

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

Philip F. Hopkins<sup>1</sup>\*

<sup>1</sup>TAPIR, Mailcode 350-17, California Institute of Technology, Pasadena, CA 91125, USA

Submitted to MNRAS, August, 2014

## ABSTRACT

We present and study two new Lagrangian numerical methods for solving the equations of hydrodynamics, in a systematic comparison with moving-mesh, SPH, and stationary (non-moving) grid methods. The new methods are designed to simultaneously capture many advantages of both smoothed-particle hydrodynamics (SPH) and grid-based or adaptive mesh refinement (AMR) schemes. They 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 parallel, second-order version of the method with coupled self-gravity & cosmological integration, in the code GIZMO:<sup>1</sup> this maintains exact mass, energy and momentum conservation; exhibits superior angular momentum conservation compared to all other methods we study; does not require “artificial diffusion” terms; and allows the fluid elements to move with the flow so resolution is automatically adaptive. We consider a large suite of test problems, and find that on all problems the new methods appear competitive with moving-mesh schemes, with some advantages (particularly in angular momentum conservation), at the cost of enhanced noise. The new methods have many advantages vs. SPH: proper convergence, 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, velocity-independence of numerical errors, accurate coupling to  $N$ -body gravity solvers, and good angular momentum conservation and elimination of “grid alignment” effects. We can, for example, follow hundreds of orbits of gaseous disks, while AMR and SPH methods break down in a few orbits. However, non-adaptive fixed meshes exhibit the lowest levels of “grid noise” among all methods we consider. All of these differences between methods are important for a wide range of astrophysical problems.

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

## 1 INTRODUCTION: THE CHALLENGE OF EXISTING NUMERICAL METHODS

Numerical hydrodynamics is an essential tool of modern astrophysics – almost all interesting problems eventually require simulations, because the complexity, diversity of physics, and dynamic range of the problems prohibit exact analytic solutions. But these aspects also pose challenges for the numerical methods used to solve the hydrodynamic equations. A variety of different numerical methods are used in the field, but to date, most hydrodynamic simulations in astrophysics (with some interesting exceptions) are based ultimately on one of two popular methods: smoothed-particle hydrodynamics (SPH; Lucy 1977; Gingold & Monaghan 1977), or stationary-grids, which can be either “fixed mesh” codes where a time-invariant mesh covers the domain (e.g. Stone & Norman 1992), or “adaptive mesh refinement” (AMR) where the meshes are static and stationary except when new sub-cells are created or destroyed within parent cells (Berger & Colella 1989).

These methods, as well as other more exotic schemes (e.g. Xu 1997; Zhang et al. 1997), have advantages and disadvantages. Unfortunately, even on simple test problems involving ideal fluid dynamics, they often give conflicting results. This severely limits their predictive power: in many comparisons of interesting problems, it is unclear whether the differences seen in various runs owe to physics that is modeled, or to purely numerical effects (for an example, see e.g. the comparison of cosmological galaxy formation in the *Aquila* project; Scannapieco et al. 2012). Unfortunately, both

SPH and AMR have fundamental problems which make them inaccurate for certain problems – because of this, the “correct” answer is often unknown in these comparisons.

In Table 1, we attempt a cursory summary of some of the methods being used in astrophysics today, making note of some of the strengths and weaknesses of each.<sup>1</sup> Below, we describe these in more detail.

### 1.1 Smoothed-Particle Hydrodynamics (SPH)

In SPH, quasi-Lagrangian discretized mass elements are followed – the conserved quantities (mass, momentum, energy) are discretized into particles (like an  $N$ -body code), and a kernel function is used to “smooth” their volumetric distributions to determine equations of motion. SPH is numerically stable, Lagrangian (follows the fluid), automatically provides continuous adaptive resolution, has truncation errors which are independent of the fluid velocity, couples exactly to  $N$ -body gravity schemes, exactly solves the particle continuity equation (eliminates certain advection errors), and the equations of motion can be exactly derived from the particle Lagrangian (Springel & Hernquist 2002) giving it excellent conservation prop-

<sup>1</sup> A public version of the GIZMO code, together with movies and additional figures, an extensive user’s guide, and the files and instructions needed to run the test problems in this paper, are all available at <http://www.tapir.caltech.edu/~phopkins/Site/GIZMO.html>

\* E-mail: phopkins@caltech.edu

erties.<sup>2</sup> This has led to widespread application of SPH in many fields (for reviews see Rosswog 2009; Springel 2010; Price 2012b).

However, it is well-known that “traditional” SPH algorithms have a number of problems. They suppress certain fluid mixing instabilities (e.g. Kelvin-Helmholtz instabilities; Morris 1996; Diltz 1999; Ritchie & Thomas 2001; Marri & White 2003; Okamoto et al. 2003; Agertz et al. 2007), corrupt or over-damp the behavior of sub-sonic (pressure-dominated) turbulence and waves (Kitsonas et al. 2009; Price & Federrath 2010; Bauer & Springel 2012; Sijacki et al. 2012), produce orders-of-magnitude higher numerical viscosity in flows which should be inviscid (leading to artificial angular momentum transfer and breakup of disks; see e.g. Cullen & Dehnen 2010), over-smooth shocks and discontinuities (over several SPH smoothing-lengths), are far noisier in following continuous fields compared to grid-based codes, and numerically converge very slowly (if at all).

The sources of these errors are known, however, and heroic efforts have been made to reduce them in “modern” SPH. First, the SPH equations of motion are inherently inviscid, so require some artificial viscosity to capture shocks and make the method stable; this generally leads to excessive diffusion (eliminating one of SPH’s main advantages). One improvement is to simply insert a Riemann solver between particles (so-called “Godunov SPH”; see Inutsuka 2002; Cha & Whitworth 2003), but this is not stable on many problems; another improvement is to use higher-order switches (based on the velocity gradients and their time derivatives) for the diffusion terms (Cullen & Dehnen 2010; Read & Hayfield 2012). Second, a significant part of SPH’s suppression of fluid mixing comes from a “surface-tension”-like error at contact discontinuities and free surfaces, which can be eliminated by kernel-smoothing all quantities (e.g. pressure), not just density, in the equations of motion (Ritchie & Thomas 2001; Saitoh & Makino 2013; Hopkins 2013). Recently, it has also been realized that artificial diffusion terms should be added for other quantities such as entropy (Price 2008; Wadsley et al. 2008); and these further suppress errors at discontinuities by “smearing” them. Third, and perhaps most fundamental, SPH suffers from low-order errors, in particular the so-called “E0” zeroth-order error. It is straightforward to show that the discretized SPH equations cannot correctly reproduce even a constant (zeroth-order) field, unless the particles obey *exactly* certain very specific geometric arrangements. There are similar first-order errors as well. This produces excessive noise, often swamping real low-amplitude effects (especially in sub-sonic flows). Various “corrected” SPH methods have been proposed which eliminate some of these errors in the equation of motion (e.g. Morris 1996; Abel 2011; García-Senz et al. 2012); however, thus far *all* such methods require numerically unstable violations of energy and momentum conservation, leading to exponentially growing errors in realistic problems (see e.g. Price 2012b). Adding terms to force them to be conservative re-instates the original problem (Rosswog 2014). They also do not eliminate all the low-order errors. The only way to decrease all such errors is to increase the number of neighbors in the SPH kernel; using higher-order kernels with several hundred neighbors,

<sup>2</sup> The “particle Lagrangian” and “particle continuity equation” are the Lagrangian/continuity equation of a *discretized* particle field, where each particle occupies an infinitely small volume. Exactly solving the continuity equation of a continuous fluid, of course, requires infinite resolution. This often leads to SPH being described as correct in the “molecular limit.” But at finite resolution, the more relevant limit is actually the “fluid limit.”

instead of the “traditional”  $\sim 32$  (Read et al. 2010; Dehnen & Aly 2012).<sup>3</sup>

However, all of these improvements only *reduce*, as opposed to *eliminate*, the associated errors. And all have costs, either in terms of CPU time and complexity, numerical noise, excessive diffusion, or slow convergence. As such, it is still unclear how general and accurate the “modern” SPH schemes are, and how competitive they are with other methods.

## 1.2 Stationary-Grid Methods (fixed-grids and AMR)

In grid-based methods, the volume is discretized into points or cells, and the fluid equations are solved across these elements. These methods are well-developed, with decades of work in computational fluid dynamics behind them. The most popular modern approach is embodied in finite-volume Godunov schemes,<sup>4</sup> which offer higher-order spatial accuracy,<sup>5</sup> numerical stability and relatively low diffusivity, and conservation of mass, linear momentum, and energy.

However, there are (inevitably) errors in these methods as well. Grid/AMR codes have significant advection errors, when fluids (especially with sharp gradients or discontinuities) move across cells in a manner un-aligned with the grid. These errors produce excessive diffusion and over-mixing, and can even manifest as unphysical forces. For example, rotating disks are “torqued” into alignment with the grid cardinal axes (“grid alignment”; see e.g. Hahn et al. 2010), shocks preferentially heat, propagate along, and “break out of” the grid axes (“carbuncle” instabilities; Peery & Im-lay 1988), contact discontinuities are rapidly “smeared out” upon advection, and so on. Related to this, the methods do not conserve angular momentum: at realistic resolutions for practical problems, gaseous orbits are usually degraded catastrophically within a couple orbital times (unless special polar/cylindrical coordinates are used, which is only possible if the problem geometry is known exactly ahead of time). The errors in these methods are also not velocity-independent: unlike SPH, “boosting” the fluid (so it uniformly moves across the grid in an AMR code) substantially increases the errors and diffusion everywhere in the problem (see Wadsley et al. 2008; Tasker & Bryan 2008; Springel 2010); this can

<sup>3</sup> It is sometimes said that “SPH does not converge,” or that “SPH is a second-order method” (i.e. converges as  $N^{-2}$  in a smooth 1D problem). Both of these are incorrect. SPH does converge at second order, but *only* in the limit where the number of neighbors inside the smoothing kernel goes to infinity ( $N_{\text{NGB}} \rightarrow \infty$ ), which eliminates the zeroth-order terms that do not converge away with increasing total particle number  $N$  alone. However increasing  $N_{\text{NGB}}$  is both expensive and leads to a loss of resolution (and in most actual practice is not actually done correctly as  $N$  increases). So the practical convergence rates of SPH are very slow.

<sup>4</sup> Older, finite-difference methods simply discretized the relevant equations onto interpolation/sampling points in a (non-moving) lattice, but these methods often do not conserve quantities like mass, momentum, and energy, require artificial viscosity/diffusion/hyperviscosity terms (as in SPH), and can be numerically unstable under long integrations for sufficiently complicated problems. As such they have proven useful mostly for weak linear-regime flows where strong shocks are absent and growth of e.g. momentum errors will not corrupt the entire domain; here there can be significant advantages from the fact that such methods very easily generalize to higher-order.

<sup>5</sup> Typically second-order, or third-order in the case of PPM methods. Some schemes claim much higher-order; however, it is almost always the case that this is true only for a sub-set of the scheme (e.g. a gradient estimator). In our convention, the order represents the convergence rate, which is limited by the lowest-order aspect of the method.

also strongly suppress fluid mixing instabilities (Springel 2010). It also means, if AMR methods are used, that the grid refinement must somehow “anticipate” the fluid motion (which is not generally possible), or else the accuracy in the refined regions is seriously compromised. Stationary-grid methods do not deal well with highly super-sonic flows (kinetic energy much larger than thermal), producing noise and excess entropy unless special fixes in the form of energy-entropy switches are adopted (Ryu et al. 1993; Bryan et al. 1995). They do not accurately maintain steep “edges” or gradients (e.g. diffusing the sharp surfaces of planets or stars outwards), and cannot naturally handle free surfaces (e.g. water flow in air). And a combination of mass fluxes through cells and the inherent mismatch between particle-based N-body methods and cell-based hydro methods means that various errors appear when the hydrodynamics are coupled to gravity, making it difficult for such methods to handle simple situations like self-gravitating hydrostatic equilibrium (see Müller & Steinmetz 1995; LeVeque 1998; Zingale et al. 2002). Worse, these errors can introduce spurious instabilities (Truelove et al. 1997, 1998).

In AMR methods specifically, the refinement criteria are flexible, but also necessarily ad-hoc, and can produce various artefacts. At refinement boundaries, there is a significant loss of accuracy (the method becomes effectively lower-order). This is related to the fact that the refinement is necessarily discontinuous. This means convergence is also slower in AMR. When coupled to gravity, various studies (O’Shea et al. 2005; Heitmann et al. 2008; Springel 2010) have also shown that these errors lead to suppression of low-amplitude gravitational instabilities (e.g. the instabilities that seed cosmological structure formation), and violate energy and momentum conservation in the long-range forces whenever cells are refined or de-refined.

Again, significant effort has gone into attempts to reduce these sources of error. Higher-order WENO-type schemes for gradients can help reduce edge effects. Various authors have implemented partially-Lagrangian or “arbitrary Lagrange-Eulerian” (ALE) schemes where partial distortion of the mesh is allowed, but then the meshes are re-mapped to regular meshes; or “patch” schemes in which sub-meshes are allowed to move, then mapped to larger “parent meshes” (see e.g. Gnedin 1995; Pen 1998; Trac & Pen 2004; Murphy & Burrows 2008). However, these approaches usually require foreknowledge of the exact problem geometry to work well (which is not possible for most interesting problems). And the re-mapping is a highly diffusive operation, so some of the errors above are actually enhanced. Moreover none of these actually eliminate the problems of continuous resolution improvement, advection errors, velocity-dependence of the errors, or angular momentum conservation.

### 1.3 Moving, Unstructured Meshes

Recently, there has been a surge in interest in moving, unstructured mesh methods. These methods are well-known in engineering (see e.g. Mavriplis 1997), and there have been earlier initial efforts in their development in astrophysics (e.g. Whitehurst 1995; Xu 1997), but recently considerable effort has gone into development of more flexible examples (Springel 2010; Duffell & MacFadyen 2011; Gaburov et al. 2012). These use a finite-volume Godunov method, but partition the volume into non-regular cells using e.g. a Voronoi tessellation, and allow the cells to move and deform continuously (usually in a manner related to the bulk flow of the fluid in the cell).

In many ways, moving meshes capture the advantages of both SPH and AMR codes: like SPH they can be Lagrangian and adapt

resolution continuously, feature velocity-independent truncation errors, couple well to gravity, and avoid preferred directions, while also like AMR treat shocks, shear flows, and fluid instabilities with high accuracy and eliminate many sources of noise, low-order errors, and artificial diffusion terms.

However, such methods are new, and still need to be tested to determine their advantages and disadvantages. It is by no means obvious that they are optimal for all problems, nor that they are the “best compromise” between Lagrangian (e.g. SPH) and Eulerian (e.g. grid) methods. And here are some problems the method does not resolve completely. Angular momentum is still not formally conserved in moving meshes, and it is not obvious (if the cell shapes are sufficiently irregular) how much it improves on stationary-grid codes. “Mesh-deformation” and “reconnection” in which distortions to the mesh lead to highly irregular cell shapes, is inevitable in complicated flows. This can lead to errors which effectively reduce the accuracy and convergence of the method, and would eventually crash the code. This is dealt with by some “mesh regularization,” by which the cells are re-shaped or prevented from deforming (i.e. made “stiff” or resistant to deformations). But such regularization obviously risks re-introducing some of the errors of stationary-grid methods which the moving-mesh method tries to avoid (the limit of a sufficiently stiff grid is simply a fixed-grid code with a uniform drift). And discontinuous cell refinement/de-refinement or “re-connection” is inevitable when the fluid motion is complicated, introducing some of the same errors as in AMR.

### 1.4 Structure of This Paper

Given the above, the intent of this paper is two-fold.

First, we will introduce and develop two new methods for solving the equations of hydrodynamics which attempt – like moving-mesh methods – to simultaneously capture the major advantages of both Lagrangian and Eulerian methods. The methods build on recent developments by many authors in the fluid dynamics community, especially Lanson & Vila (2008a,b), but have not generally been considered in astrophysics, except for recent efforts by Gaburov & Nitadori (2011). The methods move with the flow in a Lagrangian manner, adapt resolution continuously, eliminate velocity-dependent truncation errors, couple to N-body gravity methods exactly, have no preferred directions, do not require artificial diffusion terms, capture shocks, shear flows, and fluid instabilities with high accuracy, and exhibit remarkably good angular momentum conservation. We will show how these methods can be implemented into the flexible, parallel code GIZMO which includes gravity and cosmological integrations (build on the code foundations of GADGET-3), and a wide range of additional physics, as well as fully adaptive time-stepping and force resolution.<sup>1</sup>

Second, we will consider a systematic survey of a wide range of test problems, comparing both new methods, moving-mesh, modern stationary-grid, and both “traditional” and “modern” SPH methods. This is intended not just to validate our new methods, but also to test the existing major classes of numerical methods on a wide range of problems, to assess some of their relative strengths and weaknesses in different contexts.

In § 2, we describe the new methods. We present a simplified derivation of the equations of motion, and discuss how the method is implemented in the code GIZMO. This includes the hydrodynamic methods, coupling to gravity, cosmological integration, and time-integration scheme. A variety of more technical numerical details are included in the Appendices.

In § 5 we systematically compare the various numerical methods in a wide range of test problems. We briefly describe the

methods (§ 5.1), and then consider their performance in a series of tests, spanning equilibrium/steady-state configurations (§ 5.2), strong shocks and highly non-linear evolution (§ 5.3), a range of fluid mixing instabilities (§ 5.4), and tests with strong self-gravity, cosmological integrations, and non-linear additional physics such as cooling, star formation, stellar explosions, and radiation-gas coupling (§ 5.5). In § 6, we summarize the CPU performance (speed) of the different methods on the different classes of test problems.

Finally, in § 7, we summarize our conclusions and advantages/disadvantages of the new methods compared to SPH, AMR, and moving-mesh codes, and conclude with a discussion of possibilities for further improvements of the method.

## 2 A NEW NUMERICAL METHODOLOGY FOR HYDRODYNAMICS

In the last two decades, there has been tremendous effort in the computer science, engineering, and fluid dynamics literature, directed towards the development of new mesh-free algorithms for hydrodynamics; but much of this has not been widely recognized in astrophysics. Various authors have pointed out how matrix and least-squares methods can be used to define consistent, higher-order gradient operators, and renormalization schemes can be used to eliminate the zeroth-order errors of methods like SPH (see e.g. Dilts 1999; Oñate et al. 1996; Kuhnert 2003; Tiwari & Kuhnert 2003; Liu et al. 2005). Most of this has propagated into astrophysics in the form of “corrected” SPH methods, which partially-implement such methods as “fixes” to certain operators (e.g. García-Senz et al. 2012; Rosswog 2014) or in finite point methods, which simply treat all points as finite difference-like interpolation points rather than assigning conserved quantities (Maron & Howes 2003; Maron et al. 2012). However these implementations often sacrifice conservation (of quantities like mass, momentum, and energy) and numerical stability. Meanwhile, other authors have realized that the uncertain and poorly-defined artificial diffusion operators can be eliminated by appropriate solution of a Riemann problem between particle faces; this has generally appeared in the form of so-called “Godunov SPH” (Cha & Whitworth 2003; Inutsuka 2002; Cha et al. 2010; Murante et al. 2011). However on its own this does not eliminate other low-order SPH errors, and those errors can de-stabilize the solutions. A particularly intriguing development was put forward by Lanson & Vila (2008a,b). These authors showed that the advances above could be synthesized into a new, meshfree finite-volume method which is both consistent and fully conservative. Critically, rather than just attaching individual fixes piece-wise to an existing method, they re-derived the discrete operators from a consistent mathematical basis. A first attempt to implement these methods in an astrophysical context was presented in Gaburov & Nitadori (2011), and the results there for both hydrodynamic and magneto-hydrodynamic test problems appeared extremely encouraging. We therefore explore and extend two closely-related versions of this method here. Specifically, we will describe the new numerical methods and their implementation in the code GIZMO.

### 2.1 Derivation of the Meshless Equations of Motion

We begin with a heuristic derivation of the discretized equations governing the new numerical schemes. This will closely follow Gaburov & Nitadori (2011), and is aimed towards practical aspects of implementation. A fully rigorous mathematical formulation of the method, with proofs of various consistency, conservation, and convergence theorems, is presented in Lanson & Vila (2008a,b).

The Euler equations for hydrodynamics are ultimately a set of conservation laws for mass, momentum, and energy, which form a system of hyperbolic partial differential equations in a frame moving with velocity  $\mathbf{v}_{\text{frame}}$  of the form

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{F} - \mathbf{v}_{\text{frame}} \otimes \mathbf{U}) = \mathbf{S} \quad (1)$$

where  $\mathbf{S}$  is a source function for  $\mathbf{U}$ , the “state vector” of conserved (in the absence of sources) variables,

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho e \end{pmatrix} = \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho u + \frac{1}{2} \rho |\mathbf{v}|^2 \end{pmatrix} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ \rho u + \frac{1}{2} \rho |\mathbf{v}|^2 \end{pmatrix} \quad (2)$$

(where  $\rho$  is mass density,  $e$  is the total specific energy,  $u$  the specific internal energy, and the last equality expands the compact form of  $\mathbf{v}$  in 3 dimensions), and the tensor  $\mathbf{F}$  is the flux of conserved variables

$$\mathbf{F} = \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + P \mathcal{I} \\ (\rho e + P) \mathbf{v} \end{pmatrix} \quad (3)$$

where  $P$  is the pressure, and  $\mathcal{I}$  is the identity tensor.

