed to capture the shock is actually very similar. For the MFV and LDG methods, this is $\sim 3-4$ particles (< 1 kernel), only slightly larger than the $\sim 2-3$ grid cells needed in a second-order grid method; for SPH it is $\sim 7-8$ particles (~ 2 kernels). As noted in **?**, the moving mesh exhibits significant post-shock velocity oscillations, despite the slope limiter employed (we see the same with no slope limiter in the MFV and LDG methods, so suspect it is sensitive to the slope-limiting procedure).

Note that for all codes, we obtain essentially identical results if we solve this problem in 2D or 3D (i.e. as a true "tube") with the fluid having constant properties along the y and z directions (and periodic boundaries). In fixed-grid codes, it is well-known that if we rotate the tube so it is not exactly aligned with a coordinate axis, the correct solution is still recovered but shock jumps and contact discontinuities are diffused across ~ 2 times as many cells in the direction of motion. The particle and moving-mesh methods are invariant to rotations of the tube.

4.2.2 *Interacting Blastwaves: Complicated Jumps & Extreme Reimann Problems in 1D*

Another related one-dimensional test problem is the interaction of two strong blast waves, from **?**. We initialize gas with density $\rho = 1$, $\gamma = 1.4$, $v = 0$ in the domain $0 < x < 1$, with pressure $P = 1000$ for $x < 0.1$, $P = 100$ for $x > 0.9$, and $P = 0.01$ elsewhere. The boundary conditions are reflective at $x = 0$ and $x = 1$. This features multiple interactions of strong shocks and rarefactions, and provides a powerful test of Reimann solvers (many crash on this test, and essentially all codes require some hierarchy of solvers as we have implemented). We use 400 particles initially evenly spaced and equal-mass.

This is a problem where SPH does very well, actually. As in § 4.2.1, the shocks are smeared over more particles compared to other methods, and a small density "blip" appears near $x \sim 0.75$, but the structure of the density peaks is captured well even at low resolution. Moving mesh codes perform similarly well, with sharper shock resolution (and no "blip")[10]

At this resolution, both the LDG and MFV methods give results similar to moving meshes. The LDG method broadens the discontinuity at $x \sim 0.6$ by slightly more and similarly smooths the leading edge of the discontinuity at $x \sim 0.85$. The major difference is that both do not capture the full density dip without going to higher resolution (perhaps surprising given SPH's success, but this is where the Reimann solver has difficulty). But we confirm that at high resolution, the LDG and MFV methods converge to the same solution in good agreement with AREPO.

[10] Comparing the high-resolution results from the AREPO and TESS codes to our reference solution, however, shows a puzzling result, that the density jump at $x \sim 0.8$ is very slightly offset to lower x, even in the converged solution, with a narrower peak. We suspect **???**

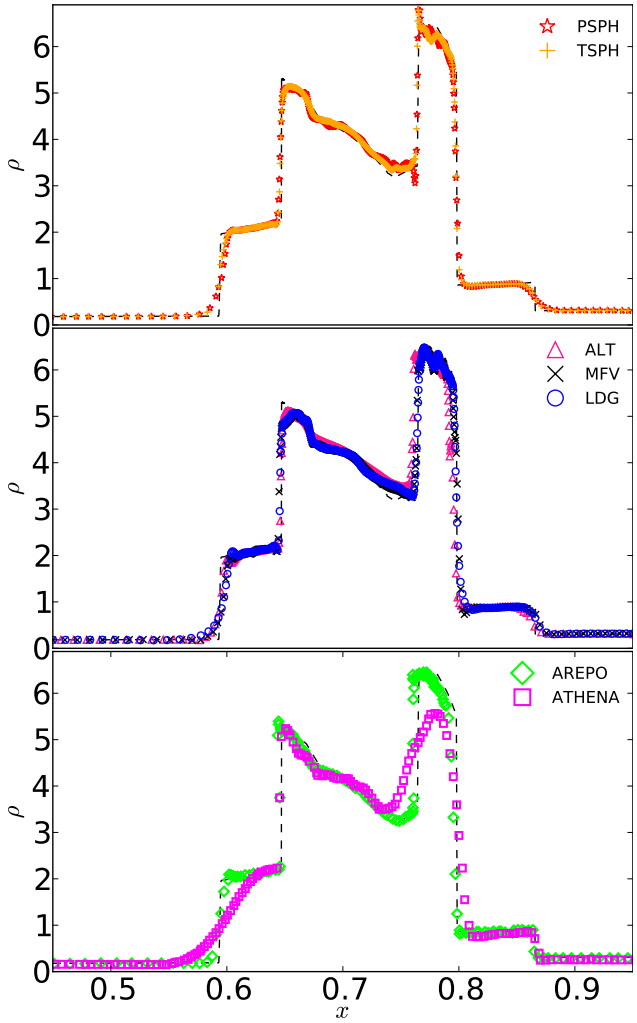

Figure 8. One-dimensional interacting blastwave test § 4.2.2. We compare all methods (computed with 400 resolution elements from $0 < x < 1$) to a reference solution computed using ATHENA with 10^5 cells, a third-order PPM solver, an exact Riemann solver, and Courant factor $= 0.1$. *Top:* SPH methods: these do well here. Contact discontinuities at $x \sim 0.6$ and $x \sim 0.85$ are noticeably smoothed and there is a "pressure blip" at $x \sim 0.75$, but the jumps are all captured. *Middle:* LDG & MFV methods: These also do well. The discontinuities are slightly less smoothed than SPH, but the density "dip" at $x \sim 0.75$ is not quite as well-traced, and there is some smoothing of the jump at $x \sim 0.65$. *Bottom:* Moving mesh & stationary mesh methods. Moving meshes do well, with the sharpest jumps and no "wiggles" in density at $x \sim 0.75-0.8$, but are slightly offset in the shock position. Stationary grids are noticeably less accurate than the other methods, severely smoothing the jump at $x \sim 0.6$ and the density peak from $x \sim 0.75-0.8$.

The largest errors at fixed resolution come from the fixed grid code. As noted in **?**, both the discontinuity at $x \sim 0.6$ and the density peak/pair of discontinuities around $x \sim 0.75$ are severely smoothed, the jump at $x \sim 0.8$ is more broadened than in any other method, and the density "dip" is captured but actually over-estimated. This stems largely from contact discontinuities being advected through the grid.

As in § 4.2.1, we obtain identical results solving this problem as a 2D or 3D "tube", except that if the tube is not exactly aligned with the grid, non-moving grid methods will diffuse it even more severely.

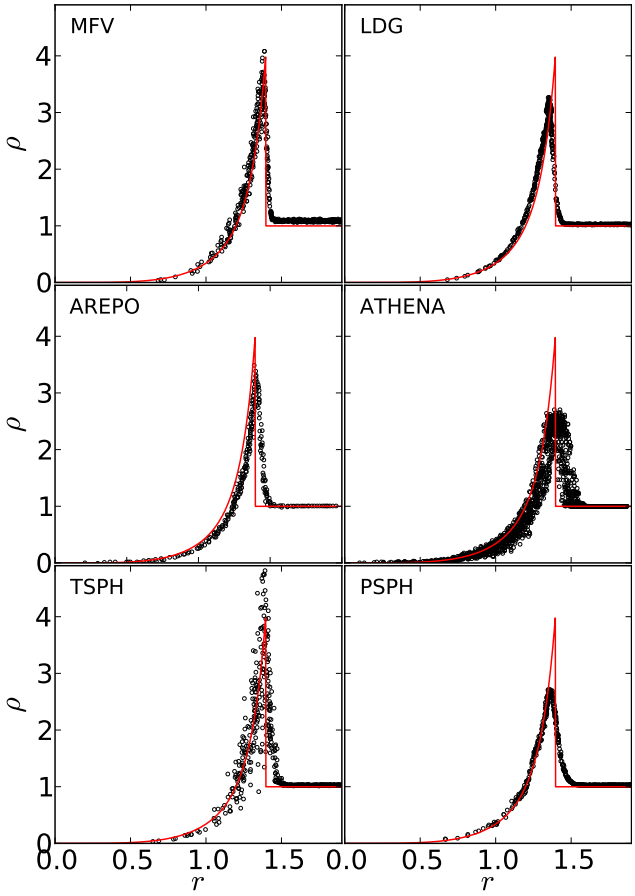

Figure 9. Three-dimensional Sedov-Taylor blastwave (§ 4.2.3). We plot the radial density profile at time $t = 0.06$; each point is one particle/cell (for clarity we plot only a random subset of cells) at 64^3 resolution; red line is the analytic solution. *Top:* MFV & LDG solutions: the MFV shows excellent capturing of the shock jump, but is slightly noisier than LDG. *Middle:* Moving-mesh (AREPO) and stationary-mesh (ATHENA) solutions: the moving-mesh solution lies "in between" our LDG and MFV solutions (there is a slight offset in shock position, which may result from the particular timestep scheme); the stationary-mesh solution is substantially more noisy and diffuses the shock (suppresses the jump) significantly. *Bottom:* SPH solutions: TSPH captures the jump, but is much noisier than any other method (and speads the jump over more particles). PSPH suppresses this noise via artificial conductivity, but this suppresses the jump amplitude and diffuses the leading-edge of the shock.

4.2.3 *Sedov Blastwaves: Conservation, Integration Stability, & Symmetry*

Here we consider a Sedov-Taylor blastwave, a point explosion with large Mach number. This provides a powerful test of the accuracy of code conservation, as well as of how well codes capture shock jumps and preserves symmetry in three dimensions. When adaptive (non-constant) timesteps are used (as they are in our code) this is also an important test of the integration stability of the method (see **?**, who show how various simple integration schemes become unstable).

We initialize a large domain with $\rho = 1$, $P = 10^{-6}$ (small enough to be irrelevant), and $\gamma = 5/3$, with 64^3 particles in the domain affected by the blastwave; we inject an energy $E = 1$ into the central **???**. We compare results at $t =$ **???**.

As expected, at fixed particle/cell number, fixed-grid methods smooth the shock jump significantly compared to Lagrangian methods (which by definition end up with more resolution in the shock). Conversely the deep interior structure of the blastwave (where densities are low and temperatures high) is better-resolved in fixed-grid methods; it depends on the problem of interest whether this is an advantage or disadvantage. However all grid codes (AMR or fixed) also suffer from variations of the carbuncle instability, in which shocks preferentially propagate along the grid axes; we see that this has a significant effect on the blast geometry, giving it an "eight

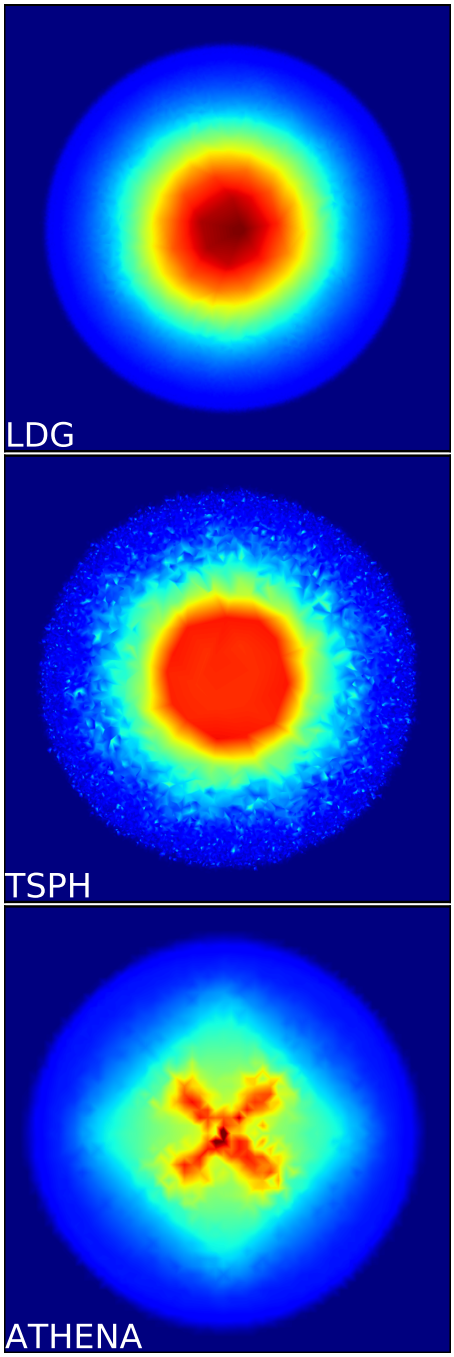

Figure 10. Sedov blastwave from Fig. 9; here we plot the gas internal energy (log-scaled from < 4 in dark blue to 2000 in red) in a 2D slice through $z = 0$, at $t = 0.06$. *Top:* LDG: The solution is smooth and shows good spherical symmetry. *Middle:* TSPH: The solution is spherical on average, but the severe noise is again visible. *Bottom:* Stationary-grid: Grid effects on the symmetry are clearly visible (the cross/diamond shapes).

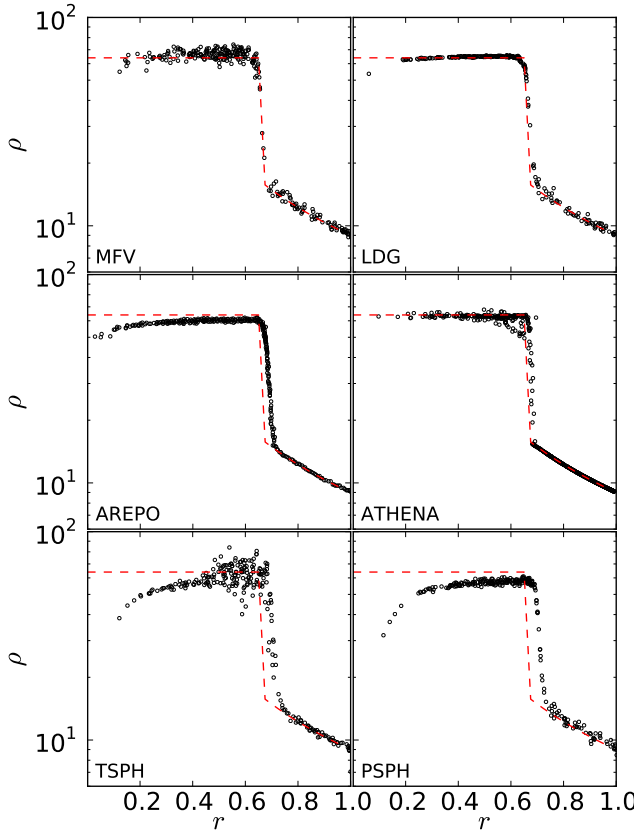

Figure 11. Three-dimensional Noh implostion problem (§ **??**). We plot the radial density profile at time $t = 0.2$; each point is one particle/cell (for clarity we plot only a random subset of cells) at 50^3 resolution; red line is the analytic solution. *Top:* MFV & LDG solutions: the MFV shows excellent capturing of the shock jump, but is noisier than LDG. *Middle:* Moving-mesh (AREPO) and stationary-mesh (ATHENA) solutions: the moving-mesh solution lies "in between" our LDG and MFV solutions in noise level, but the offset in shock position corresponds to a systematic under-estimate of the density jump, and the wall-heating is slightly more severe. The stationary-mesh solution gets the jump right (and is the only example without wall-heating), but with serious noise and asymmetry related to the carbuncle instability (see below). *Bottom:* SPH solutions: TSPH captures the jump but exhibits severe noise, shock-spreading, and wall-heating errors. PSPH suppresses the noise, but at the expense of more diffusion and enhanced wall-heating.