As in the usual Galerkin-type method, to deal with non-linear and discontinuous flows we begin by determining the weak solution to the conservation equation. We multiply Eq. 1 by a test function  $\phi$ , integrate over the domain  $\Omega$  (in space and time, such that  $d\Omega = d^\nu \mathbf{x} dt$ , where  $\nu$  is the number of spatial dimensions), and follow an integration by parts of the  $\phi \nabla \cdot \mathbf{F}$  term to obtain:

$$0 = \int_{\Omega} (\dot{\mathbf{U}} \phi - \mathbf{F} \cdot \nabla \phi - \mathbf{S} \phi) d\Omega + \int_{\partial\Omega} (\mathbf{F} \phi) \cdot \hat{\mathbf{n}}_{\partial\Omega} d\partial\Omega \quad (4)$$

where  $\dot{f} \equiv \partial f / \partial t + \mathbf{v}_{\text{frame}}(\mathbf{x}, t) \cdot \nabla f$  is the co-moving derivative of any function  $f$ , and  $\hat{\mathbf{n}}_{\partial\Omega}$  is the normal vector to the surface  $\partial\Omega$ . The test function  $\phi = \phi(\mathbf{x}, t)$  is an arbitrary (differentiable) function. Assuming the fluxes and/or  $\phi$  vanish at infinity, and integrating the  $\dot{\mathbf{U}} \phi$  term by parts as well, we obtain

$$0 = \int_{\Omega} (\mathbf{U}(\mathbf{x}, t) \dot{\phi} + \mathbf{F}(\mathbf{U}, \mathbf{x}, t) \cdot \nabla \phi + \mathbf{S}(\mathbf{x}, t) \phi) d^\nu \mathbf{x} dt \quad (5)$$

To discretize this integral, we must now choose how to discretize the domain volume onto a set of points/cells/particles  $i$  with coordinates  $\mathbf{x}_i$ . If we chose to do so by partitioning the volume between the  $\mathbf{x}_i$  with a Voronoi mesh, we would obtain the moving-mesh method of codes like AREPO with the more accurate gradient estimators implemented in Mocz et al. (2014). Here we consider a mesh-free alternative, following Lanson & Vila (2008a,b); Gaburov & Nitadori (2011). Consider a differential volume  $d^\nu \mathbf{x}$ , at arbitrary coordinates  $\mathbf{x}$ ; we can partition that differential volume fractionally among the nearest particles through the use of a weighting function  $W$ , i.e. associate a fraction  $\psi_i(\mathbf{x})$  of the volume  $d^\nu \mathbf{x}$  with particle  $i$  according to a function  $W(\mathbf{x} - \mathbf{x}_i, h(\mathbf{x}))$ :

$$\psi_i(\mathbf{x}) \equiv \frac{1}{\omega(\mathbf{x})} W(\mathbf{x} - \mathbf{x}_i, h(\mathbf{x})) \quad (6)$$

$$\omega(\mathbf{x}) \equiv \sum_j W(\mathbf{x} - \mathbf{x}_j, h(\mathbf{x})) \quad (7)$$

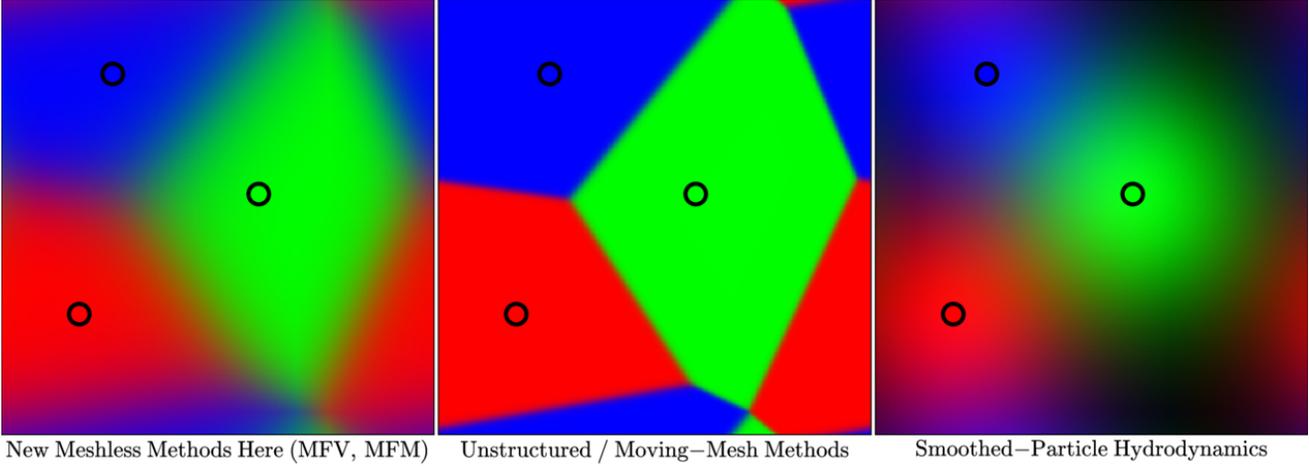
where  $h(\mathbf{x})$  is some “kernel size” that enters  $W$ . In other words, the weighting function determines how the volume at any point  $\mathbf{x}$  should be partitioned among the volumes “associated with” the tracer points  $i$ . Note that  $W$  can be, in principle, any arbitrary function; the term  $\omega(\mathbf{x})^{-1}$  normalizes the weights such that the total volume always sums correctly (i.e. the sum of fractional weights must

**Table 1.** Summary of Some Popular Numerical Hydrodynamics Methods

Method Name	Consistency /Order	Conservative? (Mass/Energy /Momentum)	Conserves Angular Momentum	Numerical Dissipation	Long-Time Integration Stability?	Number of Neighbors	Known Difficulties
<b>Smoothed-Particle Hydro. (SPH)</b>							
“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	✓	~ 128 – 442	excess diffusion, E0 errors
“Corrected” SPH (rpSPH, Integral-SPH, Morris96 SPH, Moving-Least-Squares SPH)	0-1	×	×	artificial viscosity	×	~ 32	errors grow non-linearly, “self-acceleration”
“Godunov” SPH (GSPH, GSPH-I02, Cha03 SPH)	0	✓	up to gradient errors	Riemann solver	✓	~ 300	instability, expense, E0 errors remain
<b>Finite-Difference Methods</b>							
Gridded/Lattice Finite Difference (ZEUS [some versions], Pencil code)	2-3	×	×	artificial viscosity	×	~ 8 – 128	instability, lack of conservation, advection errors
Lagrangian Finite Difference (PHURBAS, FPM)						~ 60	
<b>Finite-Volume Godunov Methods</b>							
Static Grids (ATHENA, PLUTO)	2-3	✓	×	Riemann solver	✓	~ 8 (geometric) ~ 8 – 125 (stencil)	over-mixing, ang. mom., velocity-dependent errors (VDE)
Adaptive-Mesh Refinement (AMR) (ENZO, RAMSES, FLASH)	2-3 (1)	✓	×	Riemann solver	✓	~ 8 – 48 ~ 24 – 216	over-mixing, ang. mom., VDE, refinement criteria
Moving-Mesh Methods (AREPO, TESS, FVMHD3D)	2	✓	×	Riemann solver	✓	~ 13 – 30	mesh deformation, ang. mom. (?), “remeshing”
<b>New Methods In This Paper</b>							
Meshless Finite-Mass & Meshless Finite-Volume (MFM, MFV)	2	✓	up to gradient errors	Riemann solver	✓	~ 32	remeshing noise ? (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 (this includes refinement boundaries in AMR).
- (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, *when the AM vector is unknown or not fixed in the simulation*. In this regime, 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. If the AM vector is known and fixed (e.g. for test masses around a single non-moving point-mass), it is always possible to construct a method (using cylindrical coordinates, explicitly advecting AM, etc.) which perfectly conserves it.
- (5) Numerical Dissipation: Source of numerical dissipation in e.g. shocks. Either this comes from an up-wind/Riemann solver type scheme, or artificial viscosity/conductivity/hyperdiffusion terms.
- (6) Integration Stability: States whether the method has long-term integration stability (i.e. errors do not 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! These are described in actual detail in the text. “Velocity-dependence” (as well as comments about noise and lack of conservation) here refers to the property of the *errors*, not the converged solutions. Any well-behaved code is conservative (of mass/energy/momentum/angular momentum), Galilean-invariant, noise-free, and captures the correct level of fluid mixing instabilities in the fully-converged (infinite-resolution) limit.



**Figure 1.** Illustration of key conceptual differences between some of the methods here. For an irregularly distributed set of sampling/grid points or “particles” (black circles) with locations  $\mathbf{x}_i$ , we require a way to partition the volume to solve the equations of hydrodynamics between them. *Left:* The meshless finite-mass (MFM) and meshless finite-volume (MFV) methods here. The volume partition is given by the weighted kernel at each point (Eq. 6); here the red/green/blue color channels represent the fraction of the volume at each point associated with the corresponding particle ( $\psi_i(\mathbf{x})$ ). Here we apply the same kernel function and typical kernel “width” as in the text. Note that this returns a Voronoi tessellation with the boundaries “smoothed.” Despite the *kernel* function being spherical, the *domains* associated with each particle are not, and the entire volume is represented. The fluid equations are then solved by integrating over the domain of each particle. *Center:* The unstructured/moving-mesh partition. Now the boundaries are strict step functions at the faces given by the tessellation. Note that this is (exactly) the limit of our MFM/MFV method for an infinitely sharply-peaked kernel function; technically the moving-mesh method is a special case of the MFV method. The volume integrals are then reduced to surface integrals across the faces. *Right:* The SPH partition. In SPH the contribution to volume integrals behaves as the kernel, centered on each particle location; the whole volume is “counted” only when the kernel size is infinitely large compared to the inter-particle spacing (number of neighbors is infinite). The equations of motion are evaluated at the particle locations  $\mathbf{x}_i$ , using the weighted-average volumetric quantities from the volume partition.

always be unity at every point). That said, to ensure the second-order accuracy of the method, conservation of linear and angular momentum, and locality of the hydrodynamic operations, the function  $W(\mathbf{x} - \mathbf{x}_i, h(\mathbf{x}))$  must be continuous, have compact support (i.e. have  $W = 0$  for sufficiently large  $|\mathbf{x} - \mathbf{x}_i| \gg h(\mathbf{x})$ ), and be symmetric (i.e. depend only on the absolute value of the coordinate differences  $|x - x_i|$ ,  $|y - y_i|$ , etc.). Because of the normalization by  $\omega(\mathbf{x})$ , the absolute normalization of  $W$  is irrelevant; so without loss of generality we take it (for convenience) to be normalized such that  $1 = \int W(\mathbf{x} - \mathbf{x}', h(\mathbf{x})) d^{\nu} \mathbf{x}'$ .

An example of this is shown in Fig. 1, with (for comparison), the volume partitions used in moving-mesh and SPH methods. We construct a two-dimensional periodic box of side-length unity with three randomly placed particles, and use a cubic spline kernel for  $W$  with smoothing length  $h$  set to the equivalent of what would contain  $\approx 32$  neighbors in 3D. We confirm that the entire volume is indeed partitioned correctly, like a Voronoi tessellation with the “edges” between particles smoothed (avoiding discontinuities in the “remapping” as particles move). In the limit where  $W$  is sufficiently sharply-peaked, we can see from Eq. 6 that we should recover *exactly* a Voronoi tessellation, because 100% of the weight ( $\psi(\mathbf{x})$ ) will be associated with the nearest particle. In fact, technically speaking, Voronoi-based moving-mesh methods are a special case of the method here, where the function  $W$  is taken to the limit of a delta function and the volume quadrature is evaluated exactly.

We now insert this definition of the volume partition into Eq. 5, and Taylor-expand all terms to second order accuracy in the smoothing length  $h(\mathbf{x})$  (e.g.  $f(\mathbf{x}) = f_i(\mathbf{x}_i) + h(\mathbf{x}_i) \nabla f(\mathbf{x} = \mathbf{x}_i) \cdot (\mathbf{x} - \mathbf{x}_i)/h(\mathbf{x}_i) + \mathcal{O}(h(\mathbf{x}_i)^2)$ ; the algebra is somewhat tedious but straightforward). Note that  $1 = \sum_i \psi_i(\mathbf{x})$ , and since the kernel has compact support,  $|\mathbf{x} - \mathbf{x}_i| \sim \mathcal{O}(h(\mathbf{x}_i))$  where  $W \neq 0$ . If we apply this to the integral of an arbitrary function (and assume the kernel

function is continuous, symmetric, and compact) we obtain

$$\int f(\mathbf{x}) d^{\nu} \mathbf{x} = \sum_i \int f(\mathbf{x}) \psi_i(\mathbf{x}) d^{\nu} \mathbf{x} \quad (8)$$

$$= \sum_i f_i(\mathbf{x}_i) \int \psi_i d^{\nu} \mathbf{x} + \mathcal{O}(h_i(\mathbf{x}_i)^2) \quad (9)$$

$$\equiv \sum_i f_i V_i + \mathcal{O}(h_i^2) \quad (10)$$

where  $V_i = \int \psi_i(\mathbf{x}) d^{\nu} \mathbf{x}$  is the “effective volume” of particle  $i$  (i.e. the integral of its volume partition over all of space). Applying the same to Eq. 5, evaluating the spatial integral, and dropping the  $\mathcal{O}(h^2)$  terms, we obtain

$$0 = \int dt \sum_i (V_i \mathbf{U}_i \dot{\phi}_i + V_i \mathbf{F}_i \cdot (\nabla \phi)_{\mathbf{x}=\mathbf{x}_i} + V_i \mathbf{S}_i \phi_i) \quad (11)$$

where  $\mathbf{F}_i \cdot (\nabla \phi)_{\mathbf{x}=\mathbf{x}_i}$  refers to the product of the matrix  $\mathbf{F}$  with the gradient of  $\phi$  evaluated at  $\mathbf{x}_i$ .

To go further, and remain consistent, we require a second-order accurate discrete gradient estimator. Here, we can use locally-centered least-squares matrix gradient operators, which have been described in many previous numerical studies (Dilts 1999; Oñate et al. 1996; Kuhnert 2003; Maron & Howes 2003; Maron et al. 2012; Tiwari & Kuhnert 2003; Liu et al. 2005; Luo et al. 2008; Lanson & Vila 2008a,b). Essentially, for any arbitrary configuration of points, we can use the weighted moments to defined a least-squares best-fit to the Taylor expansion of any fluid quantity at a central point  $i$ , which amounts to a simple (small) matrix calculation; the matrix can trivially be designed to give an arbitrarily high-order consistent result, meaning this method will, by construction, *exactly* reproduce polynomial functions across the particles up to the desired order, *independent* of the spatial configuration of the

particles. The second-order accurate expression is:

$$\begin{aligned}
 (\nabla f)_i^\alpha &= \sum_j \sum_{\beta=1}^{\beta=\nu} (f_j - f_i) \mathbf{B}_i^{\alpha\beta} (\mathbf{x}_j - \mathbf{x}_i)^\beta \psi_j(\mathbf{x}_i) + \mathcal{O}(h_i^2) \quad (12) \\
 &\equiv \sum_j (f_j - f_i) \tilde{\psi}_j^\alpha(\mathbf{x}_i) \\
 \tilde{\psi}_j^\alpha(\mathbf{x}_i) &\equiv \sum_{\beta=1}^{\beta=\nu} \mathbf{B}_i^{\alpha\beta} (\mathbf{x}_j - \mathbf{x}_i)^\beta \psi_j(\mathbf{x}_i) \equiv \mathbf{B}_i^{\alpha\beta} (\mathbf{x}_j - \mathbf{x}_i)^\beta \psi_j(\mathbf{x}_i)
 \end{aligned}$$

where we assume an Einstein summation convention over the Greek indices  $\alpha$  and  $\beta$  representing the elements of the relevant vectors/matrices, and the matrix  $\mathbf{B}_i$  is evaluated at each  $i$  by taking the inverse of another matrix  $\mathbf{E}_i$ :

$$\mathbf{B}_i \equiv \mathbf{E}_i^{-1} \quad (13)$$

$$\mathbf{E}_i^{\alpha\beta} \equiv \sum_j (\mathbf{x}_j - \mathbf{x}_i)^\alpha (\mathbf{x}_j - \mathbf{x}_i)^\beta \psi_j(\mathbf{x}_i) \quad (14)$$

Note that in Eqs. 12-14, we could replace the  $\psi_j(\mathbf{x}_i)$  with any other function  $\xi_j(\mathbf{x}_i)$ , so long as that function  $\xi$  is also continuous and compact. However, it is computationally convenient, and physically corresponds to a volume-weighting convention in determining the least-squares best-fit, to adopt  $\xi_j(\mathbf{x}_i) = \psi_j(\mathbf{x}_i)$ , so we will follow this convention. It is straightforward to verify that when the  $f_j$  follow a linear function in  $N$ -dimensions ( $f_j = f_i + \nabla f_{\text{true}} \cdot (\mathbf{x}_j - \mathbf{x}_i)$ ), this estimator *exactly* recovers the correct gradients.

Now, inserting this into Eq. 5, integrating by parts again, and noting that:

$$\begin{aligned}
 \sum_i V_i \mathbf{F}_i^\alpha (\nabla \phi)_i^\alpha &= \sum_i \sum_j V_i \mathbf{F}_i^\alpha (\phi_j - \phi_i) \tilde{\psi}_j^\alpha(\mathbf{x}_i) \quad (15) \\
 &= - \sum_i \phi_i \sum_j (V_i \mathbf{F}_i^\alpha \tilde{\psi}_j^\alpha(\mathbf{x}_i) - V_j \mathbf{F}_j^\alpha \tilde{\psi}_i^\alpha(\mathbf{x}_j))
 \end{aligned}$$

we obtain

$$\begin{aligned}
 0 &= \int dt \sum_i \phi_i \left( - \frac{d}{dt} (V_i \mathbf{U}_i) \right. \\
 &\quad \left. - \sum_j [V_i \mathbf{F}_i^\alpha \tilde{\psi}_j^\alpha(\mathbf{x}_i) - V_j \mathbf{F}_j^\alpha \tilde{\psi}_i^\alpha(\mathbf{x}_j)] + V_i \mathbf{S}_i \right) \quad (16)
 \end{aligned}$$

This must hold for any arbitrary test function  $\phi$ ; so therefore the expression inside the parenthesis must vanish, i.e.

$$\frac{d}{dt} (V_i \mathbf{U}_i) + \sum_j [V_i \mathbf{F}_i^\alpha \tilde{\psi}_j^\alpha(\mathbf{x}_i) - V_j \mathbf{F}_j^\alpha \tilde{\psi}_i^\alpha(\mathbf{x}_j)] = V_i \mathbf{S}_i \quad (17)$$

Now, rather than take the flux functions  $\mathbf{F}$  directly at the particle location and time of  $i$  or  $j$ , in which case the scheme would require some ad-hoc artificial dissipation terms (viscosity and conductivity) to be stable, we can replace the fluxes with the solution of an appropriate time-centered Riemann problem between the particles  $i$  and  $j$ , which automatically includes the dissipation terms. We define the flux as  $\tilde{\mathbf{F}}_{ij}$ ; this replaces both  $\mathbf{F}_i$  and  $\mathbf{F}_j$  since the solution is necessarily the same for both  $i$  and  $j$  “sides” of the problem;<sup>6</sup> this

<sup>6</sup> Note that this replacement of  $\mathbf{F}_i$  and  $\mathbf{F}_j$  can be directly derived, as well, by replacing the  $\mathbf{F}$  in the integral Eq. 16 with a Taylor expansion in space and time, multiplying the terms inside by  $1 = \sum \psi_i$ , centering the expansion about the symmetric quadrature point between  $i$  and  $j$  and centering it at the mid-point in time for a discretized time integral, and then evaluating the integrals to second order. For details, see Lanson & Vila (2008a).

gives

$$\frac{d}{dt} (V_i \mathbf{U}_i) + \sum_j \tilde{\mathbf{F}}_{ij}^\alpha [V_i \tilde{\psi}_j^\alpha(\mathbf{x}_i) - V_j \tilde{\psi}_i^\alpha(\mathbf{x}_j)] = V_i \mathbf{S}_i \quad (18)$$

Now, we can define the vector  $\mathbf{A}_{ij} = |A_{ij}| \hat{A}_{ij}$  where  $\mathbf{A}_{ij}^\alpha \equiv V_i \tilde{\psi}_j^\alpha(\mathbf{x}_i) - V_j \tilde{\psi}_i^\alpha(\mathbf{x}_j)$ , and the equations become:

$$\frac{d}{dt} (V_i \mathbf{U}_i) + \sum_j \tilde{\mathbf{F}}_{ij} \cdot \mathbf{A}_{ij} = V_i \mathbf{S}_i \quad (19)$$

This should be immediately recognizable as the form of the Godunov-type finite-volume equations. The term  $V_i \mathbf{U}_i$  is simply the particle-volume integrated value of the conserved quantity to be carried with particle  $i$  (e.g. the total mass  $m_i = V_i \rho_i$ , momentum, or energy associated with the particle  $i$ ); its time rate of change is given by the sum of the fluxes  $\tilde{\mathbf{F}}_{ij}$  into/out of an “effective face area”  $\mathbf{A}_{ij}$ , plus the total integral of source terms over the particle-associated volume  $V_i \mathbf{S}_i$ .

The approach to solving the discretized equations of the form in Eq. 19 is well-studied, and straightforward; we can use essentially the same schemes used in fixed-grid/AMR Godunov methods, and moving mesh codes. Specifically, we will employ a second-order accurate (in space and time) MUSCL-Hancock type scheme (van Leer 1984; Toro 1997), as used in state-of-the-art grid methods such as Fromang et al. (2006); Mignone et al. (2007); Cunningham et al. (2009); Springel (2010); Teyssier (2002). This involves a slope-limited, linear reconstruction of face-centered quantities from each particle/cell, a first-order drift/predict step for evolution over half a timestep, and then the application of a Riemann solver to estimate the time-averaged inter-particle fluxes for the timestep. We will describe these steps in more detail below.

We note that our method is not, strictly, a traditional Godunov scheme as defined by some authors, since we do not actually calculate a geometric particle face and transform a volume integral into a surface integral in deriving Eq. 19; rather, the “effective face” comes from solving the *actual volume integral*, over the partition defined by the weighting function, and this is simply the numerical quadrature rule that arises. But from this point onwards, it can be treated identically to Godunov-type schemes.

It should also be immediately clear from Eq. 19, that since we ultimately calculate fluxes of conserved quantities directly between particles, the conserved quantities themselves (total mass, linear momentum, and energy) will be conserved to machine accuracy *independent of the time-step, integration accuracy, and particle distribution*. Moreover, it is trivial to verify that  $\mathbf{A}_{ij} = -\mathbf{A}_{ji}$ , i.e. the fluxes are antisymmetric, so the flux “from  $i$  to  $j$ ” is always the negative of the flux “from  $j$  to  $i$ ” at the same time, and the discrete equations are manifestly conservative.

## 2.2 Gradient Estimation and Linear Reconstruction

In order to perform the particle drift/prediction operations, reconstruct quantities between particles for the Riemann problem, and perform a number of other operations, we require gradients of various hydrodynamic quantities. This is straightforward, since we have already defined an arbitrarily high-order accurate method for obtaining gradients using the least-squares matrix method in Eq. 12-14. We will use the second-order accurate version of this to define the gradient of a quantity  $(\nabla f)_i$  at position  $\mathbf{x}_i$ ; recall that these are exact for linear gradients and always give the least-squares minimizing gradient in other situations. As shown in Mocz et al. (2014), even in fixed-grid and moving-mesh codes, this gradient definition has a number of advantages over the usual finite-volume

definition (based on simple cell-to-cell differences over some stencil), especially in magnetohydrodynamics.

In Eq. 19, only the projection of the flux onto  $\hat{A}_{ij}$  is required; therefore the relevant flux  $\tilde{\mathbf{F}}_{ij} \cdot \hat{A}_{ij}$  can be obtained by solving a one-dimensional, unsplit Riemann problem in the frame of the quadrature point between the two particles. Because of the symmetry of the kernel, the relevant quadrature point at this order (the point where the volume partition between the two particles is equal) is the location along the line connecting the two which is an equal fraction of the smoothing length  $h$  from each particle, i.e.

$$\mathbf{x}_{ij} \equiv \mathbf{x}_i + \frac{h_i}{h_i + h_j} (\mathbf{x}_j - \mathbf{x}_i) \quad (20)$$

However, we note that we see very little difference in all the test problems here using this or the first-order quadrature point  $\mathbf{x}_{ij} = (\mathbf{x}_i + \mathbf{x}_j)/2$  (which can sometimes be more stable, albeit less accurate).<sup>7</sup>

So we must reconstruct the left and right states of the Riemann problem at this location: for a second-order method, we only require a linear reconstruction in primitive variables, so we require gradients and reconstructions of the density  $\rho$ , pressure  $P$  (and internal energy for a non-ideal equation of state), and velocity  $\mathbf{v}$ . For an interacting pair of particles  $i$  and  $j$ , the linearly-reconstructed value of a quantity  $f$  at a position  $\mathbf{x}$ , reconstructed from the particle  $i$ , is  $f_{\text{rec},i} = f_i + (\mathbf{x} - \mathbf{x}_i) \cdot (\nabla f)_i$  and likewise for the reconstruction from particle  $j$ ; these define the left and right states of the Riemann problem at the “interface”  $\mathbf{x}_{ij}$ . However, as in all Riemann-problem based methods, some slope-limiting procedure is required to avoid numerical instabilities near discontinuities, where the reconstruction can “overshoot” or “undershoot” and create new extrema (see e.g. Barth & Jespersen 1989). Therefore, in the reconstruction step (only), the gradient  $(\nabla f)_i$  above is replaced by an appropriately slope-limited gradient  $(\nabla f)_{\text{lim},i}$ , using the limiting procedure in Appendix B. We have experimented with a number of standard slope-limiters like that in Gaburov & Nitadori (2011) and find generally similar, stable behavior. However, as noted by Mocz et al. (2014), for unstructured point configurations in discontinuous Galerkin methods, there are some subtle improvements which can be obtained from more flexible slope-limiters. And indeed we find some significant improvement (albeit no major changes in the results we present in this paper) if we adopt the more complicated, but more flexible and closer to total variation diminishing slope-limiting procedure in Appendix B.

### 2.3 The Riemann Problem and Fluxes

The rest of the flux computation is straightforward, and for the most part follows AREPO (see Springel 2010, § 3.3). We briefly review it so we can note some subtle differences. We compute the solution to the Riemann problem in the rest-frame of the effective face between the two particles. Because the Riemann problem in our

<sup>7</sup> As shown in Inutsuka (2002), a higher-order quadrature rule between particles using the “traditional” SPH volume partition implies a quadrature point which is offset from the midpoint at  $\mathcal{O}(h^2)$ . It is straightforward to derive an analogous rule here, and we have experimented with this. However, we find no significant improvement in accuracy, presumably because the rest of the reconstruction we adopt is only second-order. Moreover, because Inutsuka (2002) derive this assuming there is always an exact linear gradient connecting the particles and extrapolate this to infinity beyond them, this can lead to serious numerical instabilities in the Riemann problem when there is some particle disorder.

method is always solved along the line connecting the particle centers of mass, the quadrature point at  $\mathbf{x}_{ij}$  moves with velocity (at second order)

$$\mathbf{v}_{\text{frame}} = \mathbf{v}_i + (\mathbf{v}_j - \mathbf{v}_i) \left[ \frac{(\mathbf{x}_{ij} - \mathbf{x}_i) \cdot (\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|^2} \right] \quad (21)$$

Beginning from a vector of primitive variables  $\mathbf{W} = (\rho, \mathbf{v}, P)$  for particles  $i$  and  $j$ , we first (for convenience) boost to the rest-frame of the face  $ij$ :

$$\mathbf{W}'_{j,i} = \mathbf{W}_{j,i} - \begin{pmatrix} 0 \\ \mathbf{v}_{\text{frame},ij} \\ 0 \end{pmatrix} \quad (22)$$

We then calculate left and right states by linear reconstruction of the values of  $\mathbf{W}'$  at  $\mathbf{x}_{ij}$  from particles  $i$  and  $j$ , following § 2.2; this gives us left and right  $\mathbf{W}'_{\text{rec},L}$  (the “ $j$  side”) and  $\mathbf{W}'_{\text{rec},R}$  (the “ $i$  side”). The states are also predicted forward in time by a half-timestep, to obtain time-centered fluxes:

$$\mathbf{W}''_{L,R} = \mathbf{W}'_{\text{rec},L,R} + \frac{\partial \mathbf{W}'_{L,R}}{\partial t} \frac{\Delta t}{2} \quad (23)$$

$$= \mathbf{W}'_{j,i} + (\nabla \mathbf{W}')_{\text{lim},j,i} \cdot (\mathbf{x}_{ij} - \mathbf{x}_{j,i}) + \frac{\partial \mathbf{W}'_{j,i}}{\partial t} \frac{\Delta t}{2} \quad (24)$$

where  $(\nabla \mathbf{W}')_{\text{lim},j,i}$  are the slope-limited gradients, and the partial time derivative is estimated based on the spatial derivatives using the Euler equations for an ideal gas, as

$$\frac{\partial \mathbf{W}}{\partial t} = - \begin{pmatrix} \mathbf{v} & \rho & 0 \\ 0 & \mathbf{v} & 1/\rho \\ 0 & \gamma P & \mathbf{v} \end{pmatrix} \nabla \mathbf{W} \quad (25)$$

(note that it is trivial to modify the pressure equation for a non-ideal gas using the gradients of  $\rho$  and  $u$ ).

We need to solve a 1D Riemann problem in the  $\mathbf{A}_{ij}$  direction; so we transform to a coordinate system aligned with the  $\hat{A}_{ij}$  axis; this can be done with the matrix  $\mathbf{\Lambda}$ , which rotates the vector (here, velocity) components, but obviously leaves the scalar components intact:

$$\mathbf{W}'''_{L,R} = \mathbf{\Lambda} \mathbf{W}''_{L,R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \mathbf{\Lambda}_\nu & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{W}''_{L,R} \quad (26)$$

where  $\mathbf{\Lambda}_\nu$  is an ordinary rotation matrix which takes the new coordinate system to coincide with the  $x'$  axis, i.e.  $\mathbf{A}'_{ij} = \mathbf{\Lambda}_\nu \mathbf{A}_{ij} = (|\mathbf{A}_{ij}|, 0, 0)$  in 3D.

We then solve the one-dimensional Riemann problem (see Appendix C), to obtain the fluxes

$$\tilde{\mathbf{F}}'_{ij} \equiv f_{\text{Riemann}}(\mathbf{W}'''_L, \mathbf{W}'''_R) \quad (27)$$

which we rotate back into the simulation axes (since we solved the one-dimensional Riemann problem in the frame aligned with  $\hat{A}_{ij}$ , this automatically projects the fluxes appropriately):

$$\tilde{\mathbf{F}}'_{ij} \equiv \hat{A}_{ij} \cdot (\mathbf{\Lambda}^{-1} \tilde{\mathbf{F}}'_{ij}) = \begin{pmatrix} F_\rho \\ \mathbf{F}_v \\ F_e \end{pmatrix} \quad (28)$$

and finally “de-boost” back into the simulation coordinate frame following Pakmor et al. (2011)

$$\tilde{\mathbf{F}}_{ij} \cdot \hat{A}_{ij} \equiv \tilde{\mathbf{F}}'_{ij} + \begin{pmatrix} 0 \\ \mathbf{v}_{\text{frame}} F_\rho \\ \frac{1}{2} |\mathbf{v}_{\text{frame}}|^2 F_\rho + \mathbf{v}_{\text{frame}} \cdot \mathbf{F}_v \end{pmatrix} \quad (29)$$

Just as in AREPO, it is easy to verify that this scheme maintains Galilean invariance and eliminates the velocity (boost)-dependent truncation errors in non-moving meshes, if the particles move with the fluid bulk velocity (discussed below).

## 2.4 Setting Particle Velocities: The Arbitrary-Lagrangian-Eulerian Nature of the Method

Note that so far, we have dealt primarily with the *fluid* velocity  $\mathbf{v} = \mathbf{v}(\mathbf{x})$ , which defined the fluid momentum associated with each particle. We have not actually specified the velocity of the *particles* (e.g. the  $\mathbf{v}_i$  which enters in determining the velocity of the frame in Eq. 21). It is, for example, perfectly possible to solve the above equations, with the particle positions *fixed*; everything above is identical except the frame velocity is zero and the particles are not moved between timesteps. This makes the method fully Eulerian; since the particle volumes depend only on the relative particle positions, they do not change in time, and we could even choose an initial particle distribution to reproduce e.g. the equations of motion of a stationary mesh method. On the other hand, we could set the velocities equal to the fluid velocity, in which case we obtain a Lagrangian and boost-independent method. Indeed, it is straightforward to see that our derivation thus far describes a truly Arbitrary Lagrangian-Eulerian (ALE) method.

In this paper, we will – with a couple of noted exceptions shown for demonstration purposes – set the particle velocities  $\mathbf{v}_i$  to match the mean fluid velocities. More specifically, we choose velocities such that the “particle momentum”  $m_i \mathbf{v}_i$  is equal to the fluid momentum integrated over the volume associated with the particle. This amounts numerically to treating the fluid and particle velocities at the particle positions as the same quantity. This is the Lagrangian mode of the method, which has a number of unique advantages we will describe. In a pure Eulerian form, on the other hand, most of the advantages of the new methods here compared to stationary grid codes are lost; moreover when there is bulk advection across the particles, their geometric arrangement becomes more important or else lower-order errors can appear.

That said, we emphasize that more complicated and flexible schemes are possible, and may be advantageous under some circumstances. For example, the particles could move with a “smoothed” fluid velocity, which would capture bulk flows but reduce re-meshing noise in complicated (e.g. locally turbulent) flows (an idea which has been explored in both SPH and moving-mesh codes; see Imaeda & Inutsuka 2002; Duffell & MacFadyen 2014).

## 2.5 What Motion of the “Face” Means: The Difference Between MFV and MFM Assumptions

There is another important subtlety in our calculation of the fluxes above. The left and right states at the quadrature points and mid-point in time are well-defined. However, the distortions of the effective volume with time are more complex. When we solve the Riemann problem in Eq. 27, we have to ask how the volumes assigned to one particle vs. the other are “shifting” during the timestep.

One choice is to assume, that since we boosted to a frame moving with the velocity of the quadrature point assuming the time-variation in smoothing lengths was second-order (compared to the fluid bulk velocity), the “face” is exactly stationary in this frame ( $\mathbf{v}_{\text{eff}}^{\text{frame}} = 0$ ). This is what we would obtain in e.g. a moving-mesh finite-volume method, where the face motion can be chosen (in principle) arbitrarily and the faces are locally simple, flat, “planes” of arbitrary extent in the directions perpendicular to the quadrature point. This was the choice made in Gaburov & Nitoro (2011), for example. We will consider this, and it defines what we call our “Meshless Finite-Volume” or MFV method. The choice is analogous to the finite-volume method: we are solving the Riemann problem across a plane whose relative position to the mesh-generating points is (instantaneously) fixed in time.

However, when the fluid flow is complicated, there is relative particle motion which changes the domain, leading to higher-order corrections. Moreover, assuming  $\mathbf{v}_{\text{eff}}^{\text{frame}} = 0$  does not necessarily capture the true up-wind motion of the face (i.e. the state is a predicted state, not necessarily the exact time-centered state). Since we derived this method with the assumption that the particles move with the fluid, we could adopt an alternative assumption, namely that the Lagrangian volume is distorting with the mean (time-centered and face-area averaged) motion of the volume partition, such that the mass on either “side” of the state is conserved. In practice, this amounts to an identical procedure as in § 2.3 and in our MFV case, but in the Riemann problem itself (Eq. 27), we solve the problem assuming the face has a residual motion  $\mathbf{v}_{\text{eff}}^{\text{frame}} = S_*$ , where  $S_*$  is the usual “star state” velocity (the speed of the contact wave in the Riemann problem), on either side of which mass is conserved. Note that this does not require that we modify our boost/de-boost procedure, since the frame we solve the problem in is ultimately arbitrary so long as we correct the quantities appropriately. This assumption defines what we call our “Meshless Finite-Mass” (MFM) method, because it has the practical effect of eliminating mass fluxes between particles. This choice is analogous to the finite-element method: we are solving the Riemann problem across a complicated Lagrangian boundary distorting with the relative fluid flow. We stress that this is *only* a valid choice if the particles are moved with the fluid velocity as in our default implementation of the methods here; otherwise the MFM choice has a zeroth-order error (obvious in the case where particles are not moving but the fluid is).

This distinction may seem arbitrary, and indeed it is. However a couple of points are important to note. First, for a smooth flow (with only linear gradients), it is straightforward to show that the MFM and MFV reduce to each other (they become *exactly* identical). So the difference is only at second-order, which is the order of accuracy in our method in any case. Second, it is true that, in situations with more complicated flows, particle disorder and finite resolution, because we cannot perfectly follow the distortion of Lagrangian faces (and calculate the position of the face only at second-order accuracy), the assumption made in the MFM method for the motion of the face in the Riemann problem will not *exactly* match the “real” motion of the face calculated by directly time-differencing the positions estimated for it across two timesteps. However, the error made is second-order (in smooth flows); and moreover, this is true for MFV and most moving-mesh finite-volume methods as well (since the faces are not “rigidly” anchored to the particles or mesh-generating points).

So both methods have different finite numerical errors. We will systematically compare both, to study how these affect certain problems. Not surprisingly, we will show that the differences are maximized at discontinuities, where the method becomes low-order in both cases, and the two should not be identical.

## 2.6 Smoothing Lengths and Particle “Volumes”

Note that, in this method, the smoothing length  $h$  does not play any “inherent” role in the dynamics and could in principle follow many different definitions; however, it does closely relate to the “effective volume” of a particle. This suggests setting it so that some (relatively small) number of neighbors is enclosed by the compact kernel function centered at each particle, as in SPH; this also makes the resolution intrinsically adaptive. Moreover it is also implicit in our derivation and the second-order accuracy of the scheme that  $h(\mathbf{x})$  vary smoothly across the flow. Therefore, like in most modern SPH schemes, we do not set the particle-centered  $h_i = h(\mathbf{x}_i)$

to enclose some actual discrete “number of neighbors,” (which is discontinuous) but rather follow Hopkins (2013) and Gaburov & Nitadori (2011) and constrain it by the smoothed particle number density  $n_i \equiv n(\mathbf{x}_i) = \omega(\mathbf{x}_i)$  (see e.g. Springel & Hernquist 2002; Monaghan 2002). In  $\nu$  dimensions, this is

$$N_{\text{NGB}} = C_\nu n_i h_i^\nu = C_\nu h_i^\nu \sum_j W(\mathbf{x}_j - \mathbf{x}_i, h_i) \quad (30)$$

where  $C_\nu = 1, \pi, 4\pi/3$  for  $\nu = 1, 2, 3$ , respectively, and  $N_{\text{NGB}}$  is a constant we set, which is the “effective” neighbor number (close to, but not necessarily equal to, the discrete number of neighbors inside  $h_i$ ). Just as in Hopkins (2013) this is solved iteratively (but the iteration imposes negligible cost).<sup>8</sup> Note that, unlike a “constant mass in kernel” approach (where the mass density instead of particle density is used to constrain  $h_i$ ) as in Springel & Hernquist (2002), this choice guarantees we successfully eliminate discontinuities in the smoothing length.<sup>9</sup>

With this definition of  $h$ , we can also now calculate the effective volume of a particle,  $V_i = \int \psi_i(\mathbf{x}) d^\nu \mathbf{x}$ . Inserting the above, and keeping terms up to second order accuracy, we find

$$V_i = \int \psi_i(\mathbf{x}) d^\nu \mathbf{x} = \omega(\mathbf{x}_i)^{-1} \left(1 + \mathcal{O}(h^2)\right) \quad (31)$$

In fact, this expression is exact if  $h$  is locally constant (does not vary over the kernel domain). For  $h$  being a general function of position, it is not possible to analytically solve for the exact  $V_i$ ; however this expression is second-order accurate so long as the variation of  $h$  across the kernel obeys  $|\nabla h| \lesssim 1$ , which our definition maintains (except at discontinuities, where this drops to first order-accurate). We stress that the method is still a “partition of unity” method (see Lanson & Vila 2008a,b), since our equations of motion were derived from an exact and conservative analytic/continuum expression for the volume partition; this is distinct from SPH where even in the continuum limit, volume is not conserved. However, the quadrature rule we use on-the-fly to estimate the volume integral associated with a given particle is only accurate in our scheme to the same order as the integration accuracy. This does not, therefore, reduce the order of the method; however, we will show that it does lead to noise in some fields, compared to methods with an exact discretized volume partition (e.g. meshes). If desired, an arbitrarily high-order numerical quadrature rule could be applied to evaluate Eq. 31; this would be more expensive but reduce noise. We will consider such experiments in future work, but for now proceed with the simpler quadrature rule here.

We should stress that while some authors have advocated using the continuity equation ( $dh/dt = \nu^{-1} h \nabla \cdot \mathbf{v}$ ) to evolve the smoothing lengths (and this is done in e.g. GASOLINE), it is not

<sup>8</sup> We use an iteration scheme originally based on that in Springel & Hernquist (2002), which uses the continuity equation to guess a corrected  $h_i$  each timestep, then uses the simultaneously computed derivatives of the particle number density to converge rapidly; and we have further optimized the scheme following Cullen & Dehnen (2010) and our own SPH experiments in Hopkins (2013); Hopkins et al. (2013b). This means that once a solution for  $h_i$  is obtained on the first timestep, usually  $< 1\%$  of active particles require multiple iteration (beyond a first-pass) in future timesteps and almost none require second iterations, so the CPU cost compared to a single-sweep is negligible (and the gains in accuracy are very large).

<sup>9</sup> We should also note that because the smoothing length is ultimately arbitrary, so long as it is continuous (as far as the formal consistency, conservation, and accuracy properties of our method are concerned), the particle number density estimator in Eq. 30 does not actually have to reproduce the “true” particle number density, just a continuous and finite approximation.

a stable or accurate choice for this method. As noted by Gaburov & Nitadori (2011), the results of such an exercise depend on the discretization of the divergence operator in a way that is not necessarily consistent, and more worryingly, this will inevitably produce dis-continuous smoothing lengths in sufficiently complex flows, reducing the accuracy and consistency of the method.

## 2.7 Integration, Timestep Criteria, and Adaptive Time-Stepping

The integration scheme here closely follows that in AREPO, itself similar to that in GADGET-3. We refer to Springel (2010), § 7, but again review briefly so we can note some differences.

Single-stage second-order accurate time integration is achieved as in Colella (1990); Stone et al. (2008). For a vector of conserved quantities  $\mathbf{Q}_i = (V\mathbf{U})_i$ ,

$$\mathbf{Q}_i^{(n+1)} = \mathbf{Q}_i^{(n)} + \Delta t \left\langle \frac{d\mathbf{Q}_i}{dt} \right\rangle \equiv \mathbf{Q}_i^{(n)} + \Delta t \frac{d\mathbf{Q}_i^{(n+1/2)}}{dt} \quad (32)$$

$$= \mathbf{Q}_i^{(n)} - \Delta t \sum_j \mathbf{A}_{ij} \cdot \tilde{\mathbf{F}}_{ij}^{(n+1/2)} \quad (33)$$

For almost all interesting problems, there is a large dynamic range and so using a global timestep imposes a severe resolution penalty. Therefore we use individual timesteps, following the standard principle in  $N$ -body problems, SPH simulations (Katz et al. 1996; Springel 2005) and AMR codes. Specifically, we follow the elegant method described in Springel (2010) (§ 7.2). We discretize allowed timestep sizes into a power-of-two hierarchy (i.e. the timestep of particle  $i$  is the largest power-of-two subdivision smaller than the locally-calculated timestep criterion), so that there is a nested hierarchy of timestep bins (i.e. on a given timestep  $\Delta t_i$ , all particles with timesteps  $\Delta t_j \leq \Delta t_i$  are synchronized). Conserved quantities exchanged between cells are always updated synchronously: whenever a flux is calculated between two adjacent particles  $i$  and  $j$ , if the timesteps differ, the conserved quantities on *both sides* of the face are updated according to the flux calculation on the smaller of the two timesteps (this is akin to “sub-cycling” in AMR codes). Whenever a cell completes its timestep, its primitive variables are updated based on the accumulated change in its conserved quantities; cells which are between timesteps (but interacting with “active” cells) use their old primitive variables and gradients (calculated from their last active timestep), drifted according to our predict-step to the synchronous time, to compute the relevant quantities for flux estimation. On the completion of any sub-timestep, the timestep size for any particle may be updated; however, the particle must move into a timestep bin which will be active on the next sub-step. In other words, a the timestep can always be reduced after a sub-step is completed, but it can only increase when the steps are appropriately synchronized. Because the scheme strictly deals with pairwise exchanges of conserved fluid quantities, it remains *manifestly* conservative of mass, momentum, and energy, even while adaptive/individual timesteps are used. We can always enforce a global timestep if desired; however, like Springel (2010), we actually find that this method performs as accurately in practice (at vastly lower computational cost) to the use of a global timestep, if we use an appropriate particle-based timestep criterion.

For hydrodynamics, we employ a local Courant-Fridrichs-Levy (CFL) timestep criterion, similar to that in SPH; for consis-

tency with previous work we define it as

$$\Delta t_{\text{CFL},i} = 2 C_{\text{CFL}} \frac{h_i}{|v_{\text{sig},i}|} \quad (34)$$

$$v_{\text{sig},i} = \text{MAX}_j \left[ c_{s,i} + c_{s,j} - \text{MIN} \left( 0, \frac{(\mathbf{v}_i - \mathbf{v}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j)}{|\mathbf{x}_i - \mathbf{x}_j|} \right) \right] \quad (35)$$

where  $h_i$  is the smoothing length defined above,  $\text{MAX}_j$  refers to the maximum over all interacting neighbors  $j$  of  $i$ , and  $|v_{\text{sig}}|$  is the signal velocity (Whitehurst 1995; Monaghan 1997b).<sup>10</sup> This, together with our individual timestep method above, is sufficient to ensure numerical stability. However, it is still possible that in e.g. a high-velocity shock, particles with a very long timestep will suddenly have neighbor particles with a much shorter timestep; we do not want the conserved properties of the long-timestep particles to change “too much” before they are active, since this would entail a loss of accuracy. We therefore combine the CFL condition with the criterion from Saitoh & Makino (2009) as updated in Durier & Dalla Vecchia (2012); Hopkins et al. (2013b). Specifically, if a particle has an interacting neighbor with a timestep more than two timebins “lower” (factor 4); it is “woken up” and moved to the shortest active timebin, with its subsequent timestep criteria re-evaluated. We find this is more than sufficient to completely eliminate artificial particle “inter-penetration” even in extreme situations (e.g. a collision of two initially well-separated cold blobs with relative  $|\mathbf{v}|/c_s \sim 10^6$ ).

Note that, because our method is Lagrangian, when the bulk velocity of the flow is super-sonic ( $|\mathbf{v}| \gg c_s$ ), the signal velocity (which only depends on the relative approach velocity of particles) is typically still close to  $c_s$ . Contrast this to stationary grid methods, where  $v_{\text{sig}}$  must include the velocity of the flow *across the grid*. This means we can take much larger timesteps with this method, with no loss of accuracy; in some of the test problems below, the difference is a factor of  $\gtrsim 1000$  in timestep size.

Finally, whenever other physics are present, we note that there are other timestep criteria; the minimum over all such criteria is always chosen. For example, with gravity present, we use a kinematic criterion as in Power et al. (2003),  $\Delta t_{\text{kin}} = (2 \alpha_k \epsilon_{\text{grav}} / |\mathbf{a}|)^{1/2}$ , where  $|\mathbf{a}|$  is the total acceleration and  $\epsilon_{\text{grav}}$  is the force softening (typical  $\alpha_k \lesssim 0.01$ ), along with standard restrictions based on the particle displacement relative to the local particle separation; other physics like diffusion, conduction, nuclear reaction networks, chemistry, cooling, star formation and stellar evolution, black hole accretion, and radiative transfer all add their own restrictions.

We also note that, like all conservative methods based on a Riemann solver, when flows are totally dominated by kinetic energy, small residual errors can appear in the thermal energy which are large compared to the correct thermal energy solution. This is a well-known problem, and there are various means to deal with it, but we adopt the combination of the “dual energy” formalism and energy-entropy switches described in § D. If we are evolving the internal energy explicitly instead of the total energy, the update scheme is identical to the scheme above; we simply replace the flux with the appropriate value. It is worth noting here, though, that the Lagrangian nature of our method minimizes this class of errors compared to stationary-grid codes.

<sup>10</sup> Note that the normalization convention here is familiar in SPH, but different from most grid codes (in part because  $h_i$  is not exactly the same as the “cell size”). For our standard choice of smoothing kernel, a choice  $C_{\text{CFL}} = 0.2$  is equivalent to  $C_{\text{CFL}} = 0.8$  in an AMR code with the convention  $\Delta t_{\text{CFL}} = C_{\text{CFL}} \Delta x_{\text{cell}} / (c_s + |\mathbf{v}_{\text{gas}}|)$ .

## 2.8 Higher-Order Versions of the Scheme

It is straightforward to extend most elements of this method to higher-order, but for now, we will only briefly comment on this. The moving, weighted-least squares gradient estimators can be trivially extended to arbitrarily high order if higher-order gradients are desired; it simply increases the size of the matrix which must be calculated between neighbors (see Bilotta et al. 2011). As discussed in Gaburov & Nitadori (2011), Appendix A, this makes it straightforward to perform the reconstruction for the Riemann at equivalent higher-order, for example they explicitly show the equations needed to make this a piecewise parabolic method (PPM). From the literature on finite-volume Godunov methods, there are also well-defined schemes for higher-order time-integration accuracy, which can be implemented in this code in an identical manner to stationary-grid codes. However, if we wish to make the method completely third-order (or higher) at all levels, we also need to re-derive an appropriate quadrature rule; that can trivially be done numerically via Gaussian quadrature (see e.g. Maron et al. 2012), however it is computationally expensive, so an analytic quadrature expression would be more desirable. Finally, this quadrature rule, if used, should also be used to re-discretize the equation of motion (i.e. correct the “effective face” terms in § 2.1), to complete the method.