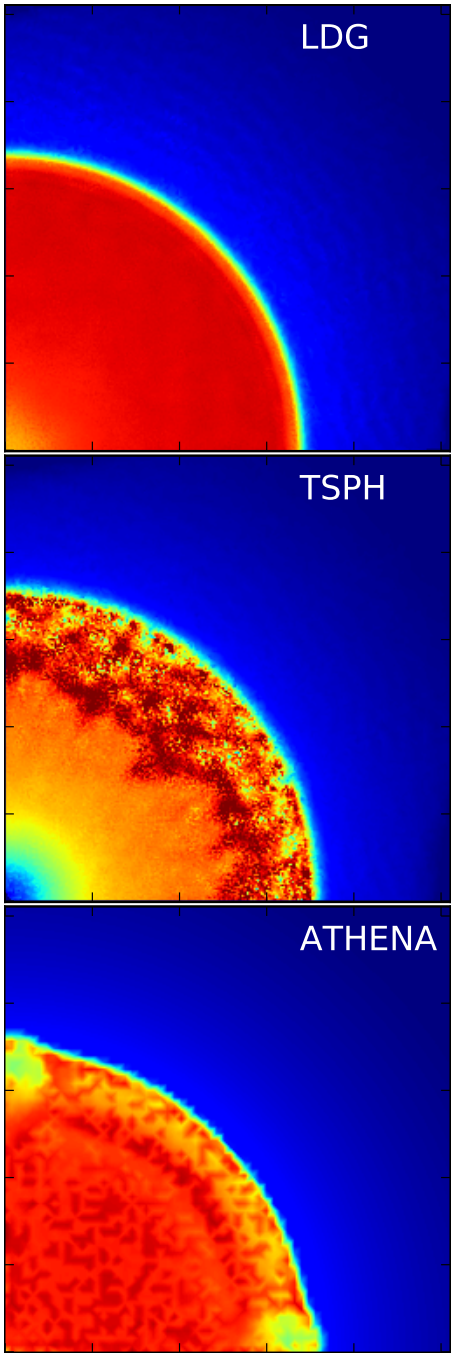

Figure 12. Noh implosion test from Fig. 11; we plot an image of the gas density (from 0 in black to > 64 in red), in a 2D slice through $z = 0$, at $t = 2$. *Top:* LDG: As in the Sedov test, the solution is smooth and shows good spherical symmetry, as it should. *Middle:* TSPH: The solution is spherical on average, but severe noise is again visible (there should be no internal structure here). *Bottom:* Stationary-grid: The carbuncle instability leads to the "hot spots" where the shock is propagating along the coordinate axes.

pointed" morphology along the grid axes which only decreases in time because diffusion tends to isotropize the blastwave.

The LDG, MFV and moving mesh methods perform similarly well here. In all cases the jump is better captured (less "smeared"), giving a maximum density ~ 3.5 (compared to the perfect case $= 4$) instead of ~ 2.7. All maintain excellent spherical symmetry in the shock front. Although a carbuncle instability still exists for moving mesh codes, it is substantially suppressed here. The meshless methods (LDG, MFV, SPH) simply have no such instability because there is no preferred axis. **something on velocity ringing???**

SPH methods generally do ok on this problem, except that the shock is spread out further (see § 4.2.1) and they give noisy solutions in the post-shock behavior unless some additional diffusion is added.[11] PSPH substantially enhances this noise, in fact, without additional diffusion. Adding artificial conductivity dramatically reduces the noise in all implementations, but at the cost of suppress-

[11] The noise arises from the E0 error when particles move through the shock.

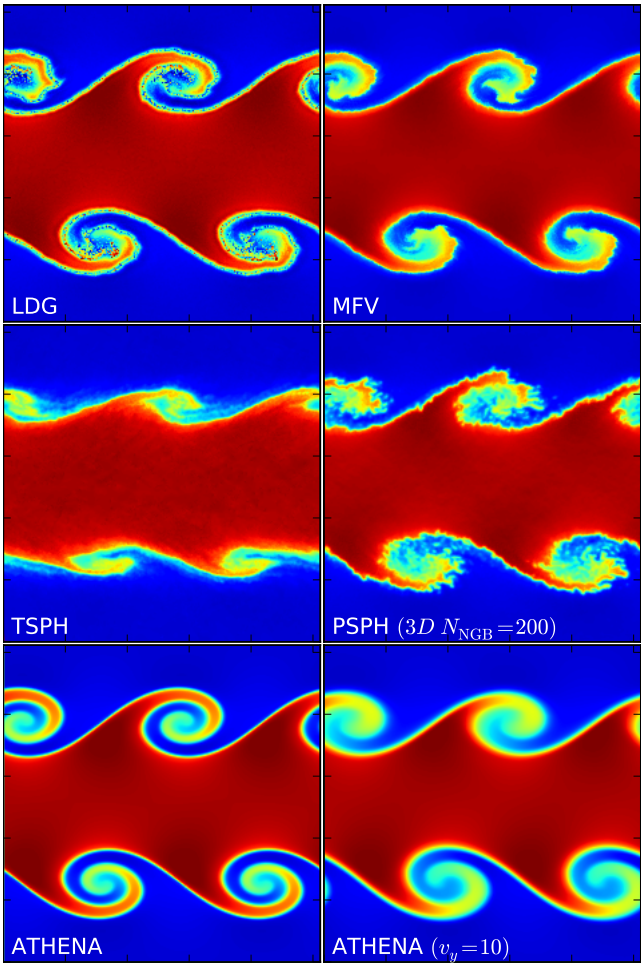

Figure 13. Kelvin-Helmholtz instability (§ 4.3.1). We compare the result of a 2D, 256^2 KH test problem at $t = 2.1$, where the rolls should be going non-linear. *Top:* In the LDG & MFV methods, the rolls are well-captured (with just the standard, small neighbor number, a 3D equivalent of $N_{\rm NGB} = 32$). There are small differences in the secondary structures developing, discussed below. *Middle:* SPH: In TSPH, a combination of surface tension and E0 errors suppress KH roll formation. In PSPH, for this initial condition, the noise is large enough that eliminating the surface tension alone does not help; we must also go to very large neighbor number to see rolls. Even then, the small-scale structure is corrupted by E0 errors. *Bottom:* Fixed-grid code. Symmetry is perfectly preserved, while diffusion suppresses small-scale (grid-seeded) modes, as the expense of structure inside the whorls. If we boost the fixed-grid solution by a uniform $v_y = 10$ (*right*), the diffusion increases (at resolution $< 128^2$, this "wipes out" the instability).

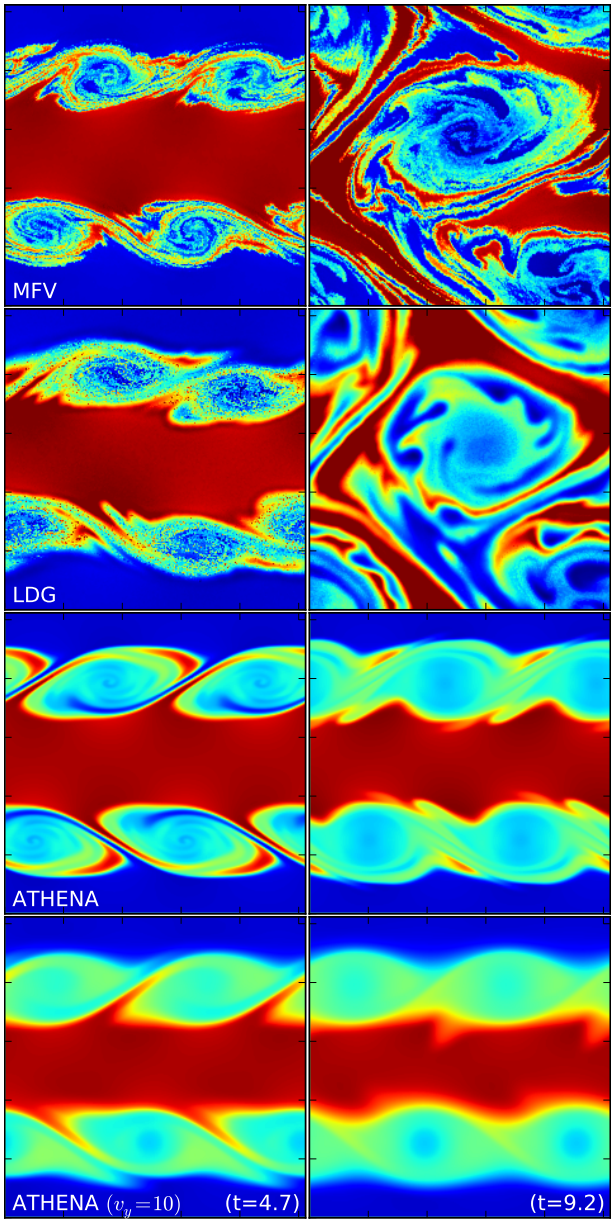

Figure 14. Non-linear evolution of the KH instability in Fig. 13, at $t = 4.7$ and $t = 9.2$. In MFV (*top*) & LDG (*second from top*) calculations, the sub-structure of the rolls is well-preserved; so they continue to "roll up" until they overlap, leading to the entire box going non-linear. The sub-structure of the non-linear rolls is especially well-preserved in the MFV calculation (remember this is only 256^2!). In contrast, in the stationary grid codes, with (*bottom*) or without (*second from bottom*) a boost applied, the rolls eventually diffuse into one another, until the non-linear state simply becomes two streams with a thick "boundary layer." Much higher resolution in grid codes is required to reduce this diffusion and see the full box go non-linear, or to see the same sub-structure in the rolls at late times.

ing the shock jump and creating an unphysical "leading" temperature jump (diffusing the entropy jump *ahead of* the shock).

A fairly extensive comparison of ~ 10 different SPH variations for this problem is shown in **?** (Figs. 1-3 therein). As shown there, using a "consistent" but non-conservative SPH method almost immediately leads to large numerical errors dominating the real solution (and runaway growth of the momentum errors). Similar catastrophic errors appear if one uses adaptive timesteps but removes the timestep limiter from **??**. Using an SPH method which does not explicitly include correction terms for the spatial gradients of the smoothing length (as in SPHS, GASOLINE, and many other non-Lagrangian SPH codes) simply leads to the shock being in the wrong place, even if the code conserves energy. Interestingly, we do not need to include explicit "∇h" terms to get the shock in the correct place with the LDG or MFV method, although the addition of such terms could well improve the code accuracy further.

If we solve this problem in 2D the differences between methods are qualitatively identical, but slightly reduced in magnitude. A 1D analogue is essentially a Riemann problem (see § 4.2.1).

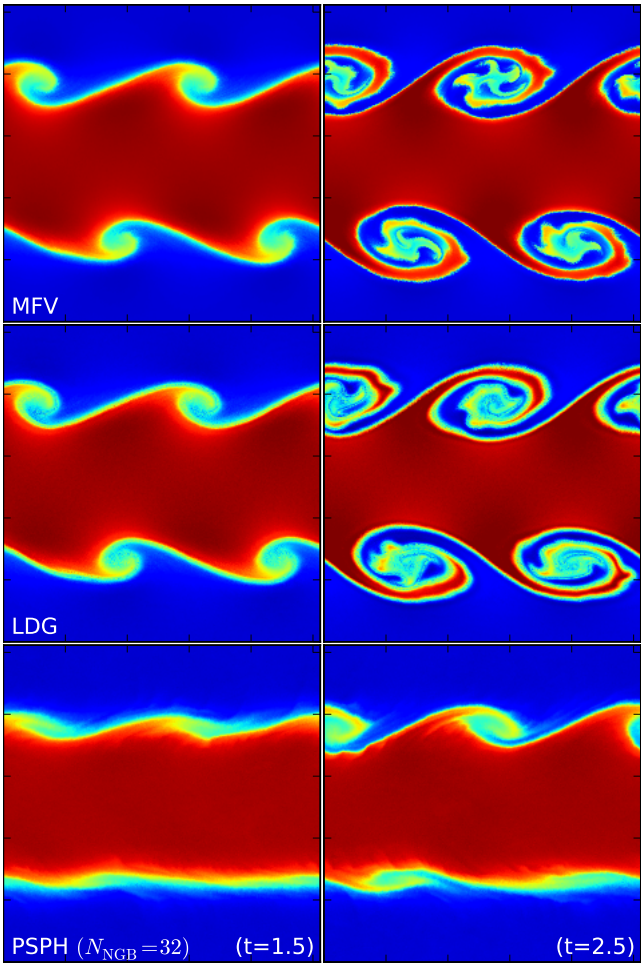

Figure 15. Evolution of a 3D (256^2x16) version of the KH instability from Figs. 13-14 at earlier times $t = 1.5$ and $t = 2.5$. The 3D instability is captured as well as to the 2D instability. Note that PSPH with low $N_{\rm NGB}$ (shown explicitly) still fails here. Also note that the early-time (linear and early non-linear) growth is nearly identical in MFV and LDG calculations; only later into the non-linear evolution do we see the differences from Figs. 13-14

4.2.4 *The Noh (Spherical Collapse) Test: Extreme Shock Jumps to Break Your Solver*

Next consider the **?** test. This is a challenging test: many codes cannot run it without crashing, or run it but see catastrophic examples of the carbuncle instability. **?** noted only four of eight schemes they studied could actually run the problem. An arbitrarily large domain[12] is initialized with $\rho = 1$, $\gamma = 5/3$, vanishing pressure, and a radial inflow (directed towards the origin from all points) with $|\mathbf{v}| = 1$ ($v_r = -1$). The analytic solution involves an infinitely strong shock with constant internal density moving outwards at speed $= 1/3$, with a density jump of $4^3 = 64$ at the shock in 3D.

We focus on the 3D case since it is considered the most difficult. All our "default" setups run on the problem, but we confirm that several approximate Riemann solvers can fail at the shock (re-

[12] For the particle codes, we simply use a huge domain so that we do not have to worry about boundary conditions. For the grid codes complicated explicit setting of inflow boundary conditions is possible and has been done here, but at fixed time it is identical to the result with a sufficiently large domain.

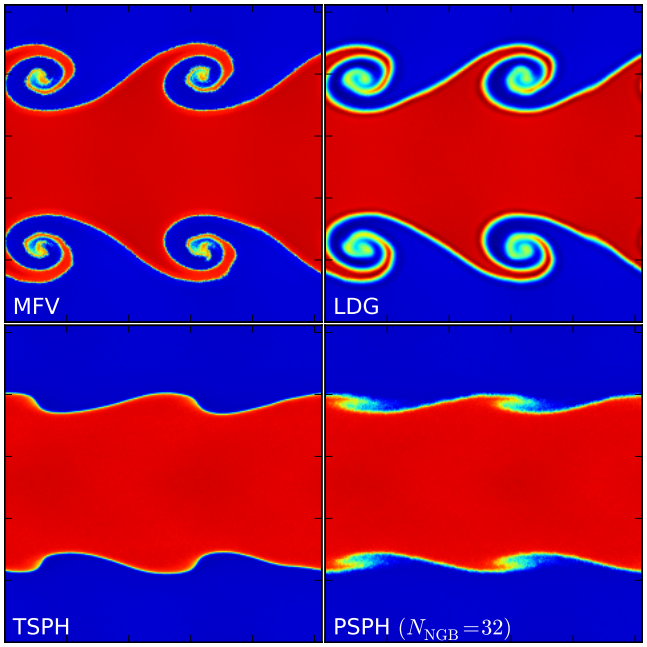

Figure 16. Alternative 3D (256^2x16) KH instability test from the Wengen suite, where the ICs include a perfectly sharp contact discontinuity (as well as different shear & seed modes from the previous test), at time $t = 3.75 \approx 1.1\,\tau_{\rm KH}$. *Top:* MFV & LDG results: the sharp discontinuity does not suppress mode growth. Here the ICs are symmetric, and we see excellent preservation of symmetry even in the non-linear parts of the rolls. As before the LDG method smears the fluid phase boundaries slightly; the MFV method preserves a sharp contrast. *Bottom:* SPH results with the same neighbor number ($N_{\rm NGB} = 32$); both TSPH and PSPH fail to capture the instability in this case.