Following Gaburov & Nitadori (2011), we have performed some limited experiments with higher-order PPM-type reconstructions and timestep schemes, and find little improvement, because the largest effects seen in Eulerian codes are generally minimized when the particles move in a Lagrangian manner, and because the  $N$ -body gravity solver itself is only second-order accurate. Higher-order schemes for gravity may be more important in most of the problems we are interested in. Still, it would be interesting to explore these further in future work, especially since we do see numerical errors associated with our finite-order quadrature rule in the tests below.

## 3 GRAVITY & COSMOLOGY IN GIZMO

In astrophysics, gravity is almost always an important force. Indeed, as stressed by Springel (2010), there is essentially no point in solving the hydrodynamic equations more accurately in most astrophysical problems if gravity is treated less accurately, since the errors from gravity will quickly overwhelm the real solution. A major advantage, therefore, of the new methods proposed here is that they, like SPH, couple naturally, efficiently, and *accurately* to  $N$ -body based gravitational solvers.

### 3.1 The General Gravitational Force Solver

The  $N$ -body solver used here is essentially the same as in GADGET-3 (from which GIZMO is directly built), with some improvements primarily for optimization purposes. Like GADGET-3, we have the option of using a hybrid Tree or Tree-Particle Mesh (TreePM) scheme; these schemes are computationally efficient, allow automatic and continuous adaptivity of the gravitational resolution in collapsing or expanding structure, and can be computed very accurately. Moreover, we have further added a fully conservative and flexible treatment of adaptive gravitational softening lengths so that the spatial resolution of gravity always accurately corresponds to that in the gas.

### 3.2 Coupling Hydrodynamics to Gravity

When gravity is present, it modifies the Euler equations with the addition of source terms for momentum and energy. As reviewed by

Springel (2010) (§ 5 therein), it has historically been challenging to couple these terms accurately to finite-volume grid codes. The “standard” approach from Müller & Steinmetz (1995); Truelove et al. (1998) in most current finite-volume codes leads to first-order energy conservation errors (i.e. these schemes are no longer actually second-order), which do not converge in time (i.e. cannot be controlled with finer timesteps); these can corrupt the true solution on problems like the Evrard test and exacerbate the spurious noise and gravitational heating in problems like the Santa Barbara cluster. A method which explicitly conserves total energy can be constructed (if we use single global timesteps and explicitly calculate the gravitational potentials, which is very expensive), but this leads to catastrophic errors in the thermal energy evolution in gravity-dominated flows which totally corrupt or crash many of the test problems here. A more accurate coupling, which is spatially and temporally second-order accurate in the integration of gravity and conservation of energy (conservation is exact for a linear gravitational force law), is given by retaining the definition of  $E_i$  as the “hydrodynamic total energy,”  $E_i = U_i + \mathbf{P}_i \cdot \mathbf{P}_i / (2m_i)$ , and adding the appropriate gravitational work corrections as

$$\mathbf{P}_i^{(n+1)} = \mathbf{P}_i^{(n)} + \Delta \mathbf{P}_{\text{hydro}} \quad (36)$$

$$\begin{aligned} & - \frac{\Delta t}{2} \left[ m_i^{(n)} \nabla_i \Phi^{(n)} + m_i^{(n+1)} \nabla_i \Phi^{(n+1)} \right] \\ E_i^{(n+1)} &= E_i^{(n)} + \Delta E_{\text{hydro}} \quad (37) \\ & - \frac{\Delta t}{2} \left[ m_i^{(n)} \mathbf{v}_i^{(n)} \cdot \nabla_i \Phi^{(n)} + m_i^{(n+1)} \mathbf{v}_i^{(n+1)} \cdot \nabla_i \Phi^{(n+1)} \right] \\ & - \frac{\Delta t}{2} \nabla_i \Phi^{(n)} \cdot \sum_j (\mathbf{x}_i - \mathbf{x}_j)^{(n)} \frac{dm_{ij}}{dt} \\ & - \frac{\Delta t}{2} \nabla_i \Phi^{(n+1)} \cdot \sum_j (\mathbf{x}_i - \mathbf{x}_j)^{(n+1)} \frac{dm_{ij}}{dt} \end{aligned}$$

where  $\Delta \mathbf{P}_{\text{hydro}}$  is the hydrodynamical momentum flux ( $\Delta E_{\text{hydro}}$  the hydrodynamical energy flux),  $-\nabla_i \Phi$  is the standard gravitational acceleration calculated from our force solver at the beginning ( $n$ ) and end ( $n+1$ ) of the timestep, and  $dm_{ij}/dt$  is the contribution to the mass flux calculated between particles  $j$  and  $i$  at the same timestep (this is the extra gravitational work owing to mass fluxes, which vanishes when they are not present).

An advantage of our particle-based method is that it removes many of the ambiguities in coupling gravity to finite-volume systems. Essentially all  $N$ -body solvers implicitly neglect mass fluxes in calculating the forces; our Lagrangian methods either completely eliminate or radically reduce these fluxes, eliminating or reducing second-order errors in the forces. And because of the symmetry of the kernel function, it is straightforward to show that the coupling of gravity to the particle at its center (i.e. calculating  $\nabla_i \Phi$  at the particle coordinates  $\mathbf{x}_i$ , as for a point mass), as is done in SPH codes, is accurate to second-order in the gravitational forces (third-order in the potential), as good as the  $N$ -body solver itself for collisionless particles.

### 3.3 Adaptive, Fully-Conservative Gravitational Softening

To calculate the gravitational forces generated by gas, (as opposed to coupling to the gas), we need to solve for the potential field. In  $N$ -body codes, this requires us to decompose the mass density field into the contributions “from” each particle/cell. For these methods this is straightforward: based on our definition of the volume partition, the differential mass at a point  $\mathbf{x}$  which is “associated with” a

given particle  $i$  is just

$$dm_i = d^{\nu} \mathbf{x} \rho(\mathbf{x}) \frac{W(\mathbf{x} - \mathbf{x}_i, h(\mathbf{x}))}{\omega(\mathbf{x})} \quad (38)$$

If we further use our definition of  $h^{\nu} n = h^{\nu} \omega \propto N_{\text{NGB}}$ , and note that our normalization of  $W$  to integrate to unity requires  $W \propto h^{-D} \tilde{w}(|\mathbf{x} - \mathbf{x}_i|/h(\mathbf{x}))$  where  $\tilde{w}$  is the dimensionless “shape function” set by the smoothing kernel choice, then we see  $dm_i \propto d^{\nu} \mathbf{x} \rho(\mathbf{x}) \tilde{w}(|\mathbf{x} - \mathbf{x}_i|/h(\mathbf{x}))$ . Expanding this to leading order in the gradients of  $\rho$  and  $h$ , we can re-write it  $dm_i \approx m_i W(\mathbf{x} - \mathbf{x}_i, h_i) d^{\nu} \mathbf{x}$ . In other words, to leading order, the density distribution “associated with” a given particle has the same functional form as the kernel centered at the particle. More exactly, it is straightforward to show that the potential computed by integrating Poisson’s equation with the source in Eq. 38 is identical at leading order to the potential we would obtain using just the particle-centered smoothing kernel mass distribution (and it is second-order accurate if we average in spherical shells).

This suggests that we should treat the particles in the  $N$ -body code as standard  $N$ -body particles “softened” by the kernel function with the same softening/smoothing; in this case the resolution of gravity and hydrodynamics is always equal and the two actually use the *same*, consistent set of assumptions about the mass distribution. This also avoids the ambiguities and associated errors of many mesh-based codes, which must usually adopt some ad-hoc assumptions to treat Cartesian cells or complicated Voronoi cells as simplified “spheres” with some “effective” radius.<sup>11</sup>

With that assumption, then, the force softening is straightforward. For the cubic spline kernel in 3D, for example, we have

$$W(q, h_i) = \frac{8}{\pi h_i^3} \begin{cases} 1 + 6q^2(q-1) & (0 \leq q < \frac{1}{2}) \\ 2(1-q)^3 & (\frac{1}{2} \leq q < 1) \\ 0 & (q \geq 1) \end{cases} \quad (39)$$

$$\Phi_i(q, h_i) \equiv G m_i \phi(q, h_i) \quad (40)$$

$$\phi(q, h_i) = -\frac{1}{q h_i} \begin{cases} \frac{14}{5}q - \frac{16}{3}q^3 + \frac{48}{5}q^5 - \frac{32}{5}q^6 & (0 \leq q < \frac{1}{2}) \\ -\frac{1}{15} + \frac{16}{5}q - \frac{32}{3}q^3 + 16q^4 & (\frac{1}{2} \leq q < 1) \\ 1 & (q \geq 1) \end{cases} \quad (41)$$

where  $q \equiv |\mathbf{x} - \mathbf{x}_i|/h_i$ . Note that on scales  $> h$ , the potential and force are exactly that of a Newtonian point mass. It is common practice to compare softenings to an “equivalent” Plummer sphere softening; for this choice, the Plummer equivalent softening is  $\sim h/3$ .

Because the smoothing lengths change, we must be careful to maintain energy and momentum conservation correctly. Fortunately, Price & Monaghan (2007) show how the appropriate terms can be rigorously derived from the particle Lagrangian to maintain manifest conservation with variable softening lengths. If we define

<sup>11</sup> Strictly speaking, at small separations, this mis-match in mesh-based methods leads to low-order errors in the sense that the gravitational forces calculated from the cell deviate from the true force associated with that cell geometry. If the field is well-resolved so that the gravitational forces require the collective effect of many cells, the errors are diminished rapidly, but for self-gravitating regions near the resolution limit, the errors can be significant (and increase as cells become less spherical).

the gravitational self-energy of a system of gas particles as

$$E_{\text{grav}} = \frac{1}{2} \sum_{i,j} G m_i m_j \phi(r_{ij}, h_j) \quad (42)$$

and then follow the same derivation as Price & Monaghan (2007), accounting for the fact that we use a slightly different convention to determine the smoothing length  $h$ , we then obtain the acceleration equation for gravity:

$$\begin{aligned} m_i \frac{d\mathbf{v}_i}{dt} &= -\nabla_i E_{\text{grav}} \quad (43) \\ &= -\sum_j \frac{G m_i m_j}{2} \left( \left. \frac{\partial \phi(r, h_i)}{\partial r} \right|_{r_{ij}} + \left. \frac{\partial \phi(r, h_j)}{\partial r} \right|_{r_{ij}} \right) \frac{\mathbf{r}_{ij}}{r_{ij}} \\ &\quad - \sum_j \frac{G}{2} \left( \zeta_i \left. \frac{\partial W(r, h_i)}{\partial r} \right|_{r_{ij}} + \zeta_j \left. \frac{\partial W(r, h_j)}{\partial r} \right|_{r_{ij}} \right) \frac{\mathbf{r}_{ij}}{r_{ij}} \\ \zeta_a &\equiv m_a \frac{h_a}{n_a \nu} \frac{1}{\Omega_a} \sum_b m_b \left. \frac{\partial \phi(r_{ab}, h)}{\partial h} \right|_{h=h_a} \quad (44) \end{aligned}$$

$$\begin{aligned} \Omega_a &\equiv 1 + \frac{h_a}{n_a \nu} \frac{\partial n_i}{\partial h_i} \quad (45) \\ &= 1 - \frac{h_a}{n_a \nu} \sum_b \left( \left. \frac{r_{ab}}{h_a} \frac{\partial W(r, h_a)}{\partial r} \right|_{r_{ab}} + \frac{\nu}{h_a} W(r_{ab}, h_a) \right) \end{aligned}$$

where  $\mathbf{r}_{ij} = \mathbf{x}_i - \mathbf{x}_j$  (so  $\partial \phi / \partial r = h^{-1} \partial \phi / \partial q$ ). The first part (in  $\partial \phi / \partial r$ ) here is just the usual  $\nabla \Phi$  term, assuming  $h$  is fixed; the force between each particle pair is symmetrized so forces are always equal and opposite.<sup>12</sup> The second part (the  $\zeta$  terms in  $\partial W / \partial r$ ) accounts for the fact that the  $h$  change, so by moving the particles we change  $h$ , which in turn does additional work by changing the potentials for the particles (note that this term always vanishes outside the smoothing radius, as it should).

We emphasize that these equations for the gravitational forces are *manifestly* conservative, based on our definition of the potential and its relationship to the kernel. Moreover, for any given mass distribution which follows the kernel  $W$ , the equations are also *exact*, to *all* orders in  $h$  and all orders in the gradients of the density field, etc. Errors in gravity still enter, of course, because we approximate the long-range force given by this equation through standard  $N$ -body methods rather than solving the direct  $N^2$  problem, but this has nothing to do with the force softening itself.

Finally, this suggests that we can and should do the same for other (non-gaseous) volume-filling fluids, like dark matter, when they appear in our simulations. This amounts to making the same assumption as for gas: that the volume is partitioned according to a kernel function. This removes the otherwise ad-hoc assumption that dark matter particles simply represent soft-edged “spheres”; it simply requires that we determine a dark matter smoothing length in an identical manner to how we determine it for gas. This is trivial: we define the kernel function  $W$  and  $n_i^{\text{DM}} = \omega_i^{\text{DM}}$  the same as gas, but based on the neighbor *dark matter* particles (not gas particles), and then apply the same constraint equation  $h_{\text{DM},i}^{\nu} n_{\text{DM},i} \propto N_{\text{DM}}^{\text{DM}}$ . This then produces an identical set of equations for gravity, for these particles, with one important caveat: when gas and/or dark matter particles interact via gravity within their smoothing radii, the

<sup>12</sup> This trivially follows in the point-mass  $r > h$  regime, but is non-trivial at small radii if the  $h$  differ. As discussed in Price & Monaghan (2007), there are other possible choices for how this can be symmetrized; for example by using a mean  $h$  between the particles, or simply using the larger of the two  $h$  (as is done in GADGET). However, these introduce discontinuous changes in the gravitational softening, which break conservation in the  $\nabla_i h_j$  terms.

$\zeta$  terms only appear if both particles  $ij$  are the same type. This is because these terms stem from how the mutual motion of the particles will change the smoothing lengths (based on inter-particle distances for the same types).<sup>13</sup>

### 3.4 Cosmological Integration

In an expanding (or contracting) Universe, the Euler equations plus gravity, as described above, must be modified to account for the expansion of space; we do so in the same manner as GADGET (see Springel 2005, for details). If we adopt the useful co-moving coordinates  $\mathbf{x}_c \equiv a^{-1} \mathbf{x}$  (our internal position variable; where  $a = 1/(1+z)$  is the scale factor at redshift  $z$ ),  $\rho_c = a^3 \rho$ ,  $P_c = (\gamma - 1) \rho_c u$ , and define the peculiar velocity  $\mathbf{v}_p = a d\mathbf{x}_c / dt$  (distinct from the physical velocity between two points with physical separation  $\mathbf{r}$ ,  $\mathbf{v}_{\text{phys}} = H(a) \mathbf{r} + \mathbf{v}_p$ , and canonical momentum  $\mathbf{p}_c = m a^2 d\mathbf{x} / dt$ , which we use as our internal velocity variable), and hydrodynamic energy  $E = U + m \mathbf{v}_p \cdot \mathbf{v}_p / 2$  (our internal energy variable), together with the Hubble expansion  $H(a) = \dot{a}/a$ , comoving gravitational potential  $\Phi_c$ , and gradient operator  $\nabla_c$  acting on co-moving coordinates, then the Euler equations take on a “normal” form with simple source terms. Specifically, per Springel (2010), the appropriate “conserved” variables in hydrodynamic interactions are still mass, momentum, and energy, and the appropriate surface/volume integrals still yield the “standard” fluxes of the Euler equations in the non-cosmological frame, evaluated using the appropriate physical fluid quantities and cell effective areas.

So we do not need to change our Riemann solution method except to make sure the units are correctly converted into physical before the flux computation and back to co-moving after. Quantities like the particle positions and momenta do not need to be explicitly evolved under the influence of the Hubble expansion since the distance units are co-moving (and velocity units are canonical momenta); the cosmological “integration” is perfect in this sense. However, we work with the “hydrodynamic energy,” as described above, and evolve the internal energy (rather than e.g. the particle entropy). This means we do need to include a source term for energy evolution, which is implemented with our usual second-order time integration scheme as gravity (so that the prediction is appropriately included in obtaining time-centered fluxes), giving

$$E_i^{(n+1)} = E_i^{(n)} - \Delta t [H(a^{(n)}) E_i^{(n)} + H(a^{(n+1)}) E_i^{(n+1)}] + \dots \quad (46)$$

where the ... represents the usual non-cosmological terms. If we evolve the internal energy directly, the integration can be done exactly, by simply summing the exact adiabatic expansion of the gas under two “pure cosmological” half-timesteps with evolution from  $a^{(n)} \rightarrow a^{(n+1/2)}$  and  $a^{(n+1/2)} \rightarrow a^{(n+1)}$ .

## 4 SMOOTHED-PARTICLE HYDRODYNAMICS IN GIZMO

Having implemented these new methods in the code GIZMO, we note that, with a few straightforward modifications, we can run the code as an SPH code. The details of “SPH mode” in GIZMO are given in Appendix F. In “SPH mode,” almost everything in the code is identical including: gravity, time-integration and timestep limitation, definition of the smoothing lengths, functional form of the

<sup>13</sup> The generalization to multi-fluid simulations is similarly straightforward. However, in some simulations, there are particle types – for example, stars in cosmological simulations – for which it is not obvious that an adaptive force softening is always appropriate (as opposed to a fixed force softening), because they do not necessarily represent a volume-filling fluid.

kernel, and gradient estimation for certain explicitly-called gradients in SPH used for e.g. the predict step. The essential changes are just in (1) the computation of volumetric quantities (e.g. density, pressure), and (2) the computation of “fluxes” (there is no mass flux in SPH and no Riemann problem, but we can use the same apparatus of our time integration treating the results from the standard SPH equation of motion as momentum and energy fluxes). We implement both a “traditional” and “modern” SPH in GIZMO, as described below.

## 5 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 stationary grid codes.

### 5.1 Reference Methods for Test Problems

In the tests below, we will generally consider six distinct numerical methods for treating the hydrodynamics (sometimes with individual variations). These and other methods are roughly summarized in Table 1. They include:

- **Meshless Finite-Volume (MFV):** This refers to the meshless finite-volume formulation which we present in § 2. This is one of the two new methods used here, specifically the quasi-Lagrangian formulation which includes inter-particle mass fluxes. We use the implementation in GIZMO for all runs, with the details of the scheme following those outlined above. But we have confirmed that it gives very similar results on the same tests to the simplified, earlier implementation in Gaburov & Nitadori (2011).

- **Meshless Finite-Mass (MFM):** This refers to the other of the two new methods developed here in § 2. Specifically, this is the Lagrangian formulation which conserves particle masses. As in the MFV case, we use the implementation in GIZMO for all runs. As such, up to the details of the frame in which the Riemann problem is solved (as discussed in § 2.5), the code structure is exactly identical.

- **“Traditional” SPH (TSPH):** This refers to the “traditional” or “old-fashioned” SPH formulations, described in § 1 and § F. This is essentially the version of SPH in the most-commonly used versions of codes like GADGET (Springel 2005), GASOLINE (Wadsley et al. 2004), TREE-SPH (Hernquist & Katz 1989), and others. By default, it uses a Lagrangian, fully-conservative equation of motion (Springel & Hernquist 2002), a cubic spline smoothing kernel with  $\sim 32$  neighbors, a standard (constant) Morris & Monaghan (1997) artificial viscosity with a Balsara (1989) switch for shear flows, no artificial conductivity or other artificial diffusion terms, and the standard (error-prone) SPH gradient estimators. To make our methods comparison as exact as possible, we use the TSPH version implemented in GIZMO (§ F1), so that the code architecture is identical up to the details of the hydro solver. As such the results are not exactly identical to other TSPH-type codes; however, we have re-run several comparisons with both GADGET and GASOLINE and find the differences are very small (and none of our conclusions would change if we used these as our reference TSPH code).

- **“Modern” SPH (PSPH):** This refers to “modern” SPH formulations described in § 1 and § F2. This is essentially the version of SPH in the P-SPH code used for the FIRE project simulations (Hopkins 2013; Hopkins et al. 2013b), and the adapted version of P-SPH in SPHGal (Hu et al. 2014). But it also gives very similar results

to the SPH formulations in SPHS (Read & Hayfield 2012), PHANTOM (Price & Federrath 2010), and Rosswog (2014). As above, to make our comparison as fair as possible, we use the version of PSPH implemented in GIZMO.

- **Moving-Mesh Method:** This refers to the unstructured, moving-mesh finite-volume Godunov schemes described in § 1.3. These are the schemes in AREPO (Springel 2010), TESS (Duffell & MacFadyen 2011), and FVMHD3D (Gaburov et al. 2012)<sup>14</sup>. While we have made partial progress in implementing moving meshes into GIZMO, this remains incomplete at present. Therefore we will use AREPO as our default comparison code for this method, instead. This is convenient, since both GIZMO and AREPO share a common evolutionary history; much of the underlying code architecture (for example, the parallelization, timestep scheme, and gravity solver) is similar. Most of the AREPO results shown here are exactly those in the code methods paper (Springel 2010).

- **Stationary Grids:** This refers to the non-moving grid codes in both time-invariant “fixed grid” and adaptive mesh (AMR) methods, described in § 1.2. There are many codes which adopt such a method, for example ATHENA (Stone et al. 2008) and PLUTO (Mignone et al. 2007) in the “fixed grid” sub-class and ENZO (O’Shea et al. 2004), RAMSES (Teyssier 2002), ART (Krafftsov et al. 1997) and FLASH in the AMR sub-class. The code ATHENA represents the state of the art and is often considered a “gold standard” for comparison studies with these methods; we therefore consider it representative of the class and will use it as our example for this method in most cases (unless otherwise specified). However, we have re-run a subset of our tests with ENZO, RAMSES, and FLASH, and confirm that we obtain very similar results. We stress that on almost every test here, AMR does not improve the results relative to fixed-grid codes *at fixed resolution* (and in fact it can introduce more noise and diffusion in several problems) – it only improves things if we allow refinement to much larger cell number, in which case the same result would be obtained by simply increasing the fixed-grid resolution. This is because most of our test problems (with a couple exceptions that we note) would require refinement everywhere simultaneously. Unfortunately, none of these codes shares a common architecture in detail with GIZMO or AREPO; so we do our best to control for this by running these codes wherever possible with the same choice of Riemann solver, slope limiter, order of the reconstruction method, Courant factor and other timestep criteria, and of course resolution. Where possible, we have also compared runs with AREPO in “fixed grid” mode (the mesh is Cartesian and frozen in time); as shown in Springel (2010), this gives very similar results to ATHENA on the problems studied.

#### 5.1.1 Comments on Other, Alternative Numerical Methods

Before going on, we briefly comment on a couple of the other methods discussed in Table 1, to which we have compared a limited sub-set of problems, but will not show a systematic comparison.

First, we note that GIZMO can easily be run in “corrected” SPH mode, and we have explicitly run several test problems using the rpSPH method (Abel 2011), the Morris (1996) method, the MLS-SPH, (Dilts 1999), and exact-integral SPH method (García-Senz et al. 2012; Rosswog 2014).<sup>15</sup> However, as noted in § 1, all

<sup>14</sup> A public version of FVMHD3D is available at <https://github.com/egaburov/fvmhd3d>

<sup>15</sup> It is worth noting that there are many classes of “corrected” SPH, many

of these methods sacrifice conservation of energy and linear momentum, and numerical integration stability. As shown in Hopkins (2013) (see also Abel 2011), on many test problems (for example, a low-resolution Sedov blastwave), this leads to catastrophic errors that grow exponentially and totally dominate the solution. Most will crash on more complicated test problems (the Keplerian disk, interacting blastwave, Noh, Zeldovich, and Santa Barbara cluster). And even on the problems where they run stably, these methods do not eliminate all the low-order SPH errors (just the zeroth-order pressure gradient error in the equation of motion); therefore we do not see much performance improvement compared to PSPH (which is stable and conservative).

Next, we can also (trivially) run GIZMO as a “Godunov SPH” code. We have done so using both the “standard” implementation by Cha & Whitworth (2003), and using the improved version from Inutsuka (2002); Cha et al. (2010); Murante et al. (2011) which uses a higher-order quadrature rule for the equation of motion. However, we note that while this eliminates the need for artificial viscosity and conductivity terms in SPH, it does not inherently resolve any of the other errors in the method (e.g. low-order errors and zeroth-order inconsistency, the surface tension error, etc.). Because of this, we find, as did Cha et al. (2010), that it does not yield any noticeable improvement for fluid mixing instabilities (e.g. the Kelvin-Helmholtz test); what *does* improve things in their tests is going to many more neighbors ( $N_{\text{NGB}} \gtrsim 300$ ) and using a higher-order quadrature rule, but this is already implicit in PSPH. And accuracy on some tests is noticeably poor compared to PSPH (e.g. the “blob” test; see Fig. 12 in Murante et al. 2011). And while Godunov SPH has certain formal stability properties, it is very difficult in practice to maintain non-linear stability in the Riemann solver given the low-order SPH errors which survive, unless huge neighbor numbers are used (this has been a significant barrier to implementation). This instability appears in mild form in large post-shock oscillations in shocktubes in the papers above, but for some of the complicated test problems here (e.g. the Zeldovich and Noh problems), the errors crash the method.

Finally, we have also compared several test problems to finite-difference codes. We have modified GIZMO to run as a Lagrangian finite-difference code, in which case the volumetric quantities (density, etc.) rather than the conserved quantities are evolved explicitly with our matrix gradient estimator; this makes it more similar to the method in PHURBAS (Maron et al. 2012) or FPM.<sup>16</sup> We have also compared to fixed-grid Cartesian finite difference methods; specifically using the ZEUS code (Stone & Norman 1992), but this is similar in many tests to other methods like those in the PENCIL code (Brandenburg & Dobler 2002). These methods perform quite well on some tests where the flow is smooth and density fluctuations are small, such as the (non-boosted) Gresho vortex test, or the sub-sonic Kelvin-Helmholtz test. However, as with “corrected” SPH, we find that the sacrifice of conservation in these methods can lead to serious non-linear errors in many of the other test problems we consider here. The methods fail on all the cosmological tests and strongly self-gravitating tests, as well as the Noh problem (see Liska & Wendroff 2003) and the interacting blastwave

of which can be summarized within the context of the “reproducing kernel particle method” as generally developed in Liu et al. (1995).

<sup>16</sup> By FPM, we refer to the “Finite Point Method” in e.g. Oñate et al. (1996); this includes methods referred to as “finite pointset methods” and “Lagrangian finite point methods” in e.g. Kuhnert (2003); Tiwari & Kuhnert (2003), as well as the “finite particle method” of Liu et al. (2005).

problem. Even where the methods run well, we do not see any major improvement compared to the Godunov method in ATHENA (as noted by the code authors themselves in e.g. Maron et al. 2012, Stone et al. 2008). For a more thorough discussion of the conservation properties and stability of these methods, we refer to Clarke (2010).

Given these results, we conclude that these methods, in their present form, are not optimal for the kinds of problems we are interested in for GIZMO. Although they may have advantages for specific sub-categories of problems (for example, sub-sonic, smooth flows where higher-order methods such as those in the PENCIL code can be easily implemented and stabilized), given their difficulties running the full set of test problems in our suite, will not consider them further.

## 5.2 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!

### 5.2.1 Linear Traveling Soundwave: Convergence Testing

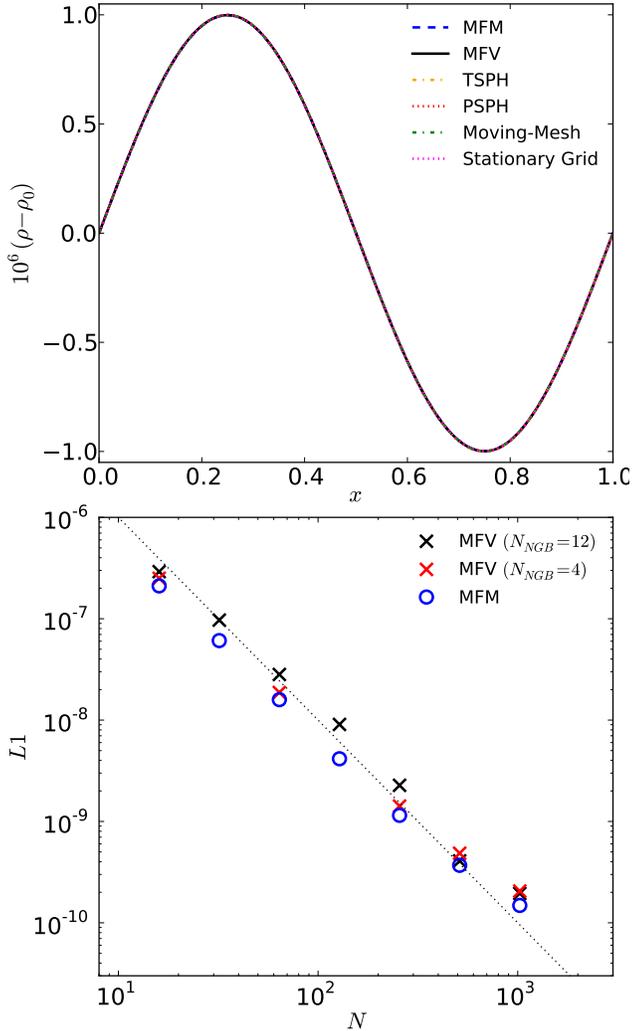
We begin by considering a simple linear one-dimensional soundwave.<sup>17</sup> This problem is analytically trivial; however, since virtually all schemes are first-order for discontinuities such as shocks, smooth linear problems with known analytic solutions are the only way to measure and quantitatively test the accuracy and formal convergence rate of numerical algorithms. Following Stone et al. (2008), 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. 2 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

$$\text{L1} = \frac{1}{N} \sum_i |\rho_i - \rho(x_i)| \quad (47)$$

where  $N$  is the number of particles,  $\rho_i$  is the numerical solution for cell  $i$ , and  $\rho(x_i)$  is the analytic solution (identical to the initial conditions for this problem). Fig 2 shows the error norm as a function of the particle number: for both the MFM and MFV methods, the results show second-order convergence (as expected for a smooth problem and a second-order accurate method). The MFM 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  $N_{\text{NGB}} = 4$  neighbors in one dimension unless otherwise specified, for this and all other 1D tests; however for the MFM and MFV methods the results are insensitive to this choice as long as it is not too small, ( $\leq 2$ ). We show this explicitly by comparing the L1 norm for  $N_{\text{NGB}} = 12$  for the MFV method (the MFM result is similar). If anything, the L1 norm is slightly *larger* for high  $N_{\text{NGB}}$ ; this is because of the additional numerical diffusion from solving the Riemann problem across more neighbors. The L1 norm becomes slightly larger again for  $N_{\text{NGB}} < 4$ , because there are not

<sup>17</sup> See <http://www.astro.princeton.edu/~jstone/Athena/tests/linear-waves/linear-waves.html>



**Figure 2.** Linear traveling soundwave test problem (§ 5.2.1). *Top:* Soundwave evolved one period; we solve the problem with each of the methods shown, corresponding to our new lagrangian method (MFM), 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 indistinguishable 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  $N_{\text{NGB}}$  is fixed while  $N$  is varied; we plot results for two choices of  $N_{\text{NGB}}$  for the MFV method. Convergence does not require increased  $N_{\text{NGB}}$ .

enough particles in the stencil, but in all cases it exhibits second-order convergence.

We have evolved the wave to  $\sim 1000$  periods in the MFM 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.

### 5.2.2 The Square Test: Advection & Surface Tension Errors

We next consider the “square” test common for recent SPH studies (Cha et al. 2010; Heß & Springel 2010; Saitoh & Makino 2013; Hopkins 2013). 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 arbitrary large initial velocity  $v_x = 142.3$ ,  $v_y = -31.4$ . We use  $64^2$  particles. Fig. 3

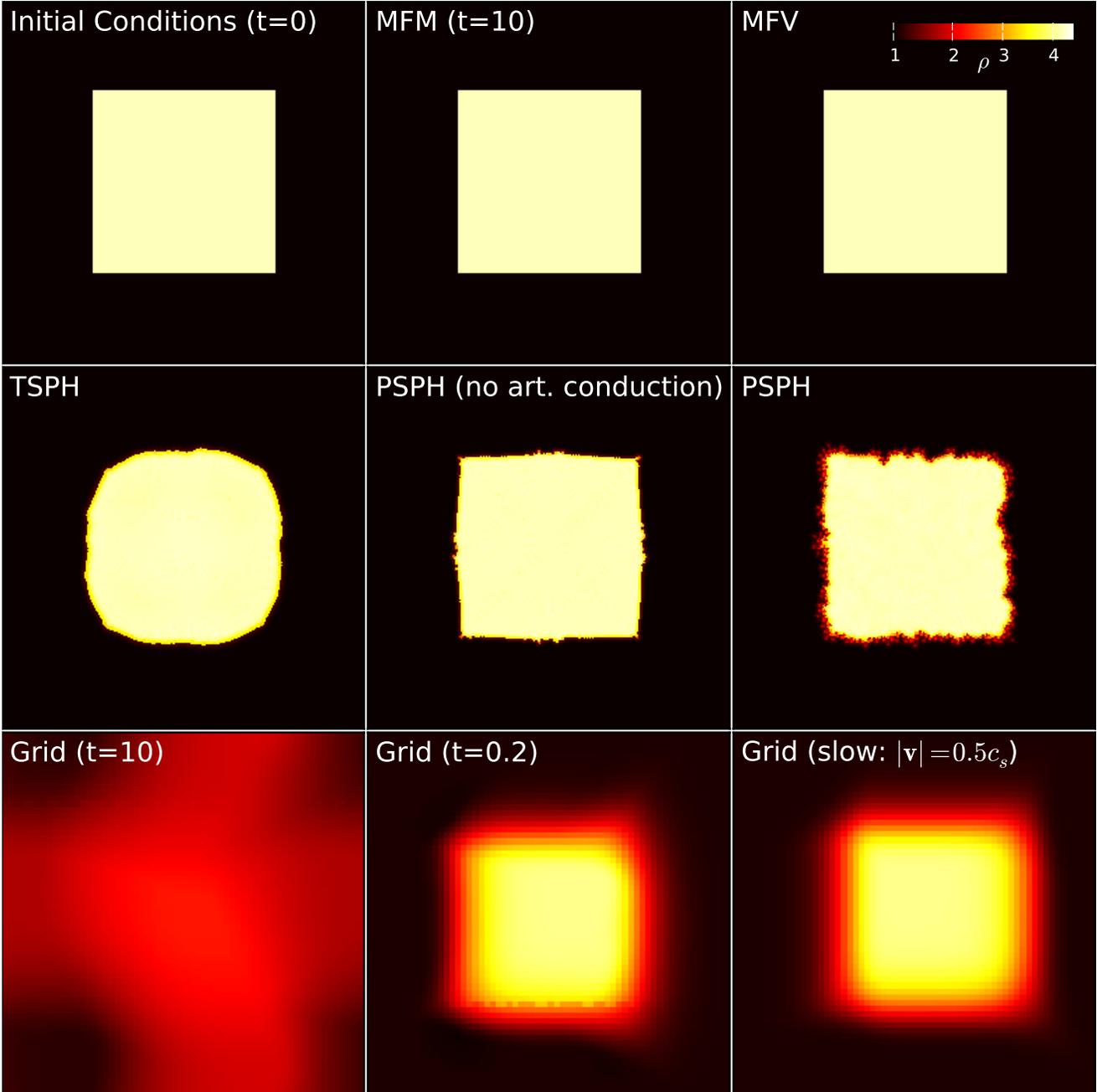
shows the initial condition and the resulting system evolved to a time  $t = 10$ , centered on the central square position at that time. The square should be in hydrostatic equilibrium, with the final state exactly identical to the initial state.

The MFM 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 well-known “pressure blips” in shocktube tests. 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 Hopkins 2013; Hu et al. 2014); 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 timestep criterion should include the relative gas-grid motion: thus these errors appear despite the fact that the timesteps in the grid code are a factor of  $\sim 1000$  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 MFM 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), MFM, 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 Springel 2010). 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-



**Figure 3.** Hydrostatic square advection test (§ 5.2.2). *Top Left:* Initial condition (yellow shows density  $\rho = 4$ , black  $\rho = 1$ ), a high-density square in hydrostatic equilibrium, with all fluid moving at constant velocity ( $|\mathbf{v}| \sim 150$ ). The periodic box ( $0 < x < 1, 0 < y < 1$ ) is shown. *Top Center:* MFM 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. Moving-meshes 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:* Stationary grid (here ATHENA): advection errors completely destroy the square, despite forcing  $\sim 1000$  times smaller timesteps in this case. *Bottom Center:* Stationary grid result at time  $t = 0.2$ , showing the magnitude of distortion after the square moves a few times its size. *Bottom Right:* Stationary grid result for a slower sub-sonic ( $|\mathbf{v}| = 0.5$ ) advection at  $t = 10$  (the square has traveled much less distance).

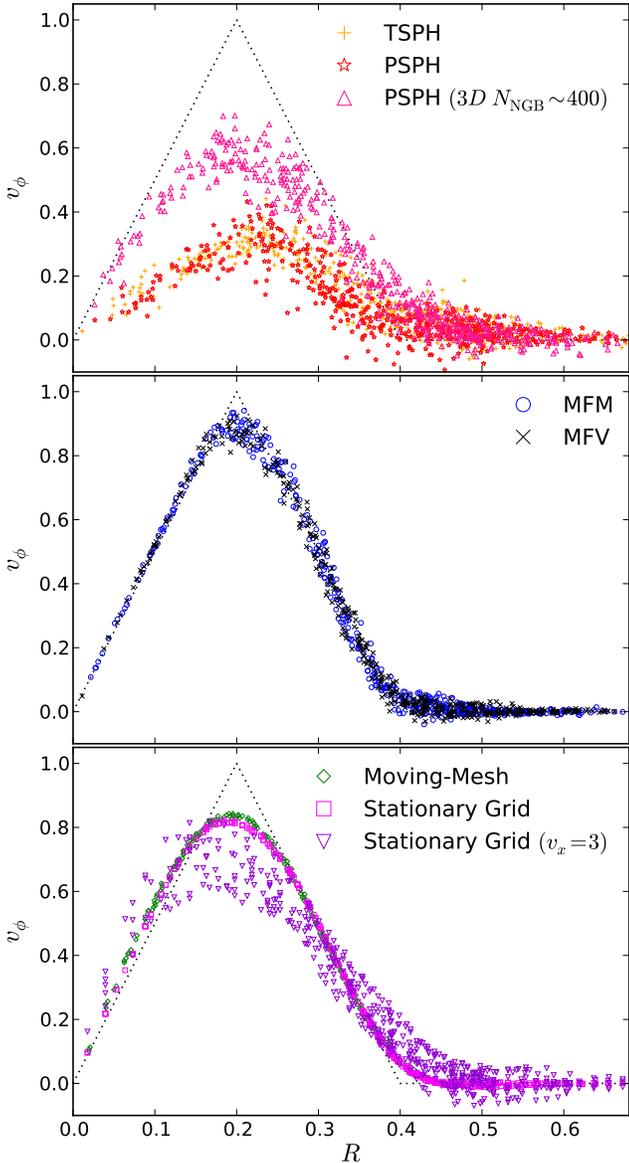
grid codes diffuse along all edges, moving-mesh, MFM, and MFV codes are machine-accurate).

domain  $0 < x < 1, 0 < y < 1$ , with zero radial velocity, pressure

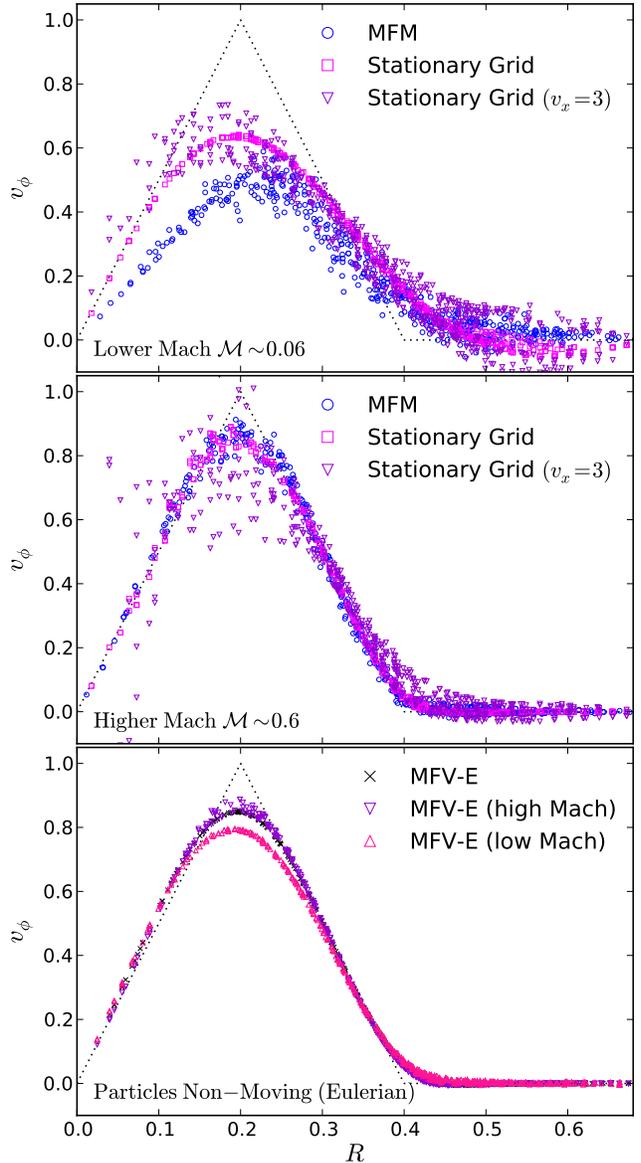
### 5.2.3 The Gresho Vortex: Sub-Sonic Turbulence & Angular Momentum

We next consider the triangular vortex of Gresho & Chan (1990). A two-dimensional gas with uniform  $\rho = 1$  is initialized in a periodic

$$P(R) = \begin{cases} 5 + 12.5R^2 & (0 \leq R < 0.2) \\ 9 + 12.5R^2 - 20R + 4 \ln(5R) & (0.2 \leq R < 0.4) \\ 3 + 4 \ln 2 & (R \geq 0.4) \end{cases} \quad (48)$$



**Figure 4.** The Gresho vortex (§ 5.2.3). 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  $\sim 1$  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:* MFM 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 MFM/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, MFM, MFV, and SPH results are machine-accuracy invariant to such boosts).



**Figure 5.** Gresho vortex as Fig. 4, but varying the Mach number of the vortex. We compare the MFM and fixed-grid methods at  $t = 3$  (with and without a moving vortex, for the fixed-grid results). MFV is very similar to MFM in all cases so is not shown. *Top:* A highly sub-sonic (rms Mach number  $\sim 0.06$ ) vortex (background pressure increased by  $P_0 = 50$ ). The accuracy of all methods degrades, but the effect is more severe in the MFM/MFV methods. *Middle:* A trans-sonic (rms Mach  $\sim 0.6$ ;  $P_0 = -5$ ) 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  $N_{\text{NGB}} = 400$ ) perform significantly worse than any other method. *Bottom:* Same tests, run with our MFV method but with the particle positions *fixed*, so the fluid is purely advected between particles (the method becomes an Eulerian stationary-finite volume method). Here even at very low Mach number, the noise is totally eliminated; this confirms that the code behaves like a stationary-grid code in the limit of zero particle motion, and the noise we see comes from the second-order “re-meshing” errors in the volume partition. This change introduces the same errors in advection and velocity-dependence of errors as fixed-grid methods, however (the examples here have no bulk fluid flow).

and azimuthal velocity

$$v_\phi(R) = \begin{cases} 5R & (0 \leq R < 0.2) \\ 2 - 5R & (0.2 \leq R < 0.4) \\ 0 & (R \geq 0.4) \end{cases} \quad (49)$$

where  $R^2 = x^2 + y^2$ . This represents a steady-state equilibrium vortex. We initialize the vortex with  $64^2$  elements. Fig. 4 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 number does help here; Read & Hayfield (2012) (see Fig. 5-6 there) and Dehnen & Aly (2012) (Fig. 9-10 therein) advocate going to  $N_{\text{NGB}} > 400$  neighbors in 3D. We have in fact repeated this test using the Wendland  $C^6$  kernel with  $N_{\text{NGB}} = 500$  or triangular kernel with  $N_{\text{NGB}} = 442$  (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  $\sim N_{\text{NGB}}^{-0.5}$ . In fact, the performance with both SPH methods even with  $N_{\text{NGB}} \sim 500$  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  $N_{\text{NGB}}^{-1/2}$ , so the CPU cost of the SPH computation for equivalent resolution scales something like  $\sim N_{\text{NGB}}^{3/2}$  – i.e. this improvement entails a factor  $> 50$  CPU cost in the SPH loops over the standard  $\sim 32$  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  $N_{\text{NGB}}$ , is slow ( $\sim N_{\text{ID}}^{-0.6}$ ).

The MFM and MFV methods show a tremendous improvement relative to SPH, despite using the simple cubic spline kernel with fixed  $N_{\text{NGB}} = 32$  in 3D ( $N_{\text{NGB}} = 16$  in 2D, following Gaburov & Nitadori 2011). The solution is less noisy than the SPH equivalent with  $N_{\text{NGB}} \gtrsim 500$ , 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 § B) leads to a smoother solution but much 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 MFM 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 (§ 5.2.4). The second modification is to simply set the problem in bulk motion. Springel (2010) 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  $v_{\text{bulk}} \gtrsim v_{\text{vortex}}$  (where  $v_{\text{vortex}} = 1$  is the peak vortex velocity here), the noise in the fixed-grid solution becomes very large. 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 MFM and MFV results (though still superior to the SPH results, until we reach vortex velocities  $\gtrsim 30 - 50$ ).<sup>18</sup>

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  $\sim v_{\text{vortex}}$ , for example, we require a resolution of  $\sim 256^2$  to achieve similar accuracy to a  $64^2$  simulation with the MFM 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  $\sim 100$ , then since the super-sonic cascade  $|\mathbf{v}^2(\lambda)|^{1/2} \propto \lambda^{1/2}$  (where  $\lambda$  is a parameter reflecting spatial scale; see Federrath et al. 2010), 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 the smallest eddy we wish to resolve should actually be  $\sim 32 - 256$  (depending on the desired accuracy) times larger along each axis than the grid scale! 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

<sup>18</sup> We have verified that what matters for these errors in stationary-grid codes is the ratio of bulk velocity to vortex velocity, not the sound speed or pressure. The errors which scale with the sound speed (discussed below) are almost entirely separable.

to rotations. The advantage over the MFV and MFM 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. 48 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 ( $P_0 < 0$  or  $P_0 > 0$ ) 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 Gresho & Chan (1990) 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 all methods, but the comparison between methods is similar (although some surprising noise of unclear origin appears in the ATHENA solution even without a velocity boost). However, the meshless methods (SPH, MFM, and 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  $\langle \mathcal{M}^2 \rangle^{1/2} \lesssim 0.2 - 0.3$ ; for MFM and MFV methods we see some vortex survive to  $t = 3$  down to  $\langle \mathcal{M}^2 \rangle^{1/2} \sim 0.03 - 0.05$ ; but with stationary grids we can reach  $\langle \mathcal{M}^2 \rangle^{1/2} \sim 0.001$  (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 MFM and MFV the convergence is closer to linear than second-order (also seen for AREPO; see Springel 2010, 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. Price 2012a; Bauer & Springel 2012; Vogelsberger et al. 2012); we confirm those results here. But while our MFM 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  $\sim 0.01$  (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.