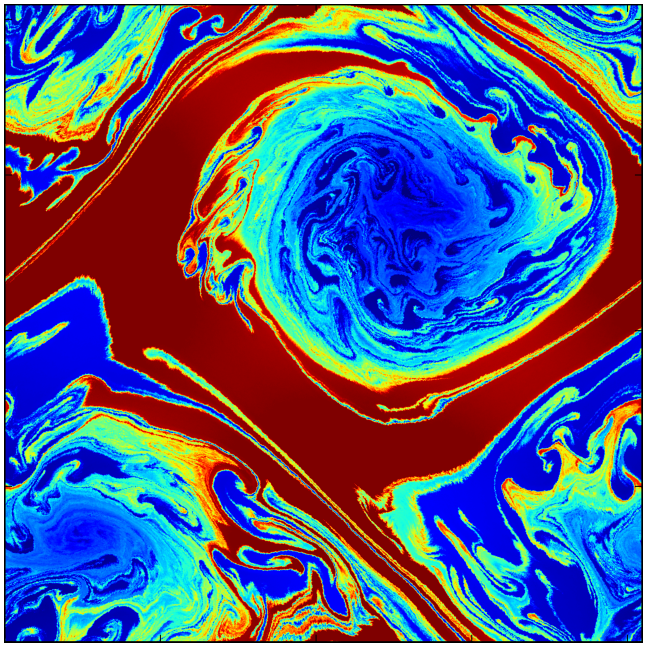

Figure 17. 2D KH instability at high-resolution (1024^2) with the MFV method, at time $t = 10$.

quiring a hierarchy of solvers). We also confirm the well-known result that in particle-based codes, an initial lattice is a pathological configuration (especially for this problem), leading to singular particle distributions (similar problems arise if initializes the moving mesh from a regular lattice); we therefore use a glass for our ICs (except in ATHENA). **move to zeldovich discussion???: and that our condition-number based slope-limiter is can be important in rare but dangerous situations where, near the shock, an ill-conditioned gradient matrix arises (because particles can be much more densely packed in one dimension post-shock).** The density profile is shown quantitatively in Fig. 11, and the spatial structure of the shock in Fig. 12.

The LDG and MFV methods give similar results here. The shock position is recovered accurately, and the shock is appropriately spherical and smooth (there is no carbuncle instability or preferential shock propagation direction). The jump is recovered very well even at this low resolution. Both have some post-shock noise in ρ owing to post-shock oscillations, but this is much weaker in the LDG result. The pre-shock ρ field also has noise which is geometrically induced (since the initial particle/mesh distribution is a glass, as opposed to a perfectly spherically symmetric lattice). Both feature some (weak) suppression of the density near the origin owing to wall-heating (as do many other codes, see **??**).

The moving-mesh method is also similar; the noise level lies between our LDG and MFV methods. As we saw with the Sedov test, the shock position is slightly offset (leading the analytic solution) and the jump is slightly under-estimated. This will eventually converge to the correct jump; and the offset may owe more to the timestepping scheme than the numerical method. The wall-heating is noticeably more severe than in the LDG or MFV methods.

In the fixed-grid code, the carbuncle instability is particularly prominent – this actually seeds most of the noise around the jump.[13] The instability is evident as the "hot spots" along the Cartesian grid axes, which at the time shown have begun to propagate faster than the rest of the shock. In ATHENA there is very little wall-heating, though this is not generally true of grid codes.

As in the Sedov test, traditional SPH dramatically enhances the noise compared to all other methods. It has no carbuncle instability but seeds considerable spurious shock structure. It also has the most severe wall-heating. The noise is reduced by adding artificial conductivity and a larger kernel in PSPH, but still exceeds most other methods, and this makes the wall-heating more severe still. Both TSPH and PSPH spread the shock well ahead of the analytic solution: this weakens the shock jump, and it requires significantly higher resolution to capture the exact jump condition.

Finally, if we consider the 2D version of this problem, as in § 4.2.3, the qualitative results are identical, but the shock jump is weaker ($4^2 = 16$ in density) and easier to capture, so the quantitative differences between methods are reduced, and all methods converge to the exact solution more rapidly. The 1D analogue (collapse along a line) is a much less interesting test because many of the challenges (pathological grid setups in particle methods, the carbuncle instability, the large density jump, preservation of symmetry in the face of grid noise) are eliminated.

[13] Note that we have run this with the "standard" version of ATHENA, which is very similar to AREPO in "fixed grid" mode, and gives similar results at fixed resolution to AMR codes like RAMSES (which we have also compared), FLASH, and PLUTO (see **?**). As noted in **?**, this can be cured with the addition of problem-specific additional dissipation in the correct places (and the pre-packaged ATHENA Noh test problem uses this approach). However we wish to compare the more general behavior in their "default" mode for all codes here.

4.3 Fluid Mixing Tests

The next set of tests focuses on various fluid instabilities which are ubiquitous in astrophysics and many other areas of fluid dynamics, especially any regimes where turbulence and/or mixing are important. Considerable attention has been paid in the literature to difficulties of SPH methods in dealing with these instabilities (see e.g. **??????????**). And in response many improvements have been made to SPH, which allow it to better handle such instabilities (see **??????????**). However, as pointed out in **?**, comparatively little attention has been paid to difficulties faced by stationary-grid codes in this regime. As shown therein (see Figs. 33 & 36 there), the fact that such codes have non-Galilean invariant errors means that simply assigning the whole fluid a bulk velocity comparable to, say, the shear velocities (for a Kelvin-Helmholtz problem) or "sinking" velocity (for a Rayleigh-Taylor problem) will substantially change the solution and can easily wipe out the instabilities entirely at low resolution. We therefore consider these in more detail below.

4.3.1 Kelvin-Helmholtz Instabilities

We will consider the Kelvin-Helmholtz (KH) instability in detail, since this has been the focus of most such tests of SPH and grid codes.

First, we consider a two-dimensional setup from **?**. This is a KH initial condition with a non-zero thickness surface layer, and seeded mode, designed to behave identically in the linear regime in all well-behaved methods (as opposed to some setups, which depend on numerical error to seed the KH instability initially). The initial density and x velocity depend on the y direction as

$$\rho(y) = \begin{cases} \rho_2 - \Delta\rho\,\exp[(y-0.25)/\Delta y] & (0 \le y < 0.25) \\ \rho_1 + \Delta\rho\,\exp[(0.25-y)/\Delta y] & (0.25 \le y \le 0.5) \\ \rho_1 + \Delta\rho\,\exp[(y-0.75)/\Delta y] & (0.5 \le y \le 0.75) \\ \rho_2 - \Delta\rho\,\exp[(0.75-y)/\Delta y] & (0.75 < y \le 1) \end{cases} \tag{4}$$

$$v_x(y) = \begin{cases} -0.5 + 0.5\,\exp[(y-0.25)/\Delta y] & (0 \le y < 0.25) \\ 0.5 - 0.5\,\exp[(0.25-y)/\Delta y] & (0.25 \le y \le 0.5) \\ 0.5 - 0.5\,\exp[(y-0.75)/\Delta y] & (0.5 \le y \le 0.75) \\ -0.5 + 0.5\,\exp[(0.75-y)/\Delta y] & (0.75 < y \le 1) \end{cases} \tag{5}$$

with $\rho_2 = 2$, $\rho_1 = 1$, $\Delta\rho = 0.5\,(\rho_2 - \rho_1)$, $\Delta y = 0.025$, and constant pressure $P = 5/2$ with $\gamma = 5/3$ throughout a periodic domain of size $0 < x < 1$, $0 < y < 1$. The system is seeded with an initial y velocity mode:

$$v_y(x) = \delta v_y^0 \sin(4\pi x) \tag{6}$$

with $\delta v_y^0 = 0.01$. The exponential terms above are designed to be the smoothing layer described above, so that the initial mode is well-defined; but essentially, this is a constant-pressure fluid with a density contrast of a factor $= 2$ between two layers, with a relative shear velocity $= 1$. The linear KH growth timescale is usually defined as

$$\tau_{\rm KH} \equiv \frac{\lambda\,(\rho_1+\rho_2)}{(\rho_1\,\rho_2)^{1/2}\,|v_{x,1}-v_{x,2}|} \tag{7}$$

where λ is the mode wavelength (here $= 1/2$); so $\tau_{\rm KH} = 2^{-1/2} \approx 0.71$.

Fig. **??** shows the results at $t = 2.1$ for a 256^2 run. In the

non-SPH methods, the mode behaves as expected. The linear growth phase is almost perfectly identical between the LDG, MFV, moving-mesh, and fixed-grid codes (we have compared quantitatively with the linear-growth curves in **?**, Figs. ???-??? and find all these methods behave similarly). The instability grows at the shear layer and the peaks of each fluid phase penetrate further, until the non-linear shear leads them to roll up into the well-known KH "whorls." In the non-linear phase, we see differences begin to appear. This is further emphasized in Fig. 14, where we compare later times.

In the LDG and MFV methods, the whorl height is nearly identical to the stationary-grid results. However, unless the initial conditions in the particle codes are a perfect lattice (symmetrized exactly about the mode center and perturbation sinusoid), which is a pathological configuration, there is some seed asymmetry which we see amplified in these late times. We see in the non-linear phase, additional small-scale modes begin to grow (as they should). Here we can also begin to see that the MFV method, by allowing mass fluxes, can more sharply capture complicated contact discontinuities. In the late non-linear phases, it is truly remarkable how much fine-structure is captured by the MFV runs, given the relatively low resolution used. In these stages, we see the expected behavior: the rolls continue to grow until they overlap, at which point the box becomes non-linear and the two fluid layers "kink" leading to the merger of the rolls into bigger and more complex structures. This is consistent (and shows good convergence with) the behavior at higher resolution; Fig. 17 shows the state of the box at $t = 10$ in an MFV run at high resolution (1024^2), showing the same character and the exceptional degree of resolved sub-structure and small-scale modes.

Since the particle volume is continuous by definition, and initial particle masses are constant, the LDG method necessarily smooths the density field over ~ 1 smoothing length. This leads to less-detailed small-scale structure in the LDG method, and in the non-linear phase to enhanced diffusion. However the behavior on large scales is similar – i.e. the LDG solution, even late into the non-linear phase, resembles a "smoothed" MFV solution, rather than departing. This is important since it demonstrates the second-order advection errors in the LDG method do not corrupt fluid mixing instabilities even in late-time, non-linear stages, where the true (physical) Lagrangian volumes of a fluid parcel would be distorted into arbitrarily complex shapes.

On the other hand, the symmetry of the ICs is manifest more obviously in the stationary-grid codes. However, the stationary grid methods are more diffusive: we see the whorls diffuse away after about one roll, and at all times there is a relatively large "fuzzy" layer in their boundaries. Especially at late times, this completely changes the character of the solution. For this IC and resolution, the whorls simply diffuse into a smooth, thick boundary layer, and the instability shuts itself down! We do not see the expected late-time non-linear kinking behavior until we go to $\sim 2048^2$ resolution! As notes by **?**, if we "boost" the problem by adding a uniform velocity to all the gas (which has no effect on the lagrangian methods), the diffusion and errors in the stationary-grid results increase substantially. The additional diffusion is especially obvious in the non-linear (late-time) solutions. The diffusion is closely related to what we saw in the "square" test (§ 4.1.2): the "rolling" is the result of the contact discontinuity being stretched and distorted, and advected across cells in an increasingly irregular (non-grid aligned) fashion. Hence the diffusion grows as time passes and the rolls become more complicated. On the other hand, in Lagrangian, mesh-free methods, the arbitrary angles the rolls necessarily form as they "roll up" do not present any problems for advection of contact discontinuities.

SPH methods, as expected, have difficulty capturing the KH instability. **???** It is well-known that TSPH suppresses this instability, owing to a combination of the surface tension error and E0 force errors swamping the low-amplitude mode. PSPH eliminates the surface tension term, but the E0 error cannot be eliminated in a conservative SPH scheme, only reduced by going to much higher neighbor number. So if we use a TSPH or PSPH method with the same $N_{\rm NGB}$ as used for the MFV and LDG kernels, or as used in traditional SPH work, then we find in Fig. 19 that the mode simply does not grow (the E0 errors are still too large). *Only* if we use a higher-order kernel with more neighbors does the mode begins to grow appropriately: for this IC, we require a 3D-equivalent neighbor number $\gtrsim 128$. However, we see that even in this case, the small-scale modes appear corrupted, with a "shredded" morphology that does not resemble the filamentary+KH morphology expected. This is because the small-scale modes are corrupted in PSPH by the addition of the artificial conductivity term. Better-looking results can be obtained by using PSPH without conductivity, as in **?** (Fig. 11 there); however, this comes at the cost of severe noise in all problems with shocks/pressure discontinuities (much worse than the noise in TSPH, which we have already shown is worse than any other method we show here).

In Fig. 15, we consider a three-dimensional version of this instability: to construct this we simply extend the ICs with constant properties in the z direction, to a 256x256x16 periodic box. Here we see essentially identical qualitative behavior, as expected. We explicitly show the earlier stages of the LDG and MFV runs, to demonstrate that the linear mode growth is identical in both methods. The transition to 3D causes no problems for either method (if anything, the extra dimension means the condition numbers of the gradient matrices tend to be slightly better-behaved, so the errors are slightly smaller). The stationary-grid results are also essentially identical. The PSPH **???**

Finally, for the sake of completeness, because it is considered in many papers, we compare a different KH IC. Specifically, we consider the 3D KH test from the Wengen multiphase test suite[14] and described in **??**. Briefly, in a periodic box with size 256, 256, 16 kpc in the x, y, z directions (centered on 0, 0, 0), respectively, $\approx 10^6$ equal-mass particles are initialized in a cubic lattice, with density, temperature, and x-velocity $= \rho_1, T_1, v_1$ for $|y| < 4$ and $= \rho_2 T_2, v_2$ for $|y| > 4$, with $\rho_2 = 0.5\,\rho_1$, $T_2 = 2.0\,T_1$, $v_2 = -v_1 = 40\,{\rm km\,s^{-1}}$. The values for T_1 are chosen so the sound speed $c_{s,2} \approx 8\,|v_2|$; the system has constant initial pressure. To trigger instabilities, a sinusoidal velocity perturbation is applied to v_y near the boundary, with amplitude $\delta v_y = 4\,{\rm km\,s^{-1}}$ and wavelength $\lambda = 128\,{\rm kpc}$.

As expected from the previous tests, both LDG and MFV methods capture the instability with high accuracy. One benefit of this version of the KH test is that the ICs are designed to have much better symmetry for particle-based codes (while the **?** IC is optimized for grid codes), and as a result we directly see that the symmetry in the MFV and LDG simulations is well-preserved (i.e. the loss of symmetry in the previous simulation is not a result of the code, but of the ICs). Another useful aspect of this IC is that, unlike the previous IC, it has a true density discontinuity, across a single particle separation. We see that this is smoothed to ~ 1 softening in the LDG method (the green "edge"; still much less

[14] Available at `http://www.astrosim.net/code/doku.php`

than in a stationary-grid code), and preserved nearly perfect in the MFV code, despite the rolls having executed multiple "wraps." This discontinuity makes the problem even more challenging for SPH methods, and we see that essentially no KH growth occurs without going to very large neighbor number.