To confirm that the major differences here are related to the volume partition and “re-meshing” error, we re-run our MFV simulations with each Mach number in Fig. 5, but with a fixed-particle (Eulerian) form of the code. Recall (§ 2.4), in the MFV form (with particle-particle mass fluxes), our method allows arbitrary particle velocities in principle – we do not have to set them to follow the fluid velocity in a Lagrangian manner (although this is our default choice). So here we re-run the method with the particle positions fixed; the fluid is entirely advected between particles, then, and the code is effectively a meshless, Eulerian stationary-grid method. At all Mach numbers, this totally eliminates the noise. Because particle “volumes” are conserved (since their relative positions do not change), there is no re-meshing error, and the code looks very sim-

ilar to the stationary-grid ATHENA results. This confirms that indeed, it is the re-meshing errors introducing noise, and that this method is, in principle, flexible and just as capable of capturing many advantages of fixed-grid codes if they would be desired for certain problems. However, we emphasize that this change means the method is no longer boost-invariant: if the fluid is in bulk motion, it must be advected constantly, and we see similar (in fact slightly larger, because the particle positions are not uniform along the velocity direction) noise as in the stationary-grid method. Likewise all the other advection and diffusion errors of fixed-grid codes pertain to this method as well.

#### 5.2.4 *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.

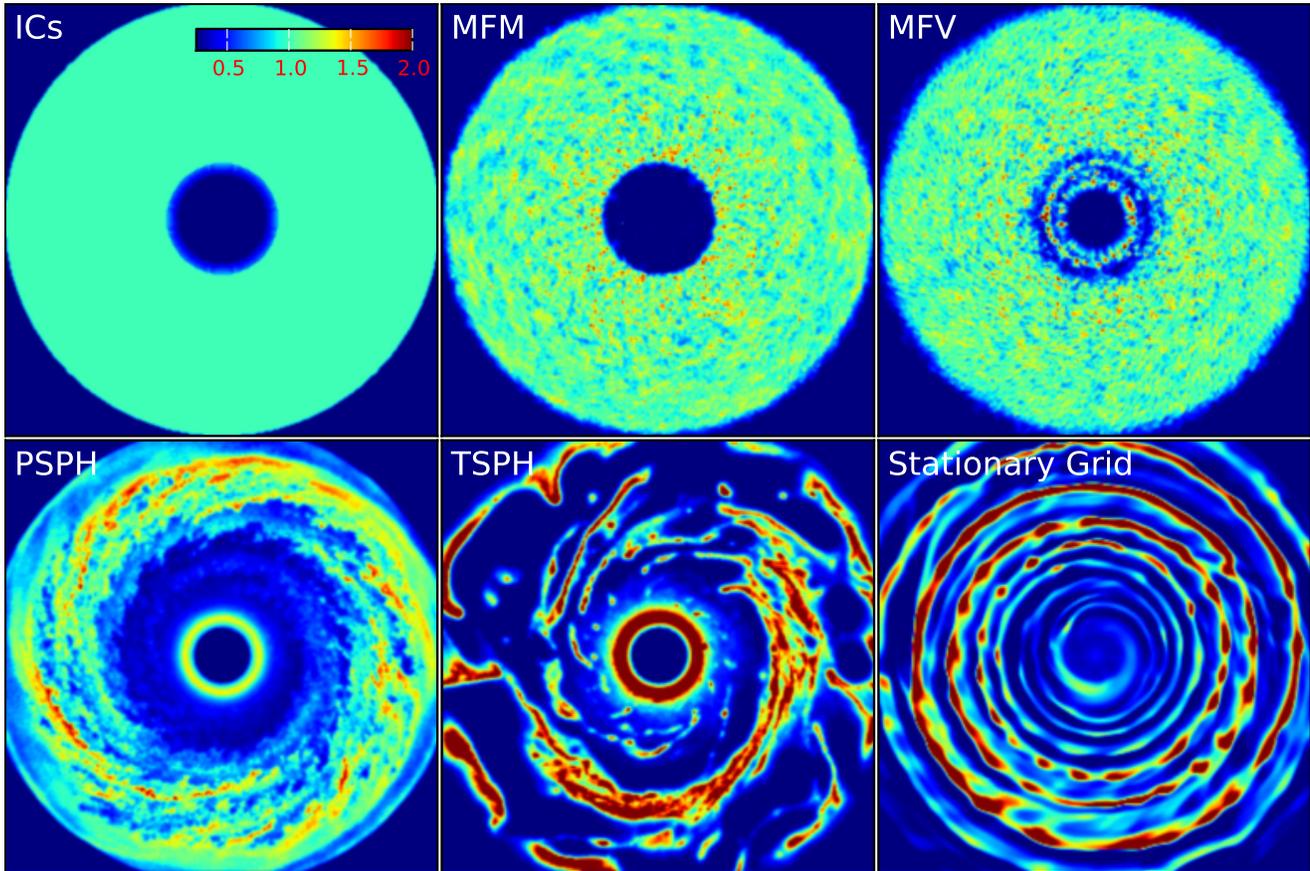
We initialize a razor thin  $\gamma = 5/3$  disk with constant surface density  $\Sigma = dM/dA = 1$ , vanishingly small pressure, and all gas on circular orbits about the origin subject to a Keplerian external acceleration  $\ddot{\mathbf{r}} = -\hat{r}GM_0/|\mathbf{r}|^2$  with  $G = M_0 = 1$  (and no self-gravity) and  $\mathbf{r}$  the vector from the origin. We focus on gas around the radius  $|\mathbf{r}| \approx 1$ .<sup>19</sup> 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, 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 MFM 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  $\sim 10$  orbits we have followed.<sup>20</sup> Of all the methods

<sup>19</sup> 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.

<sup>20</sup> As noted by Gaburov & Nitadori (2011), although the exact angular momentum conservation properties of the MFM 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



**Figure 6.** Keplerian disk problem (§ 5.2.4). *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; orbit time is  $= 2\pi r^{3/2}$ ), with effective  $256^3$  resolution (the plotted domain extends from  $-2 < x < 2$ ,  $-2 < y < 2$ ). 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  $t = 120$  ( $\sim 20$  orbits at  $r = 1$ ). *Top Middle:* MFM: 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  $\gg 100$  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  $\gtrsim 30 - 50$  orbits. *Bottom Left:* PSPH: Using a high-order artificial viscosity switch, shear viscosities are sufficiently suppressed to allow good evolution to  $\sim 5 - 8$  orbits, but the degradation is significant. *Bottom Center:* TSPH: Using “traditional” SPH artificial viscosity with a Balsara (1989) switch leads to far too much shear viscosity, and the disk undergoes the viscous instability and disrupts within  $\sim 2$  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  $\sim 1$  orbit.

we study, these appear to exhibit the lowest numerical viscosity in this specific problem (not necessarily in all problems!).

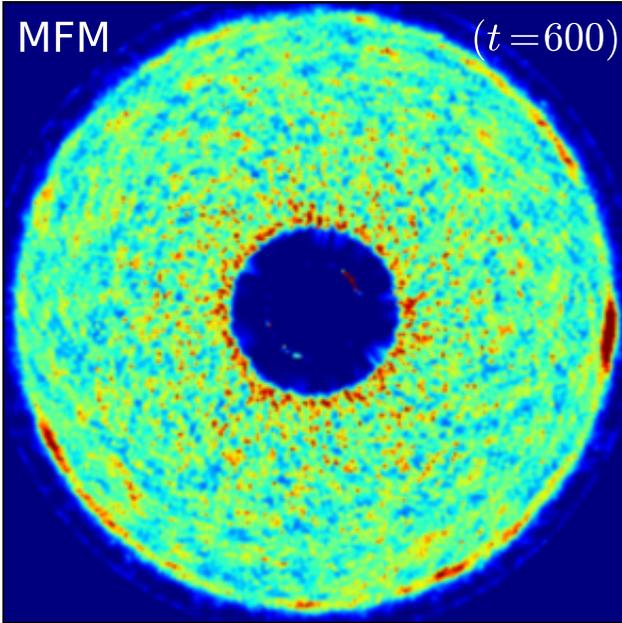
The MFV method generates some very small angular momentum errors, because there is a non-zero mass flux between particles; whenever this carries momentum flux not aligned with the line connecting the particle centers-of-mass, there can be weak violations. These begin to affect the disk evolution at  $\sim 30 - 50$  orbits; hence we see in Fig. 6 that the very inner edge of the disk has experienced some angular momentum evolution. The MFM 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 Riemann solver and accurate gradient estimator make this very low. Hence the angular momentum evolution is nearly perfect. In Fig. 7, we show the evo-

lution to time  $t = 600$ , or  $\sim 160$  orbits of the inner disk, and see the angular momentum conservation is still excellent! In fact, we have integrated as far as  $\sim 1000$  orbits, and found that the angular momentum conservation in our MFM method is nearly as good as that for collisionless test particles.

lution to time  $t = 600$ , or  $\sim 160$  orbits of the inner disk, and see the angular momentum conservation is still excellent! In fact, we have integrated as far as  $\sim 1000$  orbits, and found that the angular momentum conservation in our MFM method is nearly as good as that for collisionless test particles.

On the other hand, we see a rapid and catastrophic breakup of the disk within  $\sim 2$  orbits in our TSPH test. This is a well-known result (Maddison et al. 1996; Imaeda & Inutsuka 2002; Cullen & Dehnen 2010), and occurs because of the physical viscous instability (Lyubarskij et al. 1994), except that the disk is supposed to be inviscid! The problem is the “standard” SPH artificial viscosity produces far too much shear viscosity.<sup>21</sup>

<sup>21</sup> Specifically, our TSPH example uses the “standard” Gingold & Monaghan (1983) artificial viscosity with a Balsara (1989) 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). Cullen & Dehnen (2010) study several variations of SPH in this problem (there a Keplerian ring; see their Fig. 8), and confirm that



**Figure 7.** Keplerian disk as Fig. 6, at time  $t = 600$  (not a typo)! The inner ( $r \sim 0.5$ ) disk has executed  $> 250$  orbits, at this time, without decaying or disrupting.

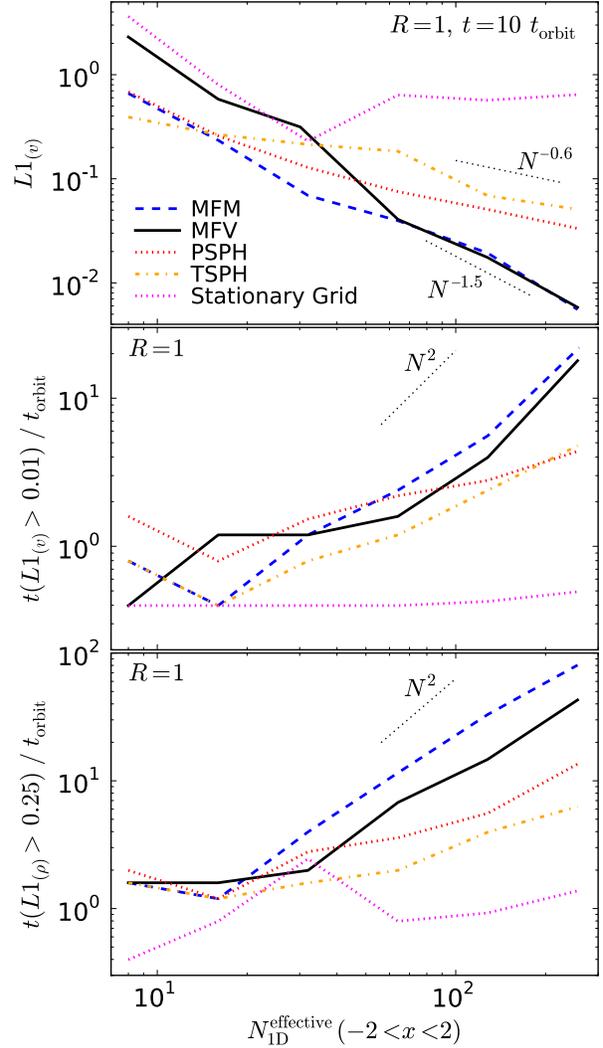
Our PSPH method uses an improved artificial viscosity switch proposed by Cullen & Dehnen (2010); this uses a least-squares matrix-base gradient estimator (similar to our MFM and MFV methods), which is zeroth-order accurate. This dramatically improves the results, allowing semi-stable evolution to  $\sim 5 - 10$  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.<sup>22</sup>

Moreover, as noted in Hu et al. (2014), 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.<sup>23</sup> Once any artificial viscosity appears,

the both the “standard” SPH artificial viscosity and the time-dependent viscosity method of Morris & Monaghan (1997), with or without the Balsara (1989) switch (e.g. methods in PHANTOM, GASOLINE, GADGET-2, and many other codes), undergo catastrophic fragmentation within  $\lesssim 2 - 3$  orbits.

<sup>22</sup> Cullen & Dehnen (2010), in their similar test problem (Fig. 8 therein), find that their method works well to  $\sim 5$  orbits, which we confirm, but we should note several differences between the test problem there and here. They use an effectively higher resolution and a carefully chosen initial particle distribution following Cartwright et al. (2009) 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.

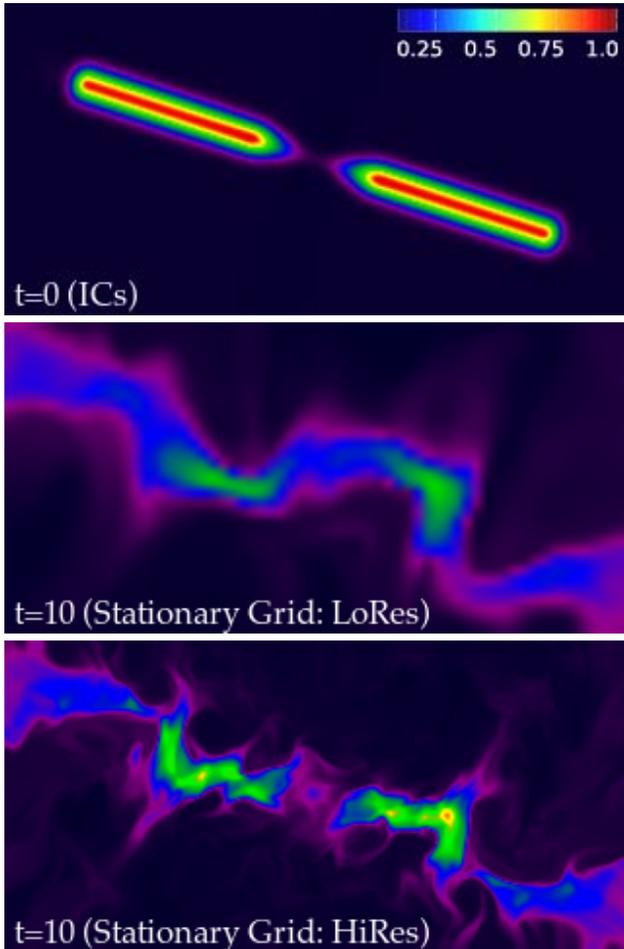
<sup>23</sup> The standard prescription for “damping” artificial viscosity in PSPH, in a supersonic disk, operates more slowly than the local dynamical time, hence the viscous instability can grow.



**Figure 8.** Convergence tests of the Keplerian disk problem in Fig. 6. *Top:*  $L1$  norm of the velocity error ( $L1_{(v)}$ ) at radius  $R \approx 1$  and time equal to 10 orbital times, as a function of resolution (number of elements on a side if the initial conditions formed a uniform grid from  $-2 < x < 2$ ,  $-2 < y < 2$ ). MFM and MFV methods converge as  $\sim N^{-1.5}$ . TSPH and PSPH converge more slowly, as  $\sim N^{-0.5}$ . The stationary grid result does not show convergence except at the lowest resolutions. *Middle:* Time (relative to the orbital time) at which the  $L1_{(v)}$  velocity norm at radius  $R = 1$  first exceeds a threshold of 0.01, as a function of resolution. This grows as  $\sim N^{1.8}$  in MFM and MFV methods; more slowly  $\sim N^{0.5}$  in TSPH/PSPH; and  $\sim N^{0.1}$  in the stationary grid result. *Bottom:* Time at which the  $L1_{(\rho)}$  norm of the gas density exceeds 0.25, as a function of resolution. This grows as  $\sim N^{1.8}$  in MFM/MFV;  $\sim N^{0.5-0.6}$  for SPH;  $\sim N^{0.2-0.4}$  for stationary grids. The higher threshold is chosen because small velocity errors in this problem lead to large density changes.

the viscous instability grows rapidly. Hu et al. (2014) 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 § 5.2.2, advection errors in non-moving grids are serious. We find (as have many others before) that these very



**Figure 9.** Keplerian disk as Fig. 6, but in 3D, with the thin disk rotated out of the  $x-y$  plane by an angle  $= \pi/20$ . We show the gas density in a slice through the  $x-z$  ( $y = 0$ ) plane (we show  $-2.3 < x < 2.3$ ,  $-1.15 < z < 1.15$ ). The Lagrangian particle-based methods (TSPH, PSPH, MFM, MFV) are invariant to such rotations, so we focus on the stationary-grid case. *Top:* The disk has constant height  $h = 0.1$  and is in equilibrium; it should be preserved at all times. *Middle:* Stationary-grid result at time  $t = 10$  (1.6 orbits at  $r = 1$ ), at lower resolution ( $64 \times 64 \times 32$ ). *Bottom:* Same, at higher resolution ( $256 \times 256 \times 128$ ). There is a strong grid-alignment effect (see § 5.2.4) whereby the disk is forced into alignment with the grid axes. This leads to more rapid angular momentum loss than in the exactly-aligned case in Fig. 6. It also produces an unphysical warp which becomes a “break” or “tear” in the disk as the alignment occurs first locally (i.e. the disk aligns at different heights) then globally.

quickly diffuse the disk, spreading the mass around and seriously distorting the shape of the disk before completely destroying its structure within  $\sim 2$  orbits. The inner parts lose angular momentum until they form a hot, hydrostatic center, and the outer parts are flung out. If we use a first-order solver, the central parts diffuse rapidly outwards; if we use a second or third-order solver (shown here), some regions move in and some move out, leading to “rings” forming which then get broken into clumps. In either case, total and local angular momentum are poorly conserved even over  $\sim 1$  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), as shown in Fig. 9. The MFM, 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 an unphysical large-scale warp in the disk on just  $\sim 1$  orbital timescale. Such grid alignment effects are well-known (e.g. Davis 1984). For example, (Hahn et al. 2010) 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  $5 \times 10^5$ ); Byerly et al. (2014) 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, run in the “simplest” initial configuration (an initial Cartesian mesh with outflow boundary conditions) the disk goes unstable and the angular momentum evolution tends to be corrupted within a few orbits. Unfortunately, some significant errors in angular momentum evolution are inevitable in moving-mesh codes, as has been discussed in e.g. Duffell & MacFadyen (2012) and Mocz et al. (2014). 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 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 (as in our MFV method). 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. This re-introduces some of the same errors as we saw with fixed grid/AMR methods. This means that the results for moving meshes are quite sensitive to choices like the mesh “stiffness,” regularization procedure, and in particular the choice of boundary conditions for the mesh-generating points (since the rigid Voronoi vol-

ume partition can lead to a “mesh tension” effect, whereby distortions in the central regions propagate outwards “through” the mesh). So there are ways to improve the situation, but like with AMR codes these generally depend on knowing the problem geometry ahead of time. For example, Duffell & MacFadyen (2012) design a moving grid which is a series of cylindrical shells free to rotate independently about a shared axis (the DISCO code); Muñoz et al. (2013) use a carefully-chosen initial grid configuration with a specially-designed boundary condition designed to prevent inward-propagation of “re-meshing”; these help considerably, but must be fine-tuned to the exact disk configuration.

Fig. 8 quantitatively compares the errors on this problem as a function of resolution and method. We define the  $L1$  norm as the mean absolute fractional deviation from the expected value in either velocity  $L1_{(v)} \equiv \langle |v_{\phi,i} - v_K[R_i]|/v_K[R_i] \rangle$  (where  $R_i$  is the radial position of particle  $i$ ,  $v_{\phi,i}$  is its azimuthal velocity, and  $v_K[R_i]$  is the Keplerian velocity at that radius) or density  $L1_{(\rho)} \equiv \langle |\rho_i - 1| \rangle = \langle |\Sigma_i - 1| \rangle$ . Because the orbits degrade on a timescale relative to the orbital time, it is useful to focus on a narrow radial annulus, here chosen to be  $0.8 < R < 1.2$  ( $R \sim 1$ ). For either  $L1$  norm, initially small errors tend to grow exponentially in all methods: if their growth timescale is related to the effective numerical viscosity and/or diffusion introduced by various methods, it should decrease with resolution. We therefore compare the  $L1_{(v)}$  norm in this annulus ( $R \approx 1$ ), at a fixed time (here  $t = 10t_{\text{orbit}}(R=1) = 20\pi$ ), for different “effective” resolution defined as above (number of elements across a side from  $-2 < x < 2$ , for a stationary grid; so our standard case is  $N_{1D}^{\text{eff}} = 256$ ). For MFM and MFV methods, the errors decrease as  $L1_{(v)} \propto N^{-1.5}$  (with, as expected, somewhat smaller errors systematically in the MFM case). This is somewhat slower than the ideal rate ( $N^{-2}$ ), but we should recall that we are well into the non-linear regime of the problem; that there are discontinuities (shocks), at which all methods are lower-order, introduced by even very small velocity perturbations because the disk is cold; and convergence as  $N^{-2}$  is not always guaranteed when external forces (here, gravity) dominate. Similar convergence rates are seen for the Gresho vortex test with these methods as well as moving-mesh and stationary-grid methods (see Springel 2010; Gaburov & Nitadori 2011). In both problems, for the MFM/MFV methods, the volume “remapping” errors tend to dominate, so this is not surprising. TSPH and PSPH do show convergence in Fig. 8, but much slower, as  $\sim N^{-0.6}$ ; again a similar scaling is seen in the Gresho problem (Dehnen & Aly 2012, Fig. 10). At much higher resolution, this should saturate at constant values because of the zeroth-order errors (unless we further increase the kernel size), but we estimate the resolution would need to be  $\gtrsim 2048^2$  before we reach this threshold. More strikingly, stationary grids show no real convergence here; except some small improvement compared to the lowest resolution. This is somewhat “noisy,” however; if we average over many radial annuli and times we see a weak convergence trend  $\sim N^{-(0.2-0.4)}$ . This is quite different from the Gresho test, which converges as  $\sim N^{-1.4}$  in stationary grids. Clearly the external forcing and large advection errors associated with the highly super-sonic, non-uniform flow are critical.

A related test is to ask how long we can evolve the disk before the  $L1$  norm exceeds some tolerance. We show this, for  $L1$  measured at  $R \sim 1$  as above, for both thresholds  $L1_{(v)} = 0.01$  and  $L1_{(\rho)} = 0.25$ . Note that, at similar times,  $L1_{(\rho)} \gg L1_{(v)}$ , because the problem setup is such that small velocity errors can lead to large density changes over many orbits. But otherwise the behavior with respect to both norms scales similarly. For MFM/MFV methods, we find the maximum time scales as  $\sim N^{1.8}$ , with MFM system-

atically reaching factor  $\sim 2$  longer times (reaching  $\sim 100$  orbits, or  $t > 600$ , for these thresholds at  $R = 1$ , consistent with Fig. 7). Note that some of the orbital differences we see in Fig. 6 with the MFV/MFM methods are not captured here, because they depend on smaller but more systematic differences. For TSPH/PSPH, we find the time increases as  $\sim N^{0.5-0.6}$ , with factor  $\sim 2-3$  larger times for PSPH vs. TSPH. And for stationary grids we see the time grow slowly, as  $\sim N^{0.1}$  in  $L1_{(v)}$  and  $\sim N^{0.2-0.4}$  in  $L1_{(\rho)}$ .<sup>24</sup>

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.

### 5.3 Shock and Non-Linear Jump Tests

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

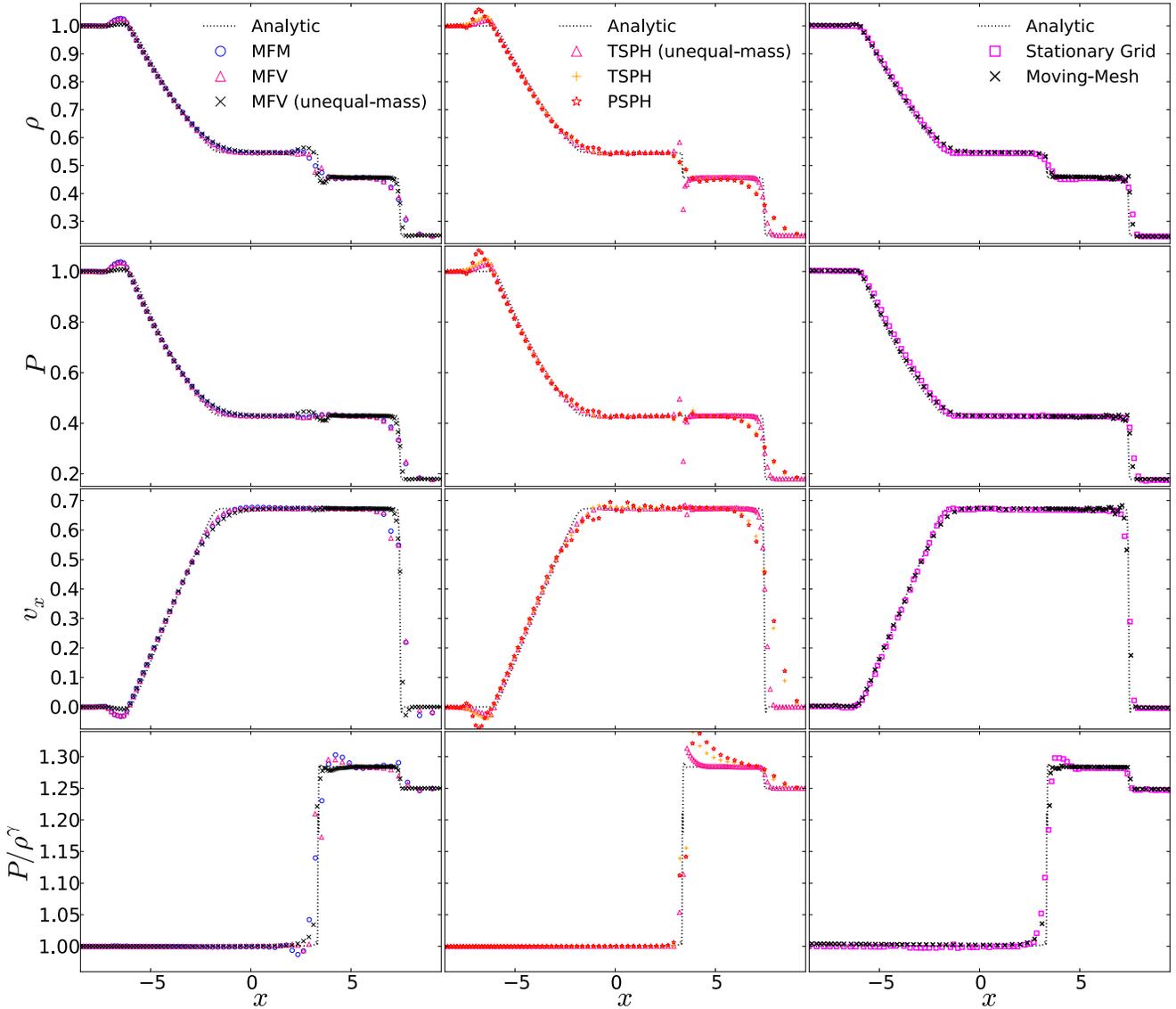
#### 5.3.1 Sod Shock Tube: A Basic Riemann Problem

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 (Hernquist & Katz 1989; Rasio & Shapiro 1991; Wadsley et al. 2004; Springel 2005, 2010). We intentionally consider a “low” resolution test, in which we place an initial 100 particles in the range  $-10 < x < 10$  (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  $\sim 4$  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 ( $x \sim 0$ ).

At the contact discontinuity ( $x \sim 3$ ), MFM 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

<sup>24</sup> Consider this in terms of the CPU cost required to evolve the (2D) problem to some time  $t$  with a given accuracy; this scales as  $\sim N^2 t / \Delta t$  where  $\Delta t$  is the timestep, which for a problem dominated by external forcing should scale as  $\Delta t \propto (\Delta x / |\mathbf{a}|)^{1/2} \propto N^{-1/2}$  (§ 2.7). This gives a scaling  $\sim N^{2.4}$  for MFM/MFV,  $\sim N^6$  for SPH,  $\sim N^{8.5}$  for cartesian grids.



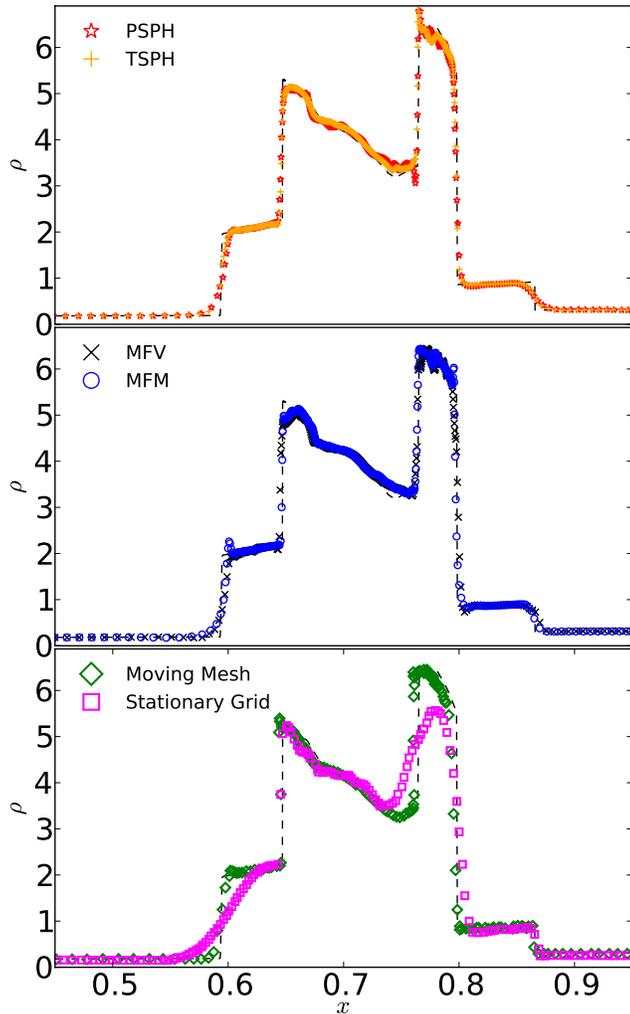
**Figure 10.** One-dimensional Sod shock tube § 5.3.1. From top to bottom, we plot density  $\rho$ , pressure  $P$ , velocity  $v_x$ , and entropy  $P/\rho^\gamma$ . We show the analytic solution (dotted) for each, compared to different methods (all with 100 resolution elements): all perform reasonably well with subtle differences. *Left:* MFM 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).

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 1) produce disastrous errors (easily order-of-magnitude deviations from the real solution, often with comparable particle-to-particle scatter) behind the shock front.

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 MFM 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 ( $\sim 2$  kernels). As noted in Springel

(2010), 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 MFM 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  $\sim 2$  times as many cells in the direction of motion. The particle and moving-mesh methods are invariant to rotations of the tube.



**Figure 11.** One-dimensional interacting blastwave test § 5.3.2. 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:* MFM & 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  $x \sim 0.75 - 0.8$ .

### 5.3.2 Interacting Blastwaves: Complicated Jumps & Extreme Riemann Problems in 1D

Another related one-dimensional test problem is the interaction of two strong blast waves, from Woodward & Colella (1984). 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 Riemann 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

§ 5.3.1, 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 extremely well, with sharper shock resolution (and no “blip”), especially around the narrow peak at  $x \sim 0.65$ , and they also capture the full under-density around  $x \sim 0.75$ .<sup>25</sup>

At this resolution, both the MFM and MFV methods give similar results. The MFM 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 between these methods and moving-meshes is that MFM/MFV do not capture the full density dip without going to higher resolution (perhaps surprising given SPH’s success, but this is where the fixed-grid method also has difficulty). But we confirm that at high resolution, the MFM and MFV methods converge to the same solution in good agreement with AREPO.

The largest errors at fixed resolution come from the fixed grid code. As noted in Springel (2010), 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 § 5.3.1, 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.

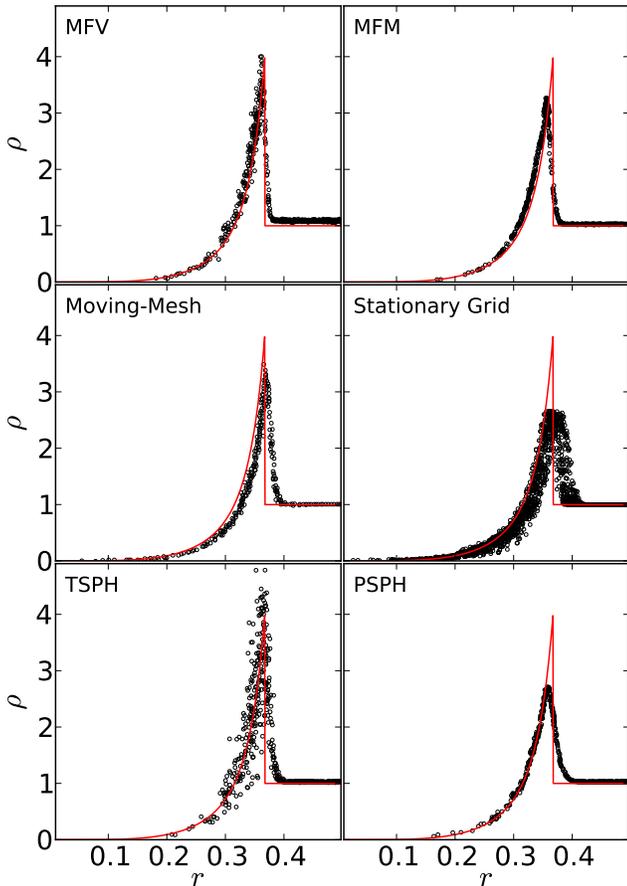
### 5.3.3 Sedov Blastwaves: Conservation, Integration Stability, & Symmetry

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 Saitoh & Makino 2009, 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 particle. We compare results at  $t = 0.06$ . A strong, spherically symmetric shock (of initially extremely high Mach number) should have formed, with a density jump of a factor  $(\gamma + 1)/(\gamma - 1) = 4$ .

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).

<sup>25</sup> One puzzling result is that, even at high-resolution, AREPO shows a slight offset in the position of the density jump at  $x \sim 0.8$ ; in contrast, MFV, MFM, SPH, fixed-grid (ATHENA), and a different moving-mesh simulation (using TESS) agree on the shock position. We suspect this has to do with either: a too-aggressive application of the entropy-energy switch (see § D) in AREPO (the switch does not trigger in our default runs with MFM/MFV, but if we modify to make the switch less conservative, we can reproduce the shock offset), or too-aggressive allowance in the code for adaptive timestepping (the pre-shock gas can have long timesteps, which lead to small offsets in time when they “become active,” hence an offset in shock position).

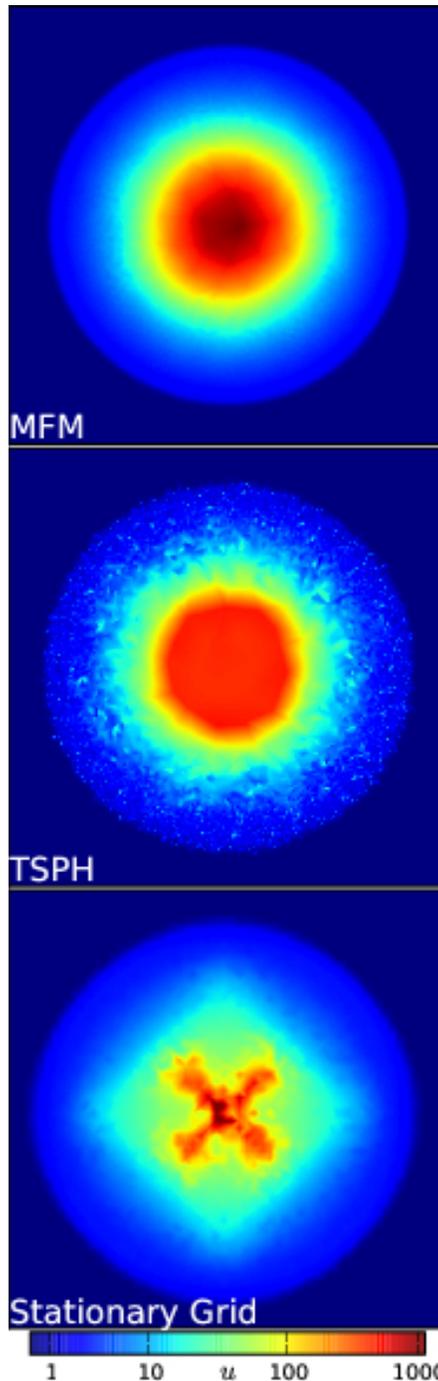


**Figure 12.** Three-dimensional Sedov-Taylor blastwave (§ 5.3.3). We plot the radial density profile at time  $t = 0.06$ ; each point is one particle/cell (for clarity we plot only a random subset of cells) at  $64^3$  resolution; red line is the analytic solution. *Top:* MFV & MFM solutions: the MFV shows excellent capturing of the shock jump, but is slightly noisier than MFM. *Middle:* Moving-mesh (AREPO) and stationary-mesh (ATHENA) solutions: the moving-mesh solution lies “in between” our MFM 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 spreads 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.

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

The MFM, MFV and moving mesh methods perform similarly well here. In all cases the jump is better captured (less “smeared”), giving a maximum density  $\sim 3.5$  (compared to the perfect case  $= 4$ ) instead of  $\sim 2.7$ . 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 (MFM, MFV, SPH) simply have no such instability because there is no preferred axis.

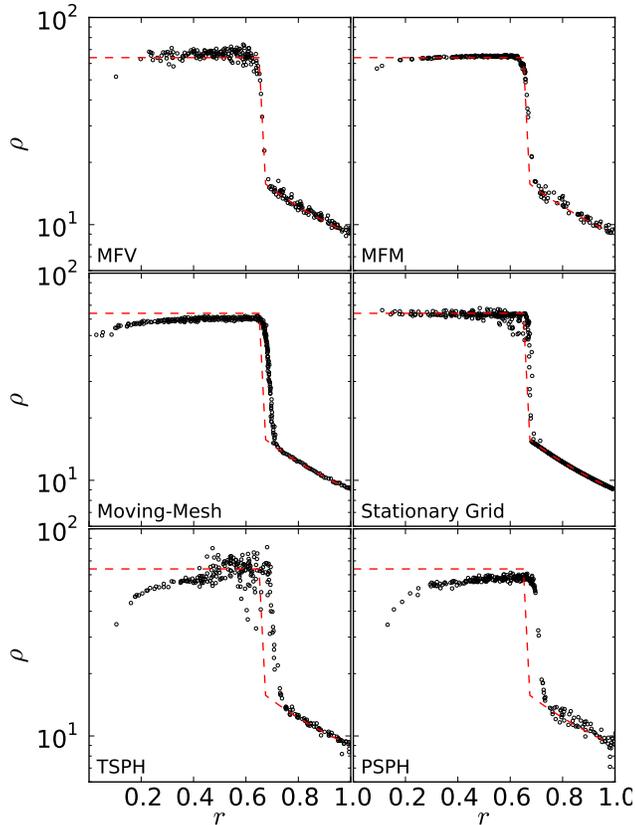
SPH methods generally do ok on this problem, except that the



**Figure 13.** Sedov blastwave from Fig. 12; here we plot the gas internal energy  $u$  (log-scaled) in a 2D slice ( $-0.45 < x, y < 0.45$ ) through  $z = 0$ , at  $t = 0.06$ . *Top:* MFM: 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).

shock is spread out further (see § 5.3.1) and they give noisy solutions in the post-shock behavior unless some additional diffusion is added.<sup>26</sup> 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-

<sup>26</sup> The noise arises from the E0 error when particles move through the shock.

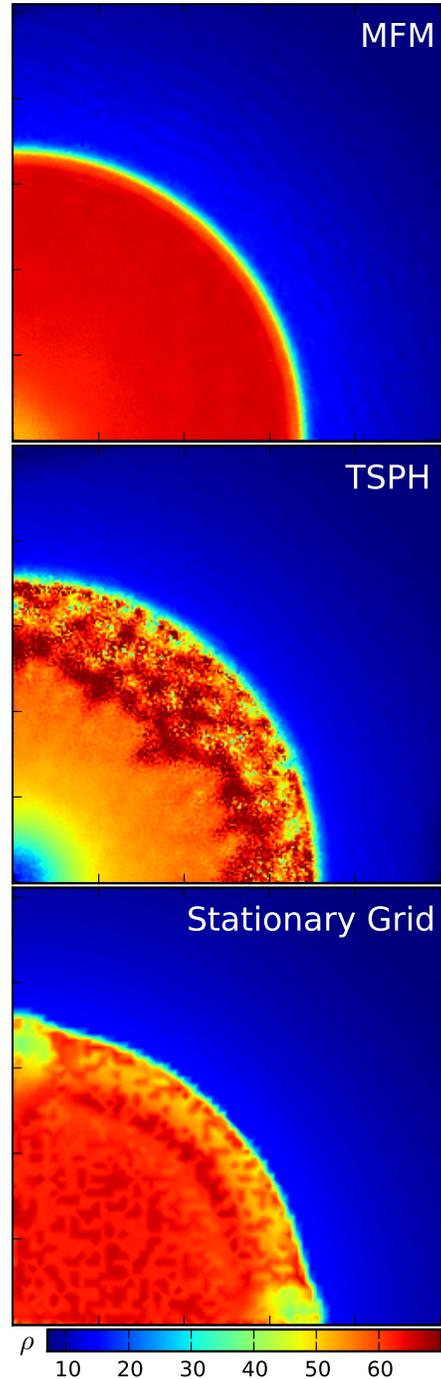


**Figure 14.** Three-dimensional Noh implosion problem (§ 5.3.4). We plot the radial density profile at time  $t = 2$ ; each point is one particle/cell (for clarity we plot only a random subset of cells) at  $50^3$  resolution; red line is the analytic solution. *Top:* MFV & MFM solutions: the MFV shows excellent capturing of the shock jump, but is noisier than MFM. *Middle:* Moving-mesh (AREPO) and stationary-mesh (ATHENA) solutions: the moving-mesh solution lies “in between” our MFM and MFV solutions in noise level, but the offset in shock position corresponds to a systematic underestimate 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.

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

A fairly extensive comparison of  $\sim 10$  different SPH variations for this problem is shown in Hopkins (2013) (Figs. 1-3 therein). As shown there, using a “consistent” (“corrected”) 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 Saitoh & Makino (2009); Durier & Dalla Vecchia (2012). 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.

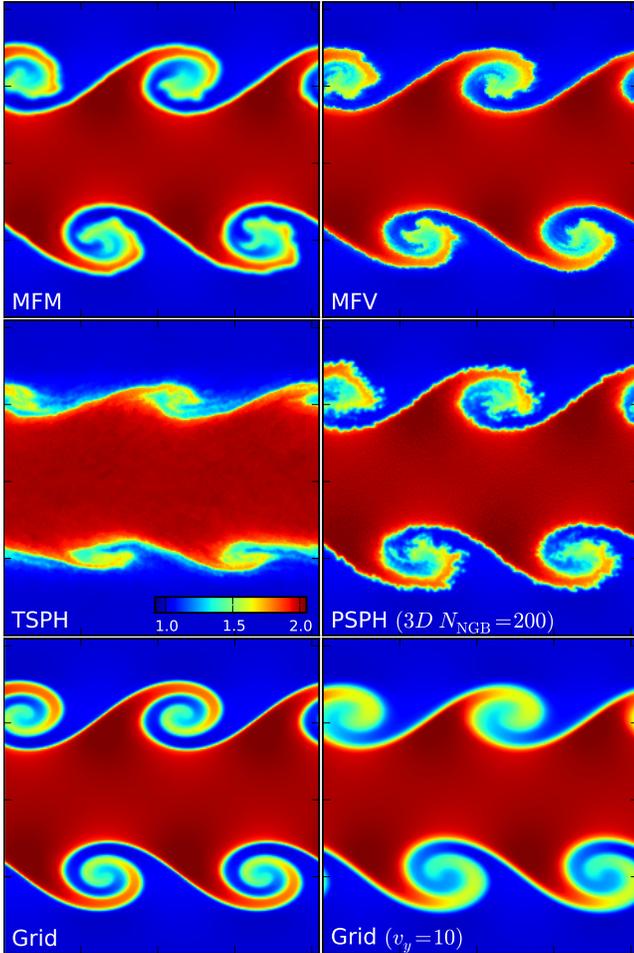
If we solve this problem in 2D the differences between methods are qualitatively identical, but slightly reduced in magnitude. A 1D analogue is essentially a Riemann problem (see § 5.3.1).



**Figure 15.** Noh implosion test from Fig. 14; we plot an image of the gas density, in a 2D slice (one quadrant:  $0 < x < 1, 0 < y < 1, z = 0$ ), at  $t = 2$ . *Top:* MFM: As in the Sedov test, the solution is smooth and shows good spherical symmetry. *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.