4.3.2 Rayleigh-Taylor Instabilities

We now consider the Rayleigh-Taylor (RT) instability, with initial conditions from **?**. In a two-dimensional domain with $0 < x < 1/2$ (periodic boundaries) and $0 < y < 1$ (reflecting boundary with particles at initial $y < 0.1$ or $y > 0.9$ held fixed for the non-grid methods), we take $\gamma = 1.4$ and initialize a density profile $\rho(y) = \rho_1 + (\rho_2 - \rho_1)/(1+\exp[-(y-0.5)/\Delta])$ where $\rho_1 = 1$ and $\rho_2 = 2$ are the density "below" and "above" the contact discontinuity and $\Delta = 0.025$ is its width; initial entropies are assigned so the pressure gradient is in hydrostatic equilibrium with a uniform gravitational acceleration $g = -1/2$ in the y direction (at the interface, $P = \rho_2/\gamma = 10/7$ so $c_s = 1$). An initial y-velocity perturbation $v_y = \delta v_y\,(1+\cos(8\pi\,(x+1/4)))\,(1+\cos(5\pi\,(y-1/2)))$ with $\delta v_y = 0.025$ is applied in the range $0.3 < y < 0.7$.

In Fig. 18 we show the evolution of the instability in a high-resolution (512x1024) run with the MFV method. As expected, the initial velocity grows and buoyancy drives the lighter fluid to rise, driving bulk motions. Secondary KH instabilities form on the shear surface between the rising/sinking fluids. The linear growth of the instability is nearly identical in MFV, LDG, ATHENA, and AREPO runs; however the non-linear dynamics start to differ. For example, in the particle methods, the vertical symmetry is eventually broken, albeit weakly. This is discussed at length in **?**, but is completely expected here, because the initial particle distribution is not perfectly mirror-symmetric with the seed mode; for any seed asymmetry, growth of the non-linear KH modes making it less symmetric is the physically correct solution. The only way to force symmetry in these methods is to use a very specific and usually pathological initial particle distribution.

Fig. 19 compares the non-linear RT evolution across different methods, with the same initial conditions at medium resolution (128x256). The MFV method appears to capture the most small-scale structure of any method we consider: this is because it is both Lagrangian and can follow contact discontinuities very sharply. The large-scale evolution of the LDG run is very similar to MFV; the growth of the RT mode is identical, but the structure of the secondary instabilities and boundaries is noticeably less sharp. As in the KH test, this is because the method enforces constant particle masses; so a contact discontinuity must necessarily be smoothed over at least one kernel smoothing length (while in the MFV method it could be captured, in principle, across two particles). The result is similar if we apply a "post-processing" density kernel smoothing to the MFV result. However both converge to the same result at high resolution.

It is well-known that TSPH suppresses this instability, owing to a combination of the surface tension error and E0 force errors swamping the low-amplitude mode. PSPH eliminates the surface tension term, but the E0 error cannot be eliminated in a conservative SPH scheme, only reduced by going to much higher neighbor number. So if we use a TSPH or PSPH method with the same $N_{\rm NGB}$ as used for the MFV and LDG kernels, or as used in traditional SPH work, then we find in Fig. 19 that the mode simply does not grow (the E0 errors are still too large). *Only* if we use a higher-order kernel with more neighbors does the mode begins to grow appropriately: for this IC, we require a 3D-equivalent neighbor number $\gtrsim 128$. However, we see that even in this case, the small-scale modes appear corrupted, with a "shredded" morphology that does not resemble the filamentary+KH morphology expected. This is because the small-scale modes are corrupted in PSPH by the addition of the artificial conductivity term. Better-looking results can be obtained by using PSPH without conductivity, as in **?** (Fig. 11 there); however, this comes at the cost of severe noise in all problems with shocks/pressure discontinuities (much worse than the noise in TSPH, which we have already shown is worse than any other method we show here).

If the fluid is stationary with respect to the grid, a stationary-grid code performs excellently on this problem. We note that the growth rate and even non-linear height of the light fluid is almost identical between MFV, LDG, AREPO, and ATHENA runs. However, the stationary-grid ATHENA run captures both fine detail in the secondary instabilities while maintaining perfect symmetry (here, the problem is set up so the grid is exactly symmetric about the perturbation; otherwise this would not hold). However, as soon as we set the fluid in motion with respect to the grid, advection errors become significant at this resolution. We show the results if we "boost" the entire system by a horizontal velocity $v_x = 10$. Physically, this should leave the solution unchanged; and in all the Lagrangian methods it has no effect. But for stationary grids, it substantially slows down the mode growth rate (hence the RT plumes have not reached the correct locations), breaks the symmetry systematically (giving the fluid a "drift" which depends on the vertical location; this is a more serious error than random symmetry breaking because it implies a systematic shear velocity generated by the grid across the whole domain), and severely diffuses the fluid (wiping out the secondary structures). As in the KH test, because the whole volume is affected, an AMR scheme does not reduce this advection error.

As in the KH test, we note there is no 1D analogue of this test, but we see the essentially identical qualitative results whether we use 2D or 3D setups (with the same caution that ill-designed SPH ICs in 3D make the method's convergence and accuracy problems more severe).

4.3.3 The "Blob" Test: KH & RT Instabilities in a Supersonic, Astrophysical Situation

Next we consider the "blob" test, which is designed to synthesize the fluid mixing instabilities above (as well as ram-pressure stripping) in a more "realistic" example of astrophysical interest representative of a multi-phase medium. The initial conditions come from the Wengen test suite and are described in **?**: we initialize a spherical cloud of uniform density in pressure equilibrium with an ambient medium, in a wind-tunnel with period boundaries. The imposed wind has Mach number $\mathcal{M} = 2.7$ (relative to the "ambient" gas) with the cloud having a density $= 10$ times larger than the ambient medium. The domain is a periodic rectangle with dimensions $x, y, z = 2000, 2000, 6000\,$kpc (the absolute units are not important), with the cloud centered on $0, 0, -2000\,$kpc; 9.6×10^6 particles/cells are initialized in a lattice (with equal-masses in the particle-based methods).

Fig. 20 shows the cloud morphology versus time. The wind-cloud collision generates a bow shock and begins to disrupt the cloud via KH and RT instabilities at the interface; within a few cloud-crossing timescales the dense material is well-mixed (the cloud is destroyed). Various additional shock fronts appear because of the periodic boundary conditions leading to the bow shock interacting with itself. The qualitative behavior is similar in our LDG and MFV results (see also **?**, Fig. 7-8, who find the same with their implementation of an MFV-like scheme), and in grid-based codes

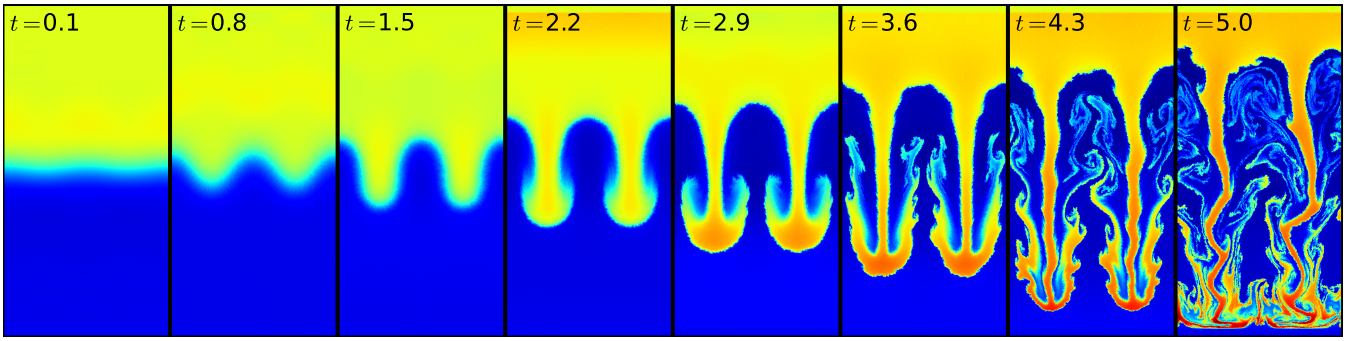

Figure 18. Rayleigh-Taylor instability test (§ 4.3.2). We plot density, from $0.8-2.8$ (black-red), in a two-dimensional simulation. Panels show the evolution of the RT instability using the MFV method at high resolution (512x1024), at different times. The linear growth of the instability is nearly identical in MFV, LDG, moving-mesh, and fixed-grid runs; in all cases it grows and secondary KH instabilities appear along the rising/sinking streams. Note the fine resolution of contact discontinuities and mixing. This run uses our standard number of particle neighbors: for both LDG and MFV runs, the instability develops regardless of the number of neighbors used (we have tested from $\sim 8-64$ in 2D). The breaking of vertical symmetry in the non-linear phase is expected from the problem setup.

including moving meshes (**?**, Figs. 4-5), fixed Cartesian grids, and AMR schemes (see **?**, Figs. 4-10). Note in particular the good agreement between MFV and LDG results for the small-scale structure of the shredded cloud and the sharp capturing of the shock fronts.

Quantitatively, Fig. 21 follows **?** and measures the degree of mixing. At each time we measure the total mass in gas with $\rho > 0.64\,\rho_c$ and $T < 0.9\,T_a$ (where ρ_c and T_a are the initial cloud density and ambient temperature). We compare our results here with a compilation from other methods in **?**. **???** For a stationary-grid result, we use the published result from ENZO, and AMR code, run with an effective resolution about equal to our runs here. The LDG, MFV, and stationary-grid results agree quite well. The cloud is "completely mixed" by this definition within a couple of KH timescales (note that there is essentially no "small residual" beyond $t \sim 2.5$ at this resolution). The "bumps" at early times are real, and owe to the choice of boundary conditions (the repeated bow-shock self-interactions each time it crosses); we suspect they are suppressed in ENZO owing to a different implementation of the boundaries in that code.

However, in "traditional" SPH the cloud is compressed to a "pancake" but surface tension prevents mixing and a sizeable fraction survives disruption for long timescales – tens of percents of the cold, dense mass survives. This is remedied in "modern" SPH schemes such as PSPH (**??**). However, it is worth noting that if we neglect artificial conductivity, PSPH allows mixing in density, but entropy is still a particle-carried quantity which does not mix as easily as it should (see e.g. **?**); so the early-time behavior agrees well with the LDG, MFV, and grid methods, but there is a long "tail" of material which is not disrupted even at much later times ($\sim 1-10\%$ of the cloud). This is eliminated by adding an artificial conductivity or thermal diffusion term; however, there is some ambiguity (just as with artificial viscosity) regarding the "best" choice of switches for controlling the diffusion (hence controlling exactly how fast the cloud is mixed). Of course, we could tune parameters until the PSPH result agreed exactly with the other codes here, but given the complicated, non-linear nature of these switches, it is by no means clear that this would be appropriate for any other problem.

In 1D there are no KH or RT instabilities so the blob is not destroyed, this simply becomes a pair of Riemann problems easily solved by all methods. In 2D we see the same qualitative behavior in all cases with the same subtle differences for SPH noted in § 4.3.1.

4.4 Tests with Self-Gravity

Now we consider several tests involving self-gravity and hydrodynamic forces on gas. Recall, the N-body gravity algorithm here is essentially identical to that in GADGET and AREPO, modulo well-tested improvements and optimizations, and this has been tested in a huge variety of situations (see e.g. **?**). We have confirmed these by re-running tests like the collisionless (dark matter) Zeldovich pancake, collisionless spherical collapse and virialization, and cosmological dark matter halo evolution using the public AGORA project initial conditions (see **?**, for details). For our purposes here, therefore, it is not interesting to test the gravity solver in and of itself. However, it is important to test the coupling of hydrodynamics to self-gravity. This is both because complicated and interesting regimes can arise, quite distinct from those in any of the pure hydrodynamic test problems above, and because there are many different choices for how to solve the coupled hydro-gravity equations, some of which can corrupt the hydrodynamics (via e.g. noise from gravity, poor total energy conservation, etc.). It is also important to test that our implementation of a cosmological integration scheme appropriately handles the hydrodynamic quantities.

4.4.1 The Evrard (Spherical Collapse) Test: Gravity-Hydrodynamic Coupling & Energy Conservation

We begin with the simple but very relevant test problem from **?**, which is commonly used to test SPH codes (**????**), but until recently had not generally been used for grid methods. On an arbitrarily large (open) domain, we initialize a three-dimensional sphere of gas with adiabatic index $\gamma = 5/3$, mass $M = 1$, radius $R = 1$, and initial density profile $\rho(r) = M/(2\pi R^2\, r) = 1/(2\pi\, r)$ for $r < R$ and $\rho = 0$ outside the sphere. The gas is initially at rest and has thermal energy per unit mass $u = 0.05$ (much less than the gravitational binding energy). When the simulation begins, the gas free-falls towards $r = 0$ under self-gravity, until a strong shock occurs and the inner regions "bounce" back, sending the shock outwards through the infalling outer regions of the sphere. Eventually, the shock propagates across the whole sphere and the system settles into a hydrostatic virial equilibrium. The test is useful because it is typical of gravitational collapse of structures, and because it involves the conversion of gravitational energy to kinetic energy then to thermal energy; so it is quite sensitive to the total energy conservation of the

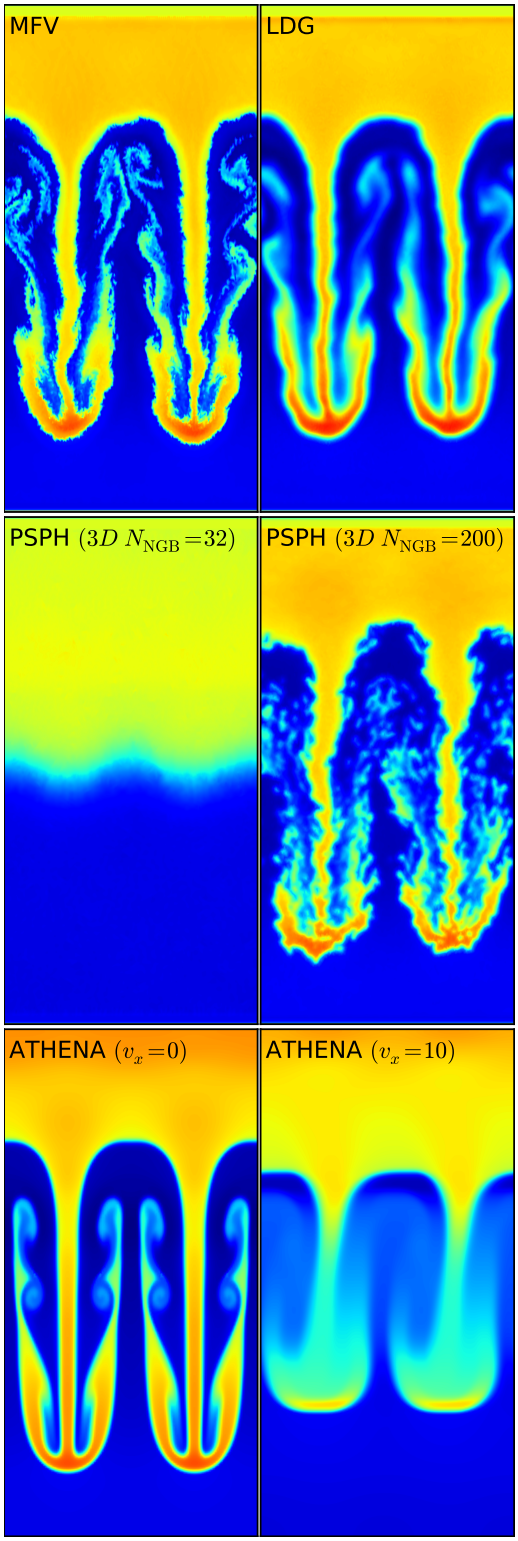

Figure 19. RT instability as Fig. 18, at fixed time $t = 4$, in medium-resolution (128x256) runs, with different methods. *Top Left:* MFV. As Fig. 18, the evolution and secondary instabilities are sharply resolved, even at lower resolution. *Top Right:* LDG. The evolution is similar to MFV, but the lack of mass fluxes means that contact discontinuities are necessarily more smoothed, so the mixing features are "less sharp" at fixed resolution. *Middle Left:* PSPH, with the same number of neighbors as the LDG and MFV runs (equivalent of 32 in 3D). No instability develops, despite the use of the pressure-based SPH formalism, artificial conduction, and more accurate gradients, because the E0 error swamps the growth of the mode. *Middle Right:* PSPH, with a higher-order kernel and the equivalent of 200 neighbors. This is required to reduce the E0 error sufficiently to see the mode grow. However the non-linear evolution is corrupted by noise in the conduction scheme. *Bottom Left:* Stationary-grid (ATHENA) run, when the fluid has no bulk velocity relative to the grid. This exhibits very sharply-

code (particularly challenging for coupled gravity-hydro methods with adaptive timestepping, as we use here).

Following **?**, we show in Fig. 22 the radial profiles of density, velocity, and entropy at time $t = 0.8$ (after the strong shock has formed but before the whole system is virialized), using a fixed number $\approx 30^3$ resolution elements for the initial sphere in all methods. There is no analytic solution here, but we use as a reference the result of a one-dimensional high-resolution, high-order (PPM) calculation in spherical coordinates; at sufficiently high resolution our LDG and MFV results are indistinguishable from this so it should be close to an exact solution.

In every method, at limited resolution, the shock front is smoothed and leads the exact shock front slightly, but this is expected. All the methods capture the key qualitative features of the problem, but with important differences.

The LDG, MFV, and moving-mesh results are very similar. LDG appears to give a slightly more accurate shock location, and as a result more accurate post-shock density profile (the others are slightly depressed because the shock is moving "too fast"). Both LDG and MFV methods exhibit some post-shock "ringing," which owes to our particular choice of slope-limiter. Moving meshes give the least-noisy result, but slightly larger shock position offset. All capture the full entropy jump, to the same width as the density and velocity jumps. All converge similarly rapidly to the exact solution. For example, we show an MFV run with 128^3 resolution, which is now almost indistinguishable from the exact solution (the same is true with LDG; for the same with moving meshes, see **?**, Fig. 41).

SPH captures the key behaviors, but with much more severe smoothing of the shock. In particular the entropy jump is flattened and spread over nearly ~ 1 dex in radius. Because of the artificial conduction terms and larger kernel size in PSPH, the smoothing effect is even more severe. In particular the artificial conduction leads to an entropy jump which is not only more smoothed, but actually leads the real shock position by a couple of smoothing lengths.

The least-accurate result (in an $L1$-norm sense) within and around the shock is produced by the fixed-grid.[15] This is mostly because at fixed resolution of the ICs, the "effective" resolution in the center of the collapsing region is much worse than the other methods (since the method is non-adaptive). But as we have shown, spherical inflow/outflow across a Cartesian grid also produces significant noise and advection errors aligned with the grid axes. As expected from our tests above, the solution quality with fixed-grids will further degrade if we set the sphere in motion across the grid. In fact comparing an AMR result where the maximum refinement is limited so that the cell number not exceed the particle number of the lagrangian methods by more than a factor of ~ 2, the result is not improved (see e.g. Fig. 12 in **?**, for an example with ENZO).

We note that a 1D or 2D analogue of this problem is straightforward to construct, and produces the same qualitative behavior in all methods.

[15] In this section, because ATHENA does not have a flexible self-gravity solver which can be fairly compared to the other methods we use, we will use as our reference "fixed grid" solutions the published results from AREPO using a fixed, Cartesian grid (i.e. not allowing its mesh to move or deform with the fluid). As shown in **?** these are very similar to those from ATHENA and other grid codes on problems where they can overlap, so do not expect the subtle code differences to be as important as the basic aspects of the method.

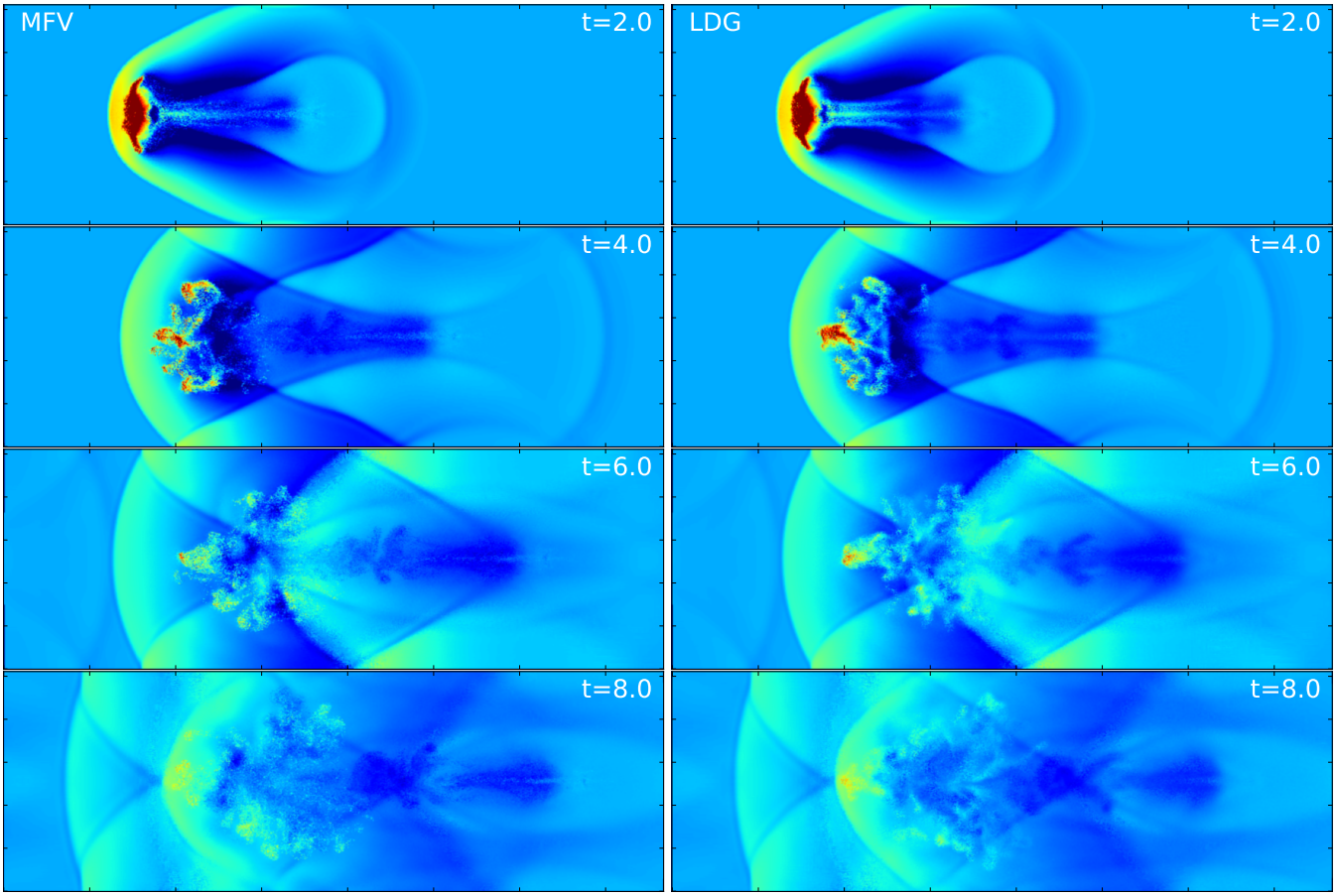

Figure 20. Rayleigh-Taylor instability test (§ 4.3.2). We plot density, from $0.8-2.8$ (black-red), in a two-dimensional simulation. Panels show the evolution of the RT instability using the MFV method at high resolution (512x1024), at different times. The linear growth of the instability is nearly identical in MFV, LDG, moving-mesh, and fixed-grid runs; in all cases it grows and secondary KH instabilities appear along the rising/sinking streams. Note the fine resolution of contact discontinuities and mixing. This run uses our standard number of particle neighbors: for both LDG and MFV runs, the instability develops regardless of the number of neighbors used (we have tested from $\sim 8-64$ in 2D). The breaking of vertical symmetry in the non-linear phase is expected from the problem setup.

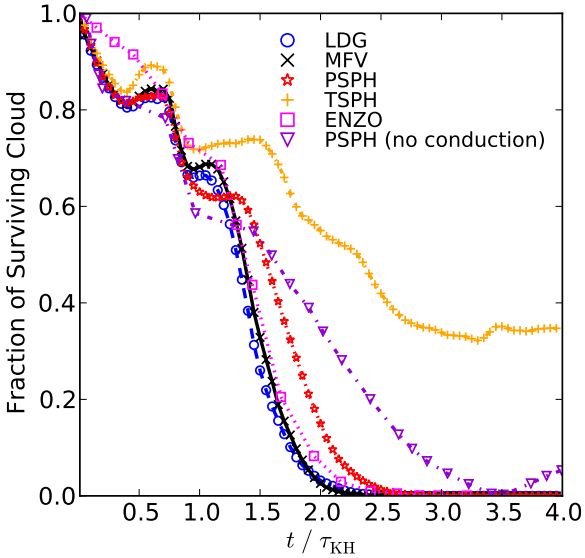

Figure 21. Decay of the blob

4.4.2 *The Zeldovich Pancake: Cosmological Integration, Anisotropic Geometries, & Entropy Conservation*

A standard test for cosmological integration is the "Zeldovich pancake": the evolution of a single Fourier mode density perturbation in an Einstein-de Sitter space. This is a useful test both for scientific reasons (it represents a "single mode" of large-scale structure formation in cosmology) and for testing a code's ability to deal with cosmological integrations, small-amplitude perturbations, extremely high Mach-number flows and shocks, and highly anisotropic cell/particle arrangements. Following **?**: assume initial (unperturbed) fluid elements have uniform density, represent Lagrangian patches, and have position q along the x-axis at redshift $z \rightarrow \infty$ as well as an un-perturbed temperature T_i at some arbitrarily large initial simulation redshift z_i, and $\gamma = 5/3$. The perturbed comoving position x, density, peculiar velocity (also in the

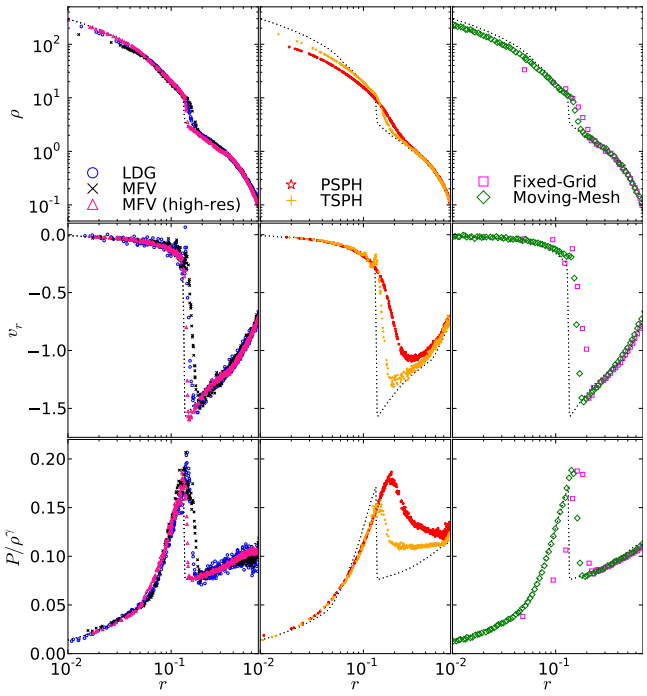

Figure 22. Noh implosion test from Fig. 11; we plot an image of the gas density (from 0 in black to > 64 in red), in a 2D slice through $z = 0$, at $t = 2$. *Top:* LDG: As in the Sedov test, the solution is smooth and shows good spherical symmetry, as it should. *Middle:* TSPH: The solution is spherical on average, but severe noise is again visible (there should be no internal structure here). *Bottom:* Stationary-grid: The carbuncle instability leads to the "hot spots" where the shock is propagating along the coordinate axes.

x-direction), and temperature are then

$$x(q,z) = q - \frac{1+z_c}{1+z}\,\frac{\sin(kq)}{k} \tag{8}$$

$$\rho(q,z) = \frac{\rho_0}{1-\frac{1+z_c}{1+z}\cos(kq)} \tag{9}$$

$$v_{\rm pec}(x,z) = -H_0\,\frac{1+z_c}{\sqrt{1+z}}\,\frac{\sin(kq)}{k} \tag{10}$$

$$T(x,z) = T_i\left[\left(\frac{1+z}{1+z_i}\right)\frac{\rho(x,z)}{\rho_0}\right]^{3}^{2/3} \tag{11}$$

with $k = 2\pi/\lambda$ the wavenumber of the perturbation (wavelength λ), ρ_0 the background (critical) density, H_0 the Hubble constant (today), z_c the redshift of "caustic formation" (i.e. non-linear collapse). This is the exact solution to the linearized perturbation equations. Following **??**, we set $\lambda = 64\,h^{-1}$ Mpc and $z_c = 1$, and start the simulations at an initial redshift $z_i = 100$ (in the linear regime) with $T_i = 100$ K (pressure forces are negligible outside the collapse region). We initialize this in a 3D periodic box of side-length $= \lambda$ (the density and temperature are uniform in the directions perpendicular to the x-axis, and the perpendicular components of the peculiar velocity are zero). This is done because the 3D version of the problem is most challenging, for reasons discussed below. For the particle-based methods, we initialize the particles in a glass rather than a lattice, since this is the "standard" for cosmological simulations; however, this seeds some small noise in the initial density fields.

Fig. 24 shows the density, peculiar x-velocity, and temperature at redshift $z = 0$, as a function of x position, where we use a low-resolution initial condition of just 32^3 particles in the domain (the results are similar, but with decreasing noise and sharper shock capturing, at 64^3 and 128^3). In early phases, $z \gg z_c$ (when pressure forces are negligible), the system simply traces the linear solution given above: this is captured well by all methods. The interesting dynamics occur after the caustic formation at z_c: the caustic collapses and forms a strong shock (factor $\sim 10^{10}$ temperature jump!), which propagates outwards, with a central temperature cavity that has (formally) divergent density at $x = 0$ as the external pressure/temperature vanishes ($T_i \rightarrow 0$). The un-shocked flow follows the extension of the linear solution.

As we saw before, stationary-grid and moving-mesh methods show the least noise in the un-shocked flow. However, because of its non-lagrangian nature, the stationary grid has the poorest resolution inside the shock, and so (at this resolution) it misses all the internal structure in the shocked region (the difference between the central divergence and outward-moving shock, for example), and suppresses the density jump by factors of ~ 100 relative to the particle-based methods.[16] The moving mesh does not suffer from this problem so captures some of the structure and obtains a factor ~ 10 higher density jump, but this is still over-smoothed and suppressed by a factor of ~ 10 relative to the LDG, MFV, and SPH methods.

SPH methods do reasonably well on this problem, avoid the need for an entropy/energy switch, and capture the density peak. As expected, however, the shock jump is spread over multiple smoothing lengths, here about twice the "true" width of the shocked region. There is also more noise, especially in the un-shocked density and temperature fields: initial noise in the density field in this problem is (correctly) amplified as if it were the seeds of cosmological structure. Finally, in TSPH, notice that the velocity solution exhibits some points near $x \sim \pm 5\,h^{-1}$ Mpc which over/under-shoot the correct solution. This is a failure of the artificial viscosity switch (here, the constant, "standard" artificial viscosity of SPH) – the artificial viscosity (even when "always on") is "too weak" to prevent particle interpenetration at these extremely super-sonic Mach numbers (particles "punch through" the shock). In PSPH, the higher-order artificial viscosity switches actually trip a stronger artificial viscosity term when a strong shock is detected, which eliminates this behavior.

The LDG and MFV methods perform very well, with substantially reduced noise (especially in temperature) relative to the SPH solution. Note that if we use a regular lattice to initialize this problem instead of a glass, the noise is almost completely eliminated (as in the moving-mesh and fixed-mesh codes); however, the particle anisotropy in the shock is more severe (discussed below). In both LDG and MFV methods, the shock temperature jump is captured as well as in the moving-mesh code, with its internal structure and the density peak very well-resolved compared to both the moving-mesh and stationary-mesh methods.

Two elements are key for good behavior on this problem. The first is some entropy-energy switch (see § **??**). *Whenever* a conservative Riemann method is used for the hydrodynamics on a problem like this (where the flows are extremely super-sonic, Mach number $\sim 10^5$), very small errors (part in $\sim 10^{10}$) in the momentum solution must (given energy conservation) appear in the tem-

[16] In AMR methods, the outward jumps can be better captured with more refinement, of course, but it requires an effective refinement level of $\sim 512^3 - 1024^3$ (five level-hierarchies or 2^5 refinement in each dimension, increasing the total cell number and CPU cost by a factor of ~ 5000 in the 3D version of this problem) to achieve the same accuracy as the moving-mesh result (see e.g. Fig. 13 in **?**).

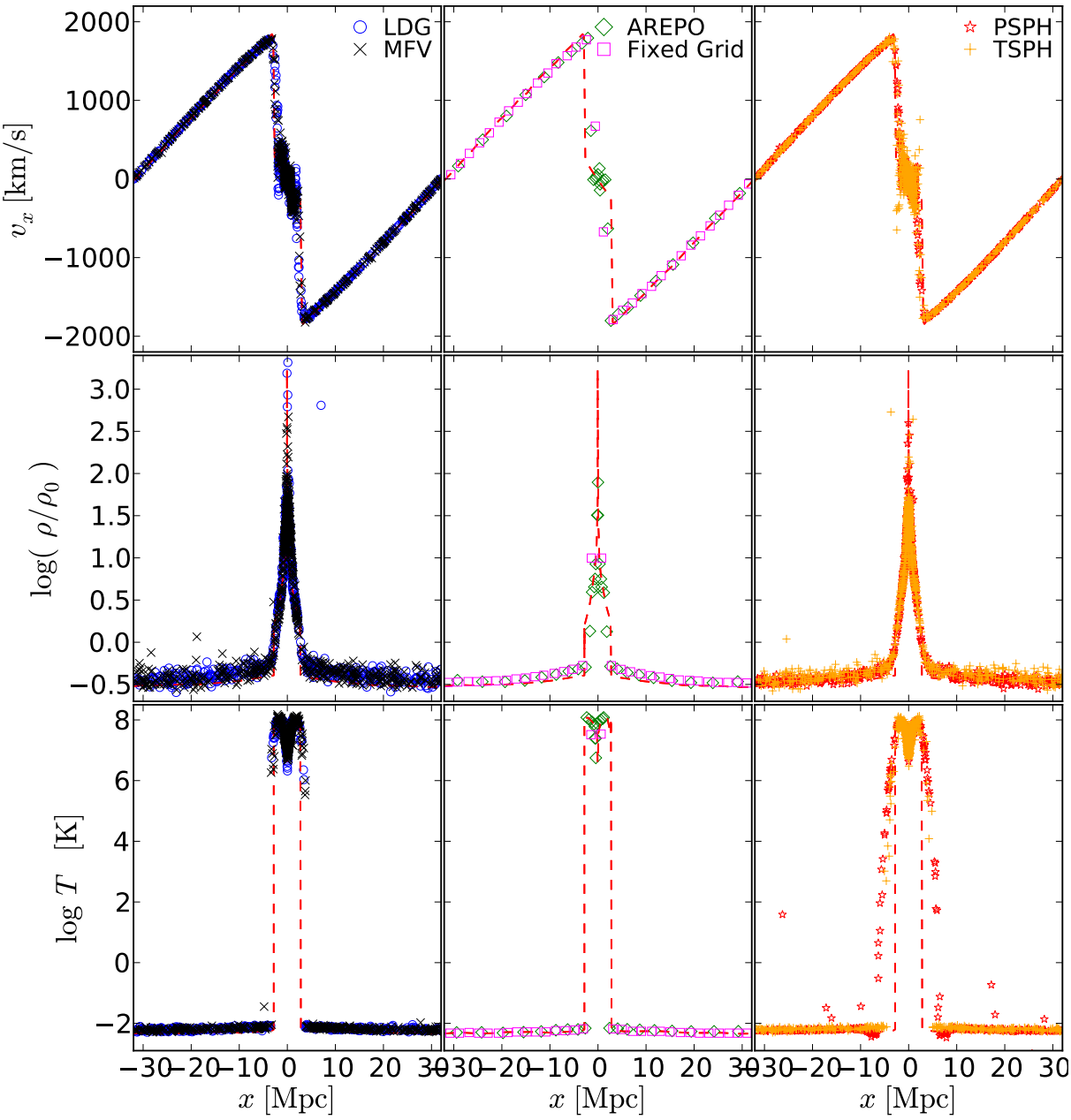

Figure 23. Decay of the blob

perature solution, which can lead to large deviations from the exact solution (although, by definition, these errors appear when the temperature is so low it has no effect on the dynamics, so this does not actually corrupt any other parts of the numerical solution). In stationary-mesh codes, the choice of entropy-energy switch totally controls the accuracy of the solution in the un-shocked regions. We find by systematic experimentation that the LDG and MFV methods are much less sensitive to this source of error compared to moving meshes and especially stationary-mesh codes (because the mass advection "across cells" is zero or reduced); however they are not free of it. Still, this reduced sensitivity allows us to use a much more conservative switch compared to even the choice used for this problem in AREPO (as described in § **??**).

Second, the code must be able to deal with an extremely anisotropic geometry: the fluid is compressed enormously (factor ~ 1000) along the x axis but not the other two axes. In stationary-meshes (including AMR), since the cells are always "regular" (usually cubical), this leads to a practical loss of resolution – obviously non-AMR methods lose resolution when the fluid is compressed, but AMR methods which would try and "refine" near $x \sim 0$ in this problem (i.e. around/within the shock) are forced to refine in the y and z directions simultaneously. So to capture a factor ~ 10 compression in the x-direction, a factor $\sim 10^3$ more cells are required (filling in the "plane")! Practically, this means that these methods always, at fixed CPU cost, under-resolve these compressions in 3D. In a moving-mesh, as the compression becomes more anisotropic, the cell becomes more irregular (less cubical or spherical) in shape, which leads to larger and larger errors in the hydrodynamics and gravity (which assumes a regular cell); this will eventually destroy the solution or crash the code if some "mesh reg-

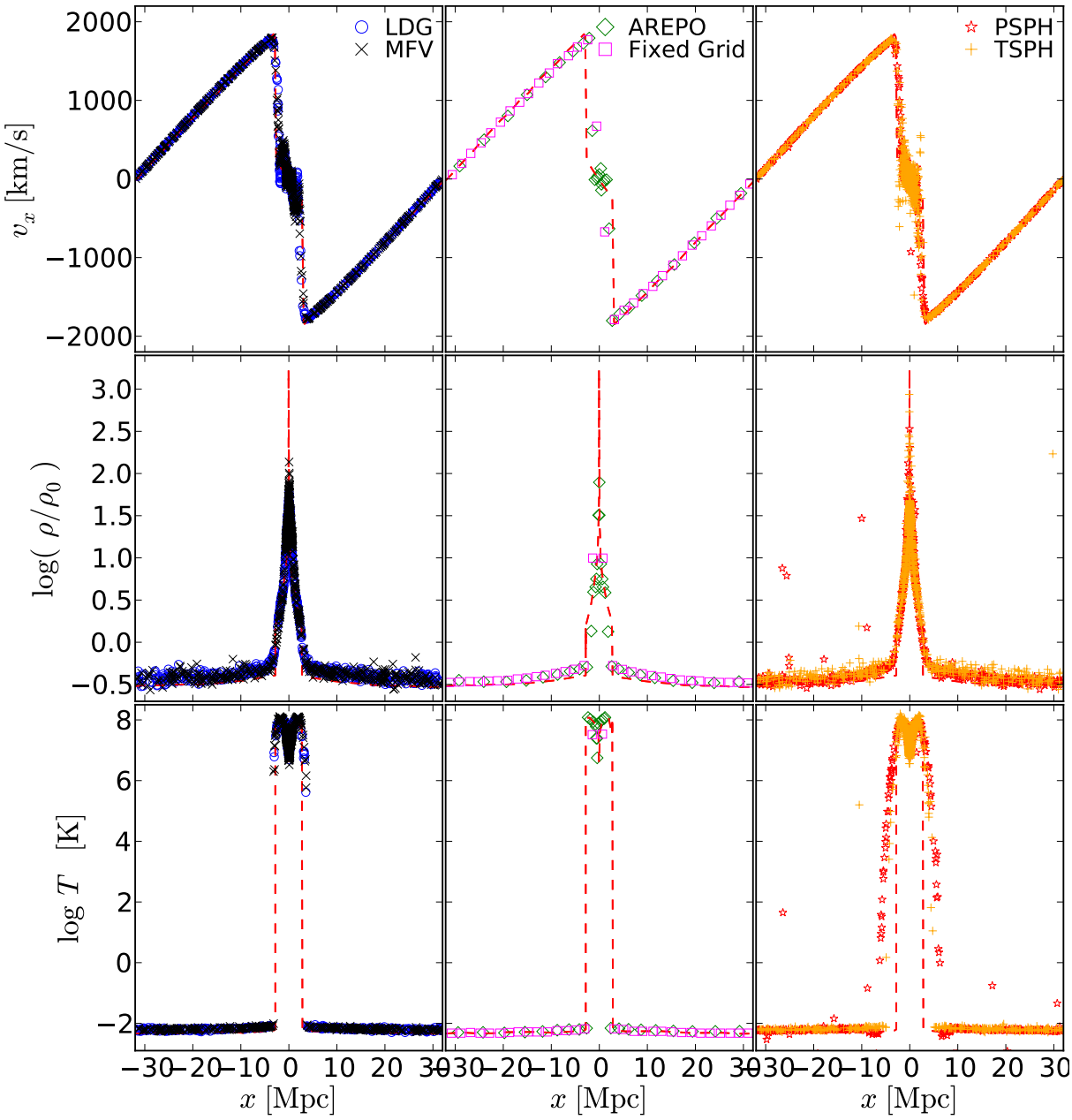

Figure 24. Decay of the blob

ularization" is not used to enforce more-regular cells (making the mesh "stiff"; this is done in AREPO). But the more mesh regularity is enforced, the more it acts like an AMR code and suffers from loss of resolution (and advection errors) – this is why the density peak is still suppressed by a factor of ~ 10 in AREPO compared to the particle-based methods. In particle-based methods, there is a different problem: as the geometry is more compressed in x, the local particle distribution becomes highly anisotropic. In SPH, that increases the zeroth-order errors in the method (hence the larger noise). In the LDF and MFV methods, these errors are eliminated by the matrix-based gradient approach; however, if the particle distribution becomes sufficiently anisotropic, the gradient matrix becomes ill-conditioned. This is especially severe if we begin from a perfect particle lattice, in which case we can end up with the pathological particle distribution where all $N_{\rm NGB}$ neighbors lie exactly alone a line in the x-direction! To handle this, the adaptive checks described in § **??** and **??** are necessary (or else the code will crash); for a glass IC, we find that the code adapts well and ends up finding well-conditioned matrices inside the shock region at $\sim 1.5-2$ times the "default" neighbor number; for the lattice IC, the initial caustic formation is the one case where the code has difficulty finding a well-conditioned matrix and resorts to the method in § **??**. This, however, produces very small differences in the final solution.

4.4.3 *The Santa Barbara Cluster: Cosmological Hydrostatic Equilibrium, Inflow, & Entropy Noise*

We next consider the "Santa Barbara Cluster" from the comparison project in **?**. This is a standard reference test problem for which many codes have been compared. It is a "zoom-in" simula-

tion in which a low-resolution cosmological background contains a higher-resolution Lagrangian region which will collapse to form an object of interest (and the region around it) by $z = 0$; here chosen so the object represents a rich galaxy cluster in an Einstein-de Sitter Universe. The details of the cluster ICs are described there; briefly, a periodic box of side-length $64\,h^{-1}\,$Mpc is initialized at redshift $z = 49$ ($a = 1/(1+z) = 0.02$), in a flat Universe with dark matter density $\Omega_{\rm DM} = 0.9$, baryonic $\Omega_{\rm b} = 0.1$, Hubble constant $H_0 = 100\,h\,{\rm km\,s^{-1}\,Mpc^{-1}}$ with $h = 0.5$, and negligible initial gas temperature $T = 100\,$K. The gas is non-radiative (ideal) with $\gamma = 5/3$.

We show the results of the cluster test at $z = 0$, at two resolutions: specifically we show the profile of dark matter and gas density, gas temperature, pressure, and entropy, as a function of radius. The largest differences between methods reflect what **?** originally identified as the main differences between SPH and stationary AMR/grid-based methods: namely, that stationary grid methods tended to predict systematically higher central entropy "cores" as compared to SPH. This difference persists even in modern, more accurate and higher-resolution versions of the test (see e.g. **?**). The difference is discussed at length in **?**, § 9.3 therein;

temperature and gas density profiles in the inner parts of the cluster. This also has an impact on cluster cooling rates if radiative cooling is allowed, and on important observables such as the emitted X-ray luminosity. In SPH, entropy is accurately conserved (Springel & Hernquist, 2002; Ascasibar et al., 2003), but it could be arti- ficially low due to the absence of entropy production through mixing and to SPHÕs tendency to spuriously suppress fluid instabilities. On the other hand, the Eulerian codes may overestimate the central entropy as a result of numerical diffusivity and overmixing. Also, they are more prone to suffer from heating caused by the noisy gravitational field produced by the collisionless matter. Recently, the idea that the difference may ultimately arise from differences in the treatment of mixing has found some support in numerical ex- periments (Mitchell et al., 2009). Presently, it remains how- ever unclear what the correct entropy profile for the Santa Barbara profile really is, even though this is an important question for numerical cosmology. Note that due to the ab- sence of radiative cooling in this problem, the Santa Barbara cluster represents comparatively clean and ÔeasyÕ physics. If even this case cannot be calculated fully reliably, it is clear that the more demanding simulations that also account for radiative cooling are fraught with numerical uncertainties.

In Figure 44, we first show the evolution of the mean mass-weighted temperature of the whole simulation box, from the starting redshift to the present time. Initially, no structures have formed yet, so that the mean mass-weighted temperature should decline as T ? a?2 for a while. Eventu- ally, the thermal energy content in the shock-heated gas of the first forming cosmic structures starts to dominate and the mean temperature begins to rise rapidly. This general evolution is reflected in the four simulation results depicted in Figure 44, albeit with interesting differences in detail. The green dashed line shows the result of the moving-mesh ap- proach when the ordinary total energy approach is applied. The red line gives the result when the energy-entropy formal- ism is used with a Mach number threshold Mthresh = 1.1, while the solid blue line uses our alternative switch for decid- ing whether the entropy should be kept instead of updating it with the total energy equation. In the latter case, the en- tropy is used if the thermal energy is at most a small fraction ?S = 0.05 of the local kinetic energy. This proves effective to yield the expected adiabatic decline of the mean temper- ature at high redshift. On the other hand, the Mach-number based switch does not make a difference in this regime, as the shock waves responsible for this high-z heating are typi- cally quite strong. However, it can still effectively act against noise-induced heating in virialized structures at lower red- shift. For comparison, the dashed light blue line gives an SPH result obtained with GADGET-2 at the same resolu- tion. It yields a high-redshift evolution very similar to the moving-mesh code when the entropy scheme is used for the cold gas, but at low redshifts its gas ends up being notice- ably colder on average. A substantial part of this difference in the final temperature is probably simply caused by the lower effective resolution of SPH, which tends to reduce the heating through shocks. Higher resolution SPH calculations yield a mean temperature that is 5-8Radial profiles of mean gas density, gas entropy, gas temperature and dark matter density of the final Santa Bar- bara cluster are given in Figure 45. We show results for the different numerical resolutions of 323 , 643 , and 1283 with solid circles, in different colours as labelled. All these sim- ulations use the entropy-energy formalism with a threshold Mach number Mthresh = 1.1 in order to suppress spuri- ous heating from the noise in the gravitational field induced by the dark matter. The thermodynamic profiles converge reasonably well, but not nearly as well as the dark matter density. Interestingly, the central cluster entropy is actually quite close to the SPH result that is shown for comparison, but the innermost entropy profile shows a shallower slope that produces a temperature profile that keeps slowly rising to the very centre. If the total energy equation is applied

throughout the calculation in the 1283 run, we obtain the result shown with hollow circles. It produces much higher core entropy and central gas temperature, as well as a low- ered central gas density, when compared with our default mesh-based calculation. We think these results clearly show that the origin of the discrepancy found first in Frenk et al. (1999) between the central cluster entropy in SPH and AMR codes is caused by dissipation in extremely weak shocks and the production of mixing entropy in effectively smooth parts of the flow. Part of this dissipation is clearly artificial and caused by gravitational N-body noise, which has much more

drastic consequence in mesh-based calculations than in SPH. It therefore appears clear that mesh-based results that use the energy equation alone will overestimate the central en- tropy. Unfortunately, it is less clear how much suppression of dissipation is warranted, and where hence the true en- tropy level ultimately lies. This will be investigated further in future work. We note that the dark matter density profiles found with AREPO converge very well, and are consistent with the ones found with GADGET-2. Also, we have found that at high resolution (643 and 1283) it makes essentially no

difference to the results whether the ÔstandardÕ approach to treat the gravitational work term is employed, or our alternative scheme based on the actual mass fluxes at the surfaces of cells. Only at the low resolution of 323 we have found that the cell-centred approach gives slightly higher central cluster entropy and temperature.

4.4.4 Isolated Galaxy Disks: No ISM Physics

iso 1

Obviously, there is no direct 1D or 2D analogue for this test.

4.4.5 Isolated Galaxy Disks: With ISM Physics

iso 2

5 PERFORMANCE

No methods paper would be complete without some discussion of the speed/computational cost of the method. This is always difficult

to quantify, however, since even comparing the identical code with different hydro solvers (as we implement here), the non-linear solutions of the test problems will become different so it is not obvious that we care comparing the "same" test anymore (for example, if one method resolves more small-scale structure or higher densities, it will necessarily lead to smaller timesteps, even if it is "faster" for identical benchmarks). Nevertheless our suite of simulations gives us some insight.

First, we compare the LDG and MFV methods to SPH, since these are all run within the same code. Note that while "traditional" SPH is computationally very simple, "modern" SPH requires higher-order switches which introduce comparable complexity to our method (in complicated pure-hydro tests such as the "blob" test, this increases the runtime by $\sim 60\%$ from TSPH to PSPH). At fixed resolution and neighbor number, the hydro loop of SPH is faster because a Reimann solver is not needed. However the performance difference is small: even in a pure hydro problem (ignoring gravity and other code costs), the addition to the hydro adds a fixed multiplier of a factor of a couple. And in fact, because of the timestep requirements which artificial viscosity schemes impose on SPH (and the elimination of various operations needed for the artificial diffusion terms), we are actually able to take larger timesteps in our method. So we actually find that running many of our pure hydro problems with the same particle and neighbor number is slightly ($\sim 10\%$) *faster* with the new methods! For example, compare the speeds of our 3D KH problem, normalized to the cpu time to run to the same point with the TSPH method: the runtimes for TSPH, PSPH ($N_{\rm NGB}=32$), PSPH ($N_{\rm NGB}=200$), LDG, and MFV are 1.0, 1.4, 2.5, 0.91, 1.5. And in many problems, where gravity is the dominant cost, the differences are small – e.g. in the isolated disk problem, with the Springel & Hernquist equation of state, the respective runtimes for TSPH, PSPH ($N_{\rm NGB}=128$), LDG, and MFV are 1.0, 1.5, 1.0, 1.2. Moreover, we should really compare performance at fixed *accuracy*. This requires at least an order-of-magnitude more neighbors in SPH than in the new method; that in turn means to compare at fixed mass resolution and accuracy means the hydro loop is more expensive by $\sim N_{\rm NGB}^{3/2}$. So it quickly becomes untenable to run even test problems at this accuracy in SPH.

Comparing our code to AREPO, in its most-optimized format as of the writing of this paper, shows that both the LDG and MFV methods are somewhat faster on all the test problems we have directly compared. The gravity solvers are nearly identical and a Reimann solver is required in both; the typical number of neighbor cells (for a second-order solver) in moving-meshes is usually $\sim 13-18$, smaller than even 32 neighbors, but this trades against the cost of constructing and completeness-testing the mesh, which is substantial (though it is not done every timestep). The bigger difference is in memory cost – the memory requirements of the LDG and MFV methods are basically identical to SPH (relatively low); however, to avoid reconstructing the Voronoi mesh "from scratch" every timestep (which would make the method much slower), moving mesh codes like AREPO must save the mesh connectivity (or faces) for each particle/mesh generating point. This places some significant limitations on how well the code can be parallelized before communication costs are large.

Comparing to grid/AMR codes is much more ambiguous, since almost everything "under the hood" in these codes is different from the method here and it is not clear how to make a fair speed comparison (after all, different grid codes on the same test problem, with the same method, differ significantly in speed). Purely regular, fixed-grid codes (e.g. ATHENA) are almost certainly faster on problems where the fluid is stationary, if all else (e.g. gravity, timestep criterion, choice of Reimann solver) is equal and a second-order method is used, since this minimizes the number of neighbors and means a neighbor "search" is unnecessary (the neighbors are always known based on cell position). However, as soon as we run with a higher order spline, a substantial part of this speed advantage is lost. Moreover, to maintain accuracy, grid codes *should* limit the timestep based on the speed of the flow over the cell; for supersonic flows this is far more demanding than the traditional Courant condition. This reduces the timesteps by factors of $\sim 100-1000$ in some of the problems we consider here, compared to the LDG and MFV methods! Such effects are far larger than the naive algorithmic speed difference. The same is true in AMR codes. Moreover, in AMR the number of neighbors is not so different from our methods, and can sometimes be even larger, so even for a stationary flow the LDG and MFV methods can have a speed advantage. Moreover, it is well-known that AMR methods impose a very large memory cost as they refine; whereas the memory cost of the Lagrangian methods is basically fixed in the initial conditions.

In short, for a complicated (and probably unfair comparison) problem like a zoom-in simulation (e.g. the Santa Barbara cluster), we find the LDG and MFV methods run in comparable (perhaps slightly faster) time than TSPH (comparable to the time for GADGET-3 runs), which is itself substantially faster than "modern" SPH and moving mesh methods, which are themselves still faster than the popular AMR methods in e.g. RAMSES, ART, and ENZO. The memory costs are similar for SPH, LDG, and MFV methods, substantially (factor >2) higher for AREPO, and much higher still for the AMR methods.

6 DISCUSSION

We have developed two new, closely related numerical methods for solving the equations of hydrodynamics. The methods are both Lagrangian (move with the fluid flow) and meshless, allowing continuous and automatic adaptive resolution and deformation with the flow, while being simultaneously second-order accurate and manifestly (machine-accurate) conservative of mass, momentum, and energy. We stress that these methods are not a form of SPH (the existence of a kernel function is the only piece of the method in common) – rather, a more accurate description of the methods is

We implement these methods in a new code GIZMO, which couples them to an accurate tree+particle mesh gravity solver, enables adaptive timestepping (while maintaining conservation), and includes cosmological integration, star formation, radiative cooling, and many additional physics (based on GADGET).

We have considered an extensive

6.1 Comparison to SPH

The methods we propose avoid many known problems with SPH methods, and as a result give more accurate results in almost every test we consider. Even in the "modern" SPH,[17] potentially serious issues arise with noise, artificial diffusion, fluid mixing, and sub-sonic flows. While the modern SPH methods have tremendously improved performance in most respects compared to "traditional" SPH, there are still fundamental problems related to the zeroth-order errors in the method. Without sacrificing conservation

[17] "Modern" SPH defined as those methods using higher-order kernels, pressure-based formulations of the equations of motion, a fully Lagrangian equation of motion, more accurate integral-based gradient approximations, and higher-order dissipation terms for artificial viscosity & conduction.

and numerical stability (which leads to disastrously large errors that quickly wipe out any real solutions), these errors can only be "beaten down" in SPH by increasing the order of the kernel and number of neighbors. So convergence is very slow. And this entails a loss of resolution (typical mass resolution going as $\sim N_{\rm NGB}^{1/2}$, depending on the choice of kernels).

Our methods eliminate the need for artificial dissipation terms and so – despite the use of a Reimann solver – are substantially *less* diffusive than even the highest-order modern SPH switches/schemes. They conserve angular momentum much more accurately owing to reduced numerical viscosity, allowing gas to be followed in hydrodynamic vortices or gravitational orbits for order-of-magnitude longer timescales. They allow sharper capturing of shocks and discontinuities (to within < 1 kernel length, instead of $\sim 2-3$). They substantially reduce the "noise" in the method and so can reliably extend to order-of-magnitude smaller Mach numbers. The treatment of fluid instabilities and mixing in the new methods is accurate and robust without requiring any special modifications or artificial diffusion terms. And the new methods eliminate the zeroth and first-order errors inherent to SPH, *while remaining fully conservative*. This means, most importantly, the methods converge *at fixed neighbor number*, as a consistent second-order method should. We are able to obtain far higher accuracy with ~ 32 neighbors than SPH with ~ 400 neighbors, on most problems we consider. As noted in § **??**, even at fixed neighbor number and resolution there is little significant performance difference between SPH and our new methods, so this means the cost at fixed accuracy is much lower in the new methods.

SPH may still have some advantages in very specific contexts. It naturally handles extremely high Mach number "cold" flows such as those in the Zeldovich problem without the need for an explicit switch to reduce noise from a Reimann solver. It is computationally an incredibly simple method. It trivially handles free surfaces with identically zero diffusion into the vacuum. And switching between fluid and particle dynamics is especially simple. And of course, there are many problems where the accuracy of the solution is not limited by convergence or formal numerical integration accuracy, but by *physics* missing owing either to their complexity or the resolution required to include them.

6.2 Comparison to AMR

Our new methods also avoid many disadvantages of stationary (non-moving) grid methods. In grid methods advection errors are large, the errors are not Galilean-invariant (solutions degrade when the fluid moves), angular momentum is not conserved, spurious "grid alignment" and "carbuncle" instabilities appear, and coupling to N-body gravity solvers is ad hoc (introducing new errors and spurious "grid heating").

By moving with the flow, our method minimizes the advection errors that plague grid methods. This leads to sharper and more accurate capturing of contact discontinuities and shocks. It also leads to dramatically reduced diffusion in any problems involving non-grid aligned motion. The new methods are Lagrangian and Galilean-invariant so can robustly follow motion of fluid with an arbitrary "boost"; this is especially important for multi-phase fluids, where, for example, advection errors in grid methods will rapidly diffuse away a self-gravitating cloud or structure moving relative to the grid. As we and **?** show, this is also important for fluid mixing instabilities: the non-Galilean invariance of errors in grid methods artificially slows down and eventually wipes out the growth of Kelvin-Helmholtz and Rayleigh-Taylor instabilities if the fluid is moving at sufficient bulk velocities (at finite resolution). There is also no "grid alignment" effect; the carbuncle instability does not exist, disks are not forcibly torqued into alignment with a coordinate axis, and shocks do not preferentially propagate along the grid.

Related to this, our method exhibits excellent angular momentum conservation, and can follow gas in gravitational orbits for hundreds of orbits. In grid codes, gas in a rotating disk loses angular momentum and the orbits break down completely in just $\sim 1-2$ orbits, even with $> 10^7$ resolution elements in the disk.[18]

The resolution in our new methods is automatically and continuously adaptive, so provides enhanced resolution where desired, without needing to introduce an "ad hoc" refinement scheme (which may or may not correctly capture the desired behavior). Moreover, it is well-known that low-order errors appear at the (necessarily discontinuous) refinement boundaries in AMR, which break the formal higher-order accuracy of the method; since the adaptivity here is continuous and built into our derivation, these do not appear.

That said, there of course will be contexts where grid codes are particularly useful. It remains to be seen whether the magneto-hydrodynamic treatment in our new method will be competitive with grid codes (this will be the subject of a paper in preparation). Grid codes, especially fixed (non-adaptive, non-moving) regular (locally orthogonal) meshes minimize certain forms of numerical noise and symmetry-breaking compared to any other methods we consider – in highly sub-sonic turbulence (Mach numbers $\sim 0.001-0.01$), for example, this can be quite important. And such simple grids allow for trivially well-optimized parallelization schemes (in the absence of any long-range forces). AMR methods share some, but not all, of these advantages. However, in an AMR scheme, one advantage is that refinement can be based on any quantity, in principle, rather than just following mass/density (the usual choice); this means that, unlike our method (unless a special particle-splitting scheme is adopted), AMR methods can be particularly useful when high resolution is desired in extremely low-density regions of a problem (e.g. around the reverse shock inside an explosion).

6.3 Comparison to Moving-Mesh Methods

Comparing our new methods to moving mesh approaches, the differences are much more subtle, and more work will be needed to determine the real advantages and disadvantages of each approach (as with any new numerical method). In every test, the methods ap-

[18] Of course, all of these errors in grid codes (and SPH codes) are resolution-dependent; the methods do formally converge, so they can be "beaten down" by increasing resolution. However, for any practical problem the resolution cannot be infinite so we care about accuracy at fixed resolution, where all the advantages above are plain. Moreover, for many problems, the convergence is slow, so formal convergence with some methods may be impossible. For example, it is well-known that in grid codes, the angular momentum converges slowly: even at $\sim 512^3$ resolution, a circular gas disk will be strongly torqued to align with one of the coordinate axes, and it will experience strong angular momentum loss, within $\lesssim 2-3$ orbits (see **?**). This is already comparable to the best-ever resolution of galaxy formation simulations of a single galaxy! To evolve a disk to $\sim 30-300$ orbits, based on the expected code scalings, would require something like $\sim 10,000^3 - 100,000^3$ ($10^{12}-10^{15}$) resolution elements, far out of reach even for exascale computing. It is also possible to reduce errors by choosing grids with "specially designed" geometries for the particular problem, but this cannot be generalized.

pear at least competitive with one another. However there are some differences already evident in our comparisons with AREPO.[19]

From the Gresho test, we see that the exact volume partition in moving meshes reduces the "remapping noise" from irregular particle motion in strong shear flows, and hence allows more accurate, smoother tracing of sub-sonic, pressure-dominated rotation (manifest in e.g. subsonic turbulence, with Mach numbers < 0.01).[20]

On the other hand, the angular momentum conservation in our method – particularly for gas in gravitational orbits (e.g. disks) – appears dramatically superior to that in moving-mesh approaches. Some of this owes to a tradeoff with exactly the errors above: the implicit "re-mapping" noise in the LDG and MFV methods arises because we map to spherical kernel functions partitioning the volume. This means angular momentum can be exactly defined and conserved. In a moving mesh, any irregular (non-symmetric) mesh shape means that the total cell angular momentum cannot even be defined (let alone conserved) at higher than second-order quadrature & integration accuracy.

6.4 Areas for Improvement & Future Work

This is a first study of new methods, and as such there is certainly considerable room for improvement.

For the sake of consistency (and simplicity), in this paper we did not systematically vary things like our slope-limiting procedure, approximate Reimann solver, kernel definition, and timestepping scheme. We have undertaken a limited exploration of these and found (not surprisingly) that for some problems, some choices give slightly better or worse results (although they do not change our qualitative conclusions). However a more thorough study could determine a more "optimal" set of choices, especially for cases where the problem structure is known ahead of time.[21]

It is also possible to generalize our method to higher order (as in PPM or WENO schemes), using the appropriate matrix-based least-squares gradient estimator. This is useful both if second derivatives are directly needed (for e.g. conduction), and to make the method itself more accurate (albeit at additional CPU cost). **?**, for example, show how to generalize this to third-order accurate reconstruction at particle faces, which together with a higher-order prediction and reconstruction leads to a PPM-like third-order method.[22] Based on their and our limited experiments, this produces a much smaller improvement than in grid codes (mainly because our advection errors are *already* much smaller than those in arbitrarily high-order grid codes, which is usually the error that motivates higher-order schemes), but it could be useful for some applications.

There is no reason why this method cannot be extended for magnetohydrodynamics (MHD), radiation-hydrodynamics (RHD) and relativistic hydrodynamics, as in many SPH-based and grid-based codes. **?** show one implementation of MHD in an MFV scheme, which we have implemented as well in our code. A systematic comparison of these new methods, SPH-MHD, and grid-MHD methods will be the subject of subsequent work (in preparation). We have only just begun to experiment with radiation-hydro schemes, but this is exciting for many problems of interest. And Lagrangian codes are naturally especially well-suited for relativistic hydrodynamics (many such SPH schemes already exist, and **?** have developed a moving-mesh implementation). And of course many additional examples of fluid physics (e.g. multi-fluid flows, aerodynamic grain-gas coupling, non-ideal MHD, conduction, complicated equations of state, cooling, chemical or nuclear reaction networks) which do not inherently depend on the hydro scheme can be implemented.

ACKNOWLEDGMENTS

We thank Dusan Keres and Lars Hernquist for helpful discussions, and Oscar Agertz, Andrey Kravtsov, Robert Feldmann, and Nick Gnedin for contributions motivating this work. We also thank Justin Read, Volker Springel, Eliot Quataert, Claudio Dalla Vecchia, and the anonymous referee for useful comments and suggestions, and thank Andreas Bauer for sharing the implementation of the turbulent driving routine. Support for PFH was provided by NASA through Einstein Postdoctoral Fellowship Award Number PF1-120083 issued by the Chandra X-ray Observatory Center, which is operated by the Smithsonian Astrophysical Observatory for and on behalf of the NASA under contract NAS8-03060.

APPENDIX A: ON THE SLOPE-LIMITING PROCEDURE FOR UNSTRUCTURED, MESHLESS REIMANN PROBLEMS

All high-order methods (grid or meshless) require a reconstruction of fluid quantities at some interface or quadrature points; in smooth flows this is straightforward. However, at discontinuities or higher-order divergences, numerical stability (the preservation of the upwind character of solutions) requires some slope or flux-limiting

[19] We caution that at least some of the subtle differences we see are not fundamental to the methods, but the result of secondary choices peculiar to each code. For example, we see that shock positions seem to be slightly offset in AREPO in many tests (Noh, Sedov, interacting blastwaves), relative to the analytic solution. We suspect this owes to either a slightly too-aggressive adaptive timestepping scheme (where neighbors differ by large values in timestep) or a too-aggressive application of the entropy-energy switch (where entropy evolution is used for flows where the entropy solution from the normal Reimann problem is non-spurious), since we find both of these effects can reproduce this error in our own LDG and MFV calculations. The latter effect has been reduced in more recent applications of AREPO (V. Springel, private communication). In some problems, we see reduced post-shock ringing/noise with our new methods, in other tests AREPO exhibits smaller "bumps" at rarefaction fronts and shocks; however these differences are much more sensitive to the slope-limiting procedure used for the reconstruction of the Reimann problem than to the basic method itself, and should not be taken as representative.

[20] Since our methods manifestly conserve the conserved variables (e.g. particle energy) but numerically only reconstruct the particle volume partition to second-order quadrature accuracy, they introduce this "remapping noise" in the volumes from irregular particle motion (in e.g. strong shear flows) which leads to small noise in the volumetric (density and pressure) fields. This can compete with the physical velocities we desire to follow when the flows are highly sub-sonic and the resolution is poor.

[21] The pros and cons of different slope-limiters, time integration (and timestep-limiting) schemes, and Riemann solvers will likely apply to both these new methods and moving-mesh, as well as fixed-mesh results (and we have borrowed heavily from insights based on fixed-mesh methods). For the kernel, we have chosen a simple, commonly-used kernel from the SPH literature; however, since the kernel function used here as a very different meaning and function from that in SPH, it is freed from many of the restrictions of SPH kernels (it does not, for example, need to be Gaussian-like). So many studies based on SPH kernels should be revisited, with a more appropriate literature being that on kernel estimation of least-squares field gradients.

[22] Note that to make the method completely third-order at all levels, the equation of motion ("effective" face terms in § **??**) need to be re-derived using the same formalism but the higher-order gradient matrix, but this is not difficult.

procedure be applied. Otherwise new extrema are introduced by over or under-shooting, and these create numerical instabilities.

The most common approach, and the standard in most fixed-grid, AMR, and moving-mesh codes using the MUSCL scheme, is to introduce a slope limiter of the gradients, which ensures that the linearly reconstructed quantities at faces or quadrature points does not exceed the extrema among the interacting neighbor cells/particles (see e.g. **?**). In performing the face reconstruction of some arbitrary quantity ϕ^i for particle i, we replace the "true" (matrix-evaluated) gradient $\nabla\phi^i_{\rm true}$ with an effective (slope-limited) gradient $\nabla\phi^i_{\rm lim}$:

$$\nabla\phi^i_{\rm lim} = \alpha_i\,\nabla\phi^i_{\rm true} \tag{A1}$$

where

$$\alpha_i \equiv {\rm MIN}\Big[1,\,\beta_i\,{\rm MIN}\Big(\frac{\phi^{\rm max}_{ij\,\rm ngb}-\phi_i}{\phi^{\rm max}_{ij,\rm mid}-\phi_i},\;\frac{\phi_i-\phi^{\rm min}_{ij,\rm ngb}}{\phi_i-\phi^{\rm min}_{ij,\rm mid}}\Big)\Big] \tag{A2}$$

where $\phi^{\rm max}_{ij,\rm ngb}$ and $\phi^{\rm min}_{ij,\rm ngb}$ are the maximum and minimum values of ϕ_j among all neighbors j of the particle i, and $\phi^{\rm max}_{ij,\rm mid}$, $\phi^{\rm min}_{ij,\rm mid}$ are the maximum and minimum values (over all pairs ij of the j neighbors of i) of ϕ re-constructed on the "i side" of the interface between particles i and j (i.e. $\phi^{\rm max}_{ij,\rm mid} = {\rm MAX}[\phi_i + \nabla\phi^i_{\rm true}\cdot(\mathbf{x}_{{\rm face},ij}-\mathbf{x}_i)]$).

As noted by **?**, the constant β must have a value $\beta > 0.5$ in order to maintain the second-order accuracy of the scheme (with lower values being more stable, but also more diffusive). AREPO, for example, adopts a scheme very similar to this with $\beta = 1$. Ideally, we would like to use a more "aggressive" (larger and more-accurate) value of β when the gradients are trustworthy and there is good particle/cell order, and a more "stable" (diffusive) value when the gradients are less trustworthy (or there are large fluctuations in quantities within the kernel). Fortunately, as noted in § **??**, we have an indicator of this already, in the condition number of the gradient matrix. After considerable experimentation, we find a very good mix of stability and accuracy on all problems in this paper with the choice

$$\beta_i = {\rm MAX}[\beta_{\rm min},\,\beta_{\rm max}\,{\rm MIN}(1,\,N^{\rm crit}_{\rm cond}/N^i_{\rm cond})] \tag{A3}$$

with $\beta_{\rm min} = 1$, $\beta_{\rm max} = 2$. We find that $\beta_{\rm min} < 1$ does not much improve stability, but does begin to introduce noticeable diffusion of discontinuities, while $\beta_{\rm max} > 2$ does not much improve accuracy and leads can lead to problems with stability in very strong interacting shocks (though for most other problems, $\beta_{\rm max} = 4$ works fine as well with slightly better accuracy).

We actually find that we achieve slightly greater numerical stability, and are able to eliminate one additional loop over the particle neighbors, at the cost of very little added diffusion, if we make this slope limiter slightly more conservative by replacing the quantities $\phi_i - \phi^{\rm min}_{ij,\rm mid}$ and $\phi^{\rm max}_{ij,\rm mid} - \phi_i$ by the value $|\nabla\phi^i_{\rm true}|\cdot|\mathbf{x}_{{\rm face},ij}-\mathbf{x}_i|^{\rm max}$ (where $|\mathbf{x}_{{\rm face},ij}-\mathbf{x}_i|$ is the distance between the particle and face for the pair ij). In other words we replace the explicitly calculated two extrema which happen to be reconstructed based on the particle positions, with the maximum/minimum value that *could be* reconstructed, independent of the geometric arrangement of the particles within the smoothing kernel. This is actually closer to what is intended by this such limiters in grid codes. And $|\mathbf{x}_{{\rm face},ij}-\mathbf{x}_i|^{\rm max}$ can be directly calculated, but given our other definitions is trivially equal to half the maximum size of the local smoothing kernel, $h_i/2$.

We note that this limiter, while useful and sufficient for most problems, is not total variation diminishing (TVD), and cannot strictly guarantee stability even if we use very conservative parameters (e.g. $\beta_i = 0.5$ always). And indeed in some problems with extremely strong shocks (e.g. the Zeldovich pancake) or non-hydrodynamic forces (e.g. galaxy evolution), we see large errors occur (albeit in a small number of particles) if we only include the above limiter. To ensure stability more generally, it is necessary to adopt a pairwise limiter between interacting particles. **???**

There are many choices for this, as in grid codes. For the sake of flexibility, we implement a general form as follows. For the particle pair ij, we begin by reconstructing $\phi_{ij,\rm mid}$ (the re-constructed value on the "i side") as above, using the slope-limited gradients $\nabla\phi^i_{\rm lim}$. We then apply a second pair-wise limiter to this, replacing our initial estimate $\phi^0_{ij,\rm mid}$ with a limited $\phi'_{ij,\rm mid}$ based on the values of ϕ_i and ϕ_j:

$$\begin{aligned}
\phi'_{ij,\rm mid} &= \begin{cases} \phi_i \quad (\phi_i = \phi_j) \\ {\rm MAX}(\phi_-,\,{\rm MIN}[\bar{\phi}_{ij}+\delta_2,\,\phi^0_{ij,\rm mid}]) \quad (\phi_i < \phi_j) \\ {\rm MIN}(\phi_+,\,{\rm MAX}[\bar{\phi}_{ij}-\delta_2,\,\phi^0_{ij,\rm mid}]) \quad (\phi_i > \phi_j) \end{cases} \\
\phi_- &= \begin{cases} \phi_{\rm min}-\delta_1 \;\; ({\rm SIGN}(\phi_{\rm min}-\delta_1) = {\rm SIGN}(\phi_{\rm min})) \\ \dfrac{\phi_{\rm min}}{1+\delta_1/|\phi_{\rm min}|} \;\; ({\rm SIGN}(\phi_{\rm min}-\delta_1) \neq {\rm SIGN}(\phi_{\rm min})) \end{cases} \\
\phi_+ &= \begin{cases} \phi_{\rm max}+\delta_1 \;\; ({\rm SIGN}(\phi_{\rm max}+\delta_1) = {\rm SIGN}(\phi_{\rm max})) \\ \dfrac{\phi_{\rm max}}{1+\delta_1/|\phi_{\rm max}|} \;\; ({\rm SIGN}(\phi_{\rm max}+\delta_1) \neq {\rm SIGN}(\phi_{\rm max})) \end{cases} \\
\bar{\phi}_{ij} &\equiv \phi_i + \frac{|\mathbf{x}_{ij}-\mathbf{x}_i|}{|\mathbf{x}_j-\mathbf{x}_i|}(\phi_j-\phi_i) = (\phi_i+\phi_j)/2 \\
\phi_{\rm min} &\equiv {\rm MIN}(\phi_i,\,\phi_j) \\
\phi_{\rm max} &\equiv {\rm MAX}(\phi_i,\,\phi_j) \\
\delta_1 &\equiv \psi_1\,|\phi_i-\phi_j| \\
\delta_2 &\equiv \psi_2\,|\phi_i-\phi_j|
\end{aligned} \tag{A4}$$

While these expressions are somewhat non-intuitive, they are easy to efficiently evaluate, and ultimately allow considerable freedom of slope-limiters, based on our choice of the free parameters ψ_1 and ψ_2. Many popular slope limiters can be expressed as variations of these parameters: for example, the monotonized central (**?**), minmod and superbee (**?**), **?**, **?**, and **?** limiters all fall in this class. We have experimented with all of these; as always, there is no uniformly "correct" choice, but for the problems here we find a good mix of stability and accuracy adopting $\psi_1 = 1/2$, $\psi_2 = 1/4$. As in our convention for β, these are defined such that smaller values are more conservative/stable but also more diffusive (with $0 \leq \psi_1 \leq 1$ and $0 \leq \psi_2 \leq 1/2$ being the physically reasonable ranges).

If we make the analogy to a regular Cartesian mesh code, we can directly compare this to the standard limiters defined as a function $\phi_{\rm lim}(r) = \phi_{ij\,\rm mid}\,\psi(r)$ of $r = (\phi_i-\phi_{i-1})/(\phi_{i+1}-\phi_i)$, where following **?** we take $\phi_{i-1} = \phi_j$ and ϕ_{i+1} is calculated by projecting the gradient calculated at i in the opposite direction from j by the same distance. Our default choice ($\psi_2 = 1/4$) is then, for $r > 0$, equivalent to $\psi = 2\,r$ for $r < 1/2$ and $\psi = 1$ for $r \geq 1/2$, which is the slope limiter that recovers the "correct" (i-centered least-squares) gradient most accurately while still satisfying the TVD condition. We do confirm that $\psi_2 > 1/4$ leads to unstable behavior, with $\psi_2 > 1/2$ being sufficiently unstable that most Riemann solvers will diverge. Unlike some grid-based slope-limiters, however, we find we do not require $\psi = 0$ for $r < 0$ ($\psi_1 = 0$) to ensure stability, because in *this* regime, the previous limiter based on the max/min values in the kernel provides stability so long as $\psi_1 \leq 1/2$. For $\psi_1 > 0$, however, we include the SIGN terms above to prevent a sign change of extrapolated quantities in the projection (i.e. if both ϕ_i and ϕ_j are positive, the reconstructed quantity can never be negative, and vice versa). The particular form chosen (which is not unique, but is quite

flexible) simply assumes that the derivative measured at i, if it were to lead to an implied sign change, actually describes a power-law declining (instead of linearly declining) function.

Comparing this to the "standard" choice of a single, less-flexible limiter such as the Van Leer, minmod, or superbee limiters, we find it enables a significant improvement in accuracy and reduction in uneccessary diffusion while maintaining stability in every problem considered here. This suggests it might be generally useful for other non-regularly gridded methods, including moving mesh codes (both AREPO and TESS find a pair-wise limiter must be used in addition to the global min/max criterion to ensure stability on more complicated problems, but use more diffusive default choices), and even AMR codes (since the usual way of handling cases where the grid is not perfectly uniform but refined more in one direction is to effectively "down-sample" to a lower-level grid, increasing numerical diffusion).

APPENDIX B: THE CHOICE OF REIMANN SOLVER

We use **???**