### 5.3.4 The Noh (Spherical Collapse) Test: Extreme Shock Jumps to Break Your Solver

Next consider the Noh (1987) test. This is a challenging test: many codes cannot run it without crashing, or run it but see catastrophic examples of the carbuncle instability. Liska & Wendroff (2003) noted only four of eight schemes they studied could actually run

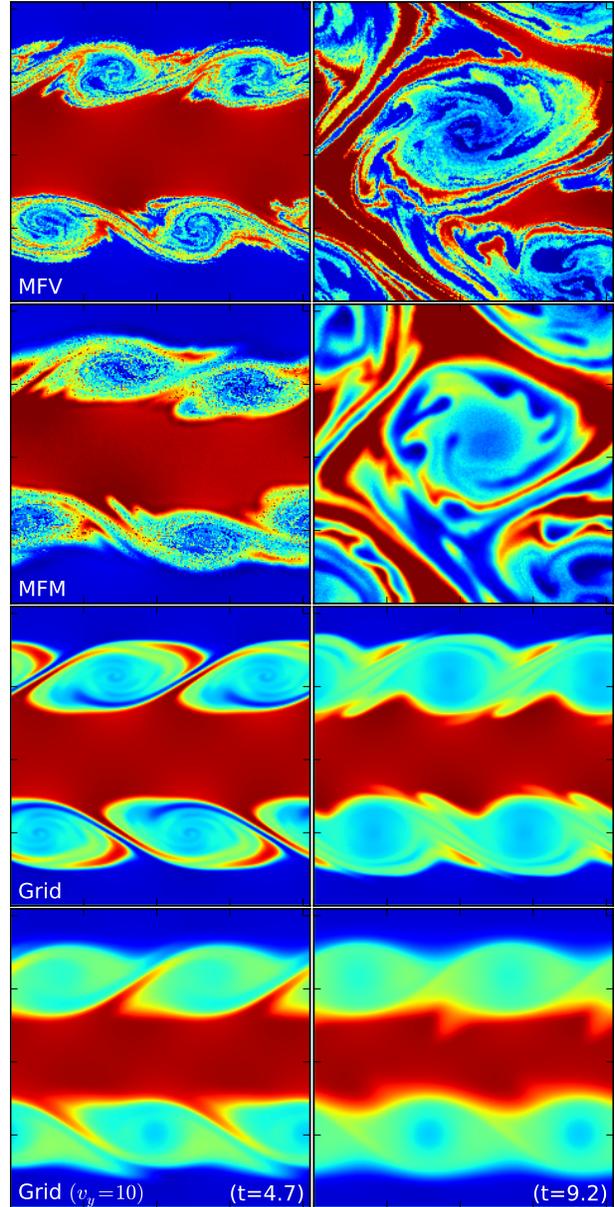


**Figure 16.** Kelvin-Helmholtz instability (§ 5.4.1). We compare the result of a 2D,  $256^2$  KH test problem at  $t = 2.1$ , where the rolls should be going non-linear. *Top:* In the MFM & MFV methods, the rolls are well-captured (with just the standard, small neighbor number, a 3D equivalent of  $N_{\text{NGB}} = 32$ ). 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).

the problem. An arbitrarily large domain<sup>27</sup> 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

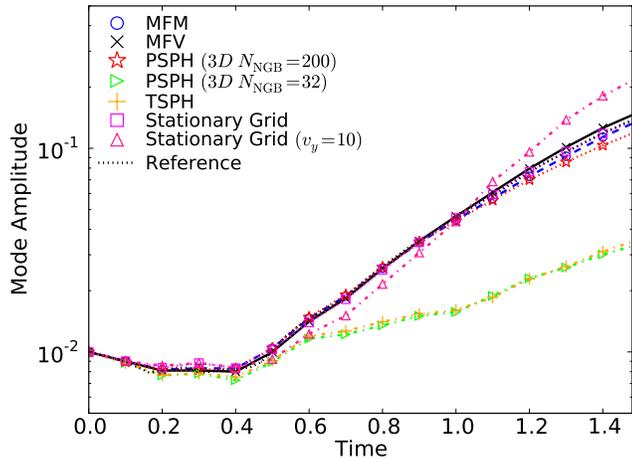
<sup>27</sup> For the particle codes, we simply use a huge domain so that we do not have to worry about boundary conditions. For the grid codes complicated explicit setting of inflow boundary conditions is possible and has been done here, but at fixed time it is identical to the result with a sufficiently large domain.



**Figure 17.** Non-linear evolution of the KH instability in Fig. 16, at  $t = 4.7$  and  $t = 9.2$ . In MFV (*top*) & MFM (*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.

that several approximate Riemann solvers can fail at the shock (requiring 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. The density profile is shown quantitatively in Fig. 14, and the spatial structure of the shock in Fig. 15.

The MFM and MFV methods give similar results here. The



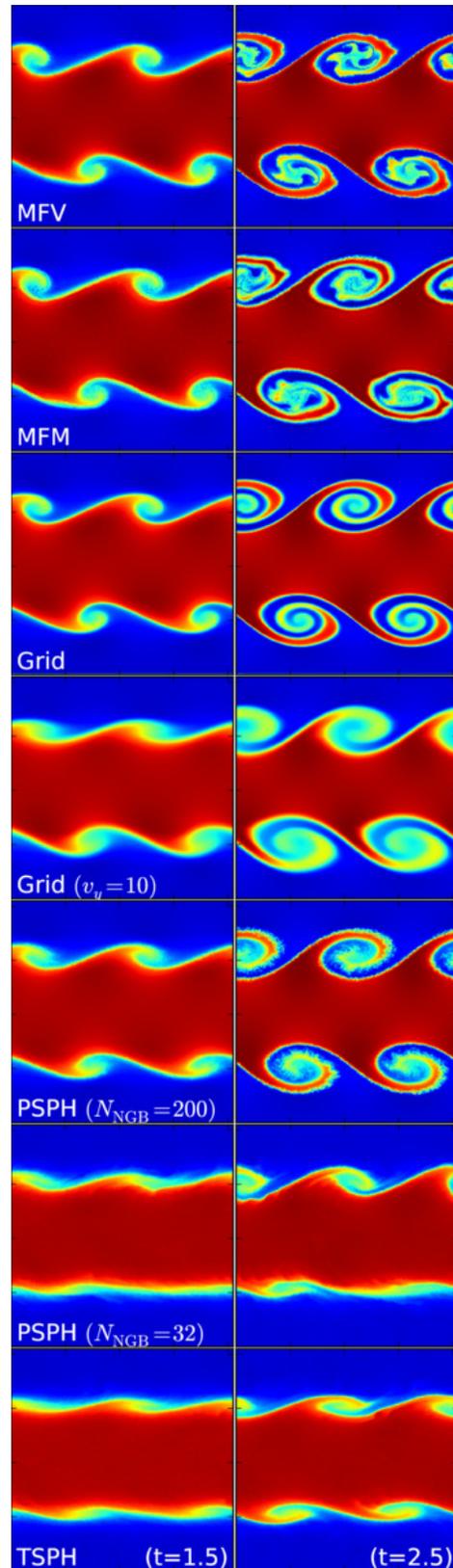
**Figure 18.** KH mode amplitude as a function of time for the  $256^2$  runs in Fig. 16. We compare a reference solution at  $4096^2$ , from McNally et al. (2012), which is well-converged when linear evolution still dominates ( $t \lesssim 1.5$ ). MFM, MFV, and stationary grid results (with no velocity boost) are essentially identical at these times, and well-converged. PSPH with high neighbor number is similar, though begins to depart at  $t \gtrsim 1.1$ . Mode growth is strongly suppressed in TSPH or PSPH with modest neighbor number (comparable to a grid calculation at  $\sim 32^2$  resolution). Grid results with a velocity boost converge more slowly (errors are similar to a first-order or  $\sim 50^2$  grid calculation).

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  $\rho$  owing to post-shock oscillations, but this is much weaker in the MFM result. The pre-shock  $\rho$  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 Liska & Wendroff 2003; Stone et al. 2008).

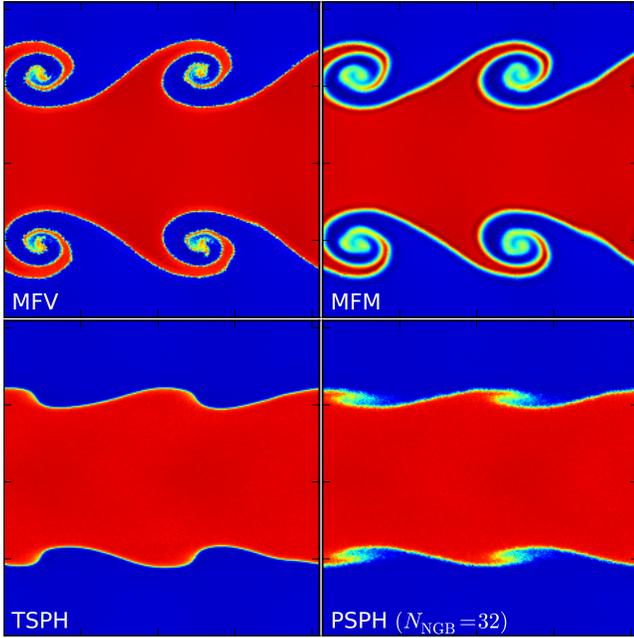
In the moving-mesh method, the noise level lies between our MFM and MFV methods. However some details appear slightly less accurate than our MFV or MFM calculation. The jump with moving-meshes is slightly under-estimated; this does eventually converge to the correct jump but requires somewhat higher resolution. As we saw with the Sedov test, the shock position is slightly offset (leading the analytic solution); we suspect this owes more to the timestepping scheme than the numerical method. And the wall-heating is noticeably more severe than in the MFM or MFV methods.

In the fixed-grid code, the carbuncle instability is particularly prominent – this actually seeds most of the noise around the jump.<sup>28</sup> 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.

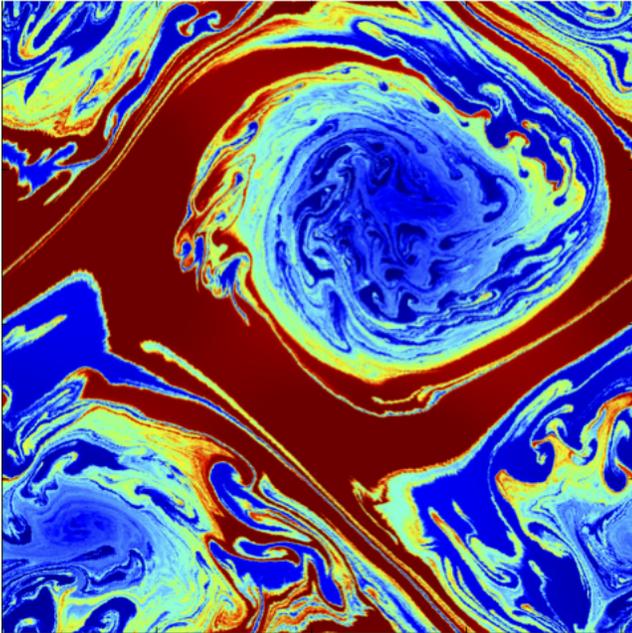
<sup>28</sup> Note that we have run this with the “standard” version of ATHENA, which is very similar to AREPO in “fixed grid” mode, and gives similar results at fixed resolution to AMR codes like RAMSES (which we have also compared), FLASH, and PLUTO. As noted in Stone et al. (2008), this can be cured 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.



**Figure 19.** Evolution of a 3D ( $256^2 \times 16$ ) version of the KH instability from Figs. 16-17 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  $N_{\text{NGB}}$  (shown explicitly) still fails here. Also note that the early-time (linear and early non-linear) growth is nearly identical in MFV, MFM, and stationary-grid calculations (though the grid result degrades when “boosted”); only later into the non-linear evolution do we see the differences from Figs. 16-17.



**Figure 20.** Alternative 3D ( $256^2 \times 16$ ) 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  $t = 3.75 \approx 1.1 \tau_{KH}$ . *Top:* MFV & MFM 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 MFM method smears the fluid phase boundaries slightly; the MFV method preserves a sharp contrast. *Bottom:* SPH results with the same neighbor number ( $N_{NGB} = 32$ ); both TSPH and PSPH fail to capture the instability in this case.



**Figure 21.** 2D KH instability at high-resolution ( $1024^2$ ) with the MFV method, at time  $t = 10$ .

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 correct jump condition.

Finally, if we consider the 2D version of this problem, as in § 5.3.3, 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.

#### 5.4 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. Morris 1996; Dils 1999; Ritchie & Thomas 2001; Marri & White 2003; Okamoto et al. 2003; Agertz et al. 2007; Kitsionas et al. 2009; Price & Federrath 2010; Bauer & Springel 2012; Sijacki et al. 2012). And in response many improvements have been made to SPH, which allow it to better handle such instabilities (see Monaghan 1997a; Ritchie & Thomas 2001; Price 2008; Wadsley et al. 2008; Read et al. 2010; Read & Hayfield 2012; Abel 2011; García-Senz et al. 2012; Saitoh & Makino 2013; Hopkins 2013; Valdarnini 2012). However, as pointed out in Springel (2010), 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 velocity-dependent truncation 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 even wipe out the instabilities entirely at low resolution. We therefore consider these in more detail below.

##### 5.4.1 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 McNally et al. (2012). 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

$$\rho(y) = \begin{cases} \rho_2 - \Delta\rho \exp[(y-0.25)/\Delta y] & (0 \leq y < 0.25) \\ \rho_1 + \Delta\rho \exp[(0.25-y)/\Delta y] & (0.25 \leq y \leq 0.5) \\ \rho_1 + \Delta\rho \exp[(y-0.75)/\Delta y] & (0.5 \leq y \leq 0.75) \\ \rho_2 - \Delta\rho \exp[(0.75-y)/\Delta y] & (0.75 < y \leq 1) \end{cases} \quad (50)$$

$$v_x(y) = \begin{cases} -0.5 + 0.5 \exp[(y-0.25)/\Delta y] & (0 \leq y < 0.25) \\ 0.5 - 0.5 \exp[(0.25-y)/\Delta y] & (0.25 \leq y \leq 0.5) \\ 0.5 - 0.5 \exp[(y-0.75)/\Delta y] & (0.5 \leq y \leq 0.75) \\ -0.5 + 0.5 \exp[(0.75-y)/\Delta y] & (0.75 < y \leq 1) \end{cases} \quad (51)$$

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:

$$v_y(x) = \delta v_y^0 \sin(4\pi x) \quad (52)$$

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

$$\tau_{\text{KH}} \equiv \frac{\lambda(\rho_1 + \rho_2)}{(\rho_1 \rho_2)^{1/2} |v_{x,1} - v_{x,2}|} \quad (53)$$

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

Fig. 16 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 MFM, MFV, moving-mesh, and fixed-grid codes (we have compared quantitatively with the linear-growth curves in McNally et al. 2012, and find all these methods behave similarly; see also Fig. 19). 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. 17, where we compare later times. In Fig 18, we quantitatively compare the amplitude of the y-velocity perturbation in the early (linear) phase, where we define the amplitude following McNally et al. (2012) (their Eq. 6-13), and compare to the converged reference solution therein at  $4096^2$  resolution.

In the MFM and MFV methods, the whorl height and linear growth 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. 21 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 MFM method necessarily smooths the density field over  $\sim 1$  smoothing length. This leads to less-detailed small-scale structure in the MFM method, and in the non-linear phase to enhanced diffusion. However the behavior on large scales is similar – i.e. the MFM solution, even late into the non-linear phase, resembles a “smoothed” MFV solution, rather than departing from it. This is important since it demonstrates the second-order advection errors in the MFM 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 noted by Springel (2010), 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, even in the early (linear) phase, where the errors are comparable to those from a lower-order method or a much lower-resolution simulation. 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 (§ 5.2.2): 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  $N_{\text{NGB}}$  as used for the MFV and MFM kernels, or as used in traditional SPH work, then we find in Fig. 23 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 begin 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. 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 Hopkins (2013) (Fig. 6 there); however, this comes at the cost of serious noise in 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. 19, 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  $256 \times 256 \times 16$  periodic box. Here we see essentially identical qualitative behavior, as expected. We explicitly show the earlier stages of the runs, to demonstrate again that the linear mode growth is identical in MFM, MFV, and stationary-grid methods (when the fluid has no net velocity). The transition to 3D causes no problems for either MFM or MFV methods (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. If the fluid is boosted in the stationary-grid method, we see the linear-phase mode growth is artificially suppressed (the whorls have not reached the same height at this resolution), diffusion is increased (especially at later times), and the symmetry is broken (the “upper” set of rolls now differ in amplitude from the “lower” set). PSPH is able to do reasonably well with large neighbor numbers; although the linear-phase growth is slightly slower than the converged solution from MFM/MFV/non-boosted grid methods, it is close, and the late-time solution looks reasonable. However, once again, with small neighbor number, both TSPH and PSPH fail to form rolls properly.

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<sup>29</sup> and described in Agertz et al. (2007); Read et al. (2010). 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  $\rho_2 = 0.5\rho_1, T_2 = 2.0T_1, v_2 = -v_1 = 40 \text{ km s}^{-1}$ . 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  $\delta v_y = 4 \text{ km s}^{-1}$  and wavelength  $\lambda = 128 \text{ kpc}$ .

As expected from the previous tests, both MFM 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 McNally et al. 2012 IC is optimized for grid codes), and as a result we directly see that the symmetry in the MFV and MFM 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  $\sim 1$  softening in the MFM method (the green “edge”; still much less 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.<sup>30</sup>

#### 5.4.2 Rayleigh-Taylor Instabilities

We now consider the Rayleigh-Taylor (RT) instability, with initial conditions from Abel (2011). 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. 22 we show the evolution of the instability in a high-resolution ( $512 \times 1024$ ) 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, MFM, 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 Springel (2010), 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 exact symmetry in these methods is to use a very specific and usually pathological initial particle distribution.<sup>31</sup>

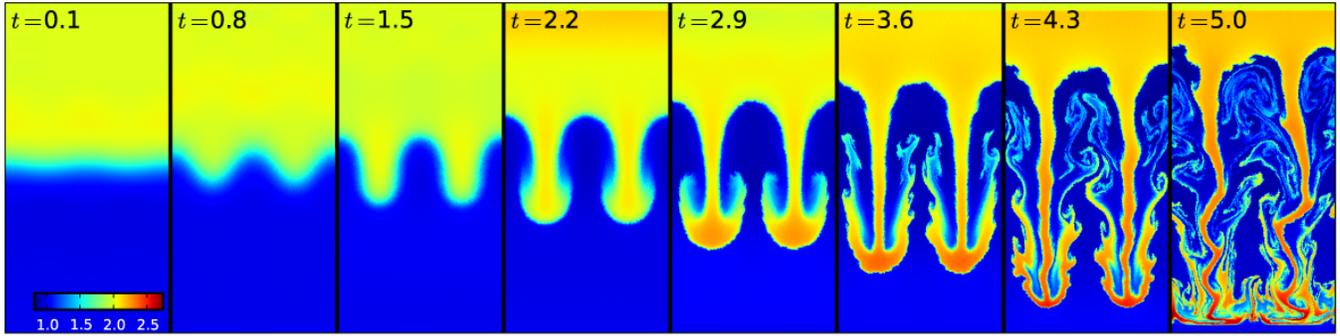
Fig. 23 compares the non-linear RT evolution across different methods, with the same initial conditions at medium resolution ( $128 \times 256$ ). The MFV and moving-mesh methods capture the most small-scale structure: this is because they are both Lagrangian and can follow contact discontinuities very sharply. The large-scale evolution of the MFM 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.

We see the same problems with SPH as in the KH test: at low neighbor number, E0 errors and surface tension (in TSPH) suppress the growth of the instability entirely, and even in PSPH we require a 3D-equivalent  $N_{\text{NGB}} \gtrsim 128$  to see good linear growth. As in the KH problem, conductivity in PSPH helps the mode initially grow but corrupts the non-linear structure of small-scale KH modes (here the problem looks better without conductivity as shown in Fig. 11 of Hopkins 2013, but as noted above this leads to excessive noise in other problems).

<sup>31</sup> We do see here and in the KH tests that the MFV and MFM methods appear to preserve symmetry longer in time and more accurately than moving-meshes (compare Figs. 35-36 in Springel 2010). For the MFV and MFM methods, it is easily verified that the numerical equations are manifestly symmetry-preserving (provided the problem setup and initial particle distribution are symmetric). The growth of asymmetry in symmetric ICs stems purely from roundoff errors. In moving-mesh codes, however, the fact that mesh boundaries are “sharp” means that when cells are sufficiently deformed, they must eventually dis-continuously change their connectivity in a manner that does not necessarily preserve symmetry. This leads to a sort of “mesh tension” or “mesh bending” instability discussed in Springel (2010).

<sup>29</sup> Available at <http://www.astrosim.net/code/doku.php>

<sup>30</sup> In Hopkins (2013) Figs. 8-9, we showed that PSPH was able to capture at least some “whorl” structure using a very similar IC to the Wengen IC, still using a simple cubic spline with  $\sim 32$  neighbors, if the initial seed mode amplitude was larger ( $\sim 10\%$ , as opposed to  $\sim 1\%$ ). This is because the E0 errors were then smaller than the seed mode.



**Figure 22.** Rayleigh-Taylor instability test (§ 5.4.2). 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, MFM, 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 MFM 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.

If the fluid is not moving with respect to the grid, a stationary-grid method 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, MFM, 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.

#### 5.4.3 The “Blob” Test: KH & RT Instabilities in a Supersonic, Astrophysical Situation

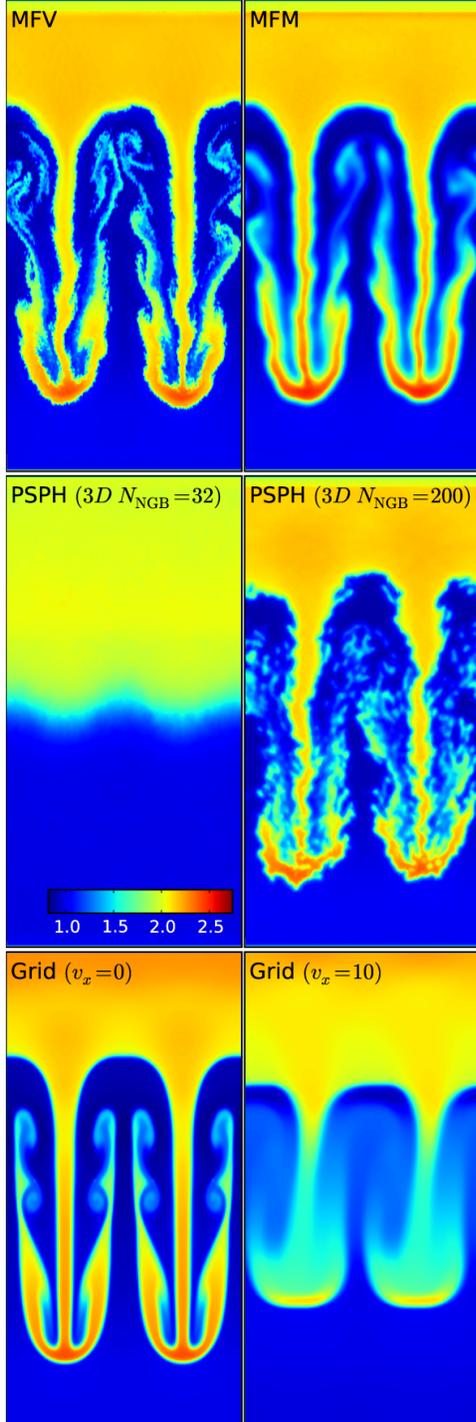
Next we consider the “blob” test, which is designed to synthesize the fluid mixing instabilities above (as well as ram-pressure stripping) in a more “realistic” example of astrophysical interest representative of a multi-phase medium. The initial conditions come from the Wengen test suite and are described in Agertz et al. (2007): we initialize a spherical cloud of uniform density in pressure equilibrium with an ambient medium, in a wind-tunnel with periodic 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;  $9.6 \times 10^6$  particles/cells are initialized in a lattice (with equal-masses in the particle-based methods).

Fig. 24 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 MFM and MFV results (see also Gaburov & Nitadori 2011, Fig. 7-8, who find the same with their implementation of an MFV-like scheme), and in grid-based codes including moving meshes (Sijacki et al. 2012, Figs. 4-5), fixed Cartesian grids, and AMR schemes (see Agertz et al. 2007, Figs. 4-10). Note in particular the good agreement between MFV and MFM results for the small-scale structure of the shredded cloud and the sharp capturing of the shock fronts.

Quantitatively, Fig. 25 follows Agertz et al. (2007) 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  $\rho_c$  and  $T_a$  are the initial cloud density and ambient temperature). We compare our results here with a compilation from other methods in Agertz et al. (2007). For a stationary-grid result, we use the published result from ENZO (an AMR code), run with an effective resolution about equal to our runs here. The MFM, 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 “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 PSPH (Saitoh & Makino 2013; Hopkins 2013). 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. Wadsley et al. 2008); so the early-time behavior agrees well with the MFM, 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



**Figure 23.** RT instability as Fig. 22 with different methods, at medium-resolution ( $128 \times 256$ ) & time  $t = 4$ . *Top Left:* MFV. Secondary instabilities are sharply resolved (as Fig. 22), even at lower resolution. *Top Right:* MFM. The evolution is similar to MFV, but contact discontinuities are not as sharply resolved. *Middle Left:* PSPH, with same neighbor number as MFM/MFV runs. No instability develops, despite other improvements over TSPH, because E0 errors swamp the mode growth. *Middle Right:* PSPH, with a higher-order kernel and increased neighbor number; this reduces E0 errors allowing the mode to grow. However 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 gives sharply-defined features and excellent symmetry. *Bottom Right:* Stationary-grid, with a bulk velocity  $v_x = 10$ . Velocity-dependent advection errors (most severe in Eulerian methods) severely affect the symmetry and accuracy of the solution.

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.

#### 5.4.4 Driven Turbulence: Sub-Sonic & Super-Sonic Limits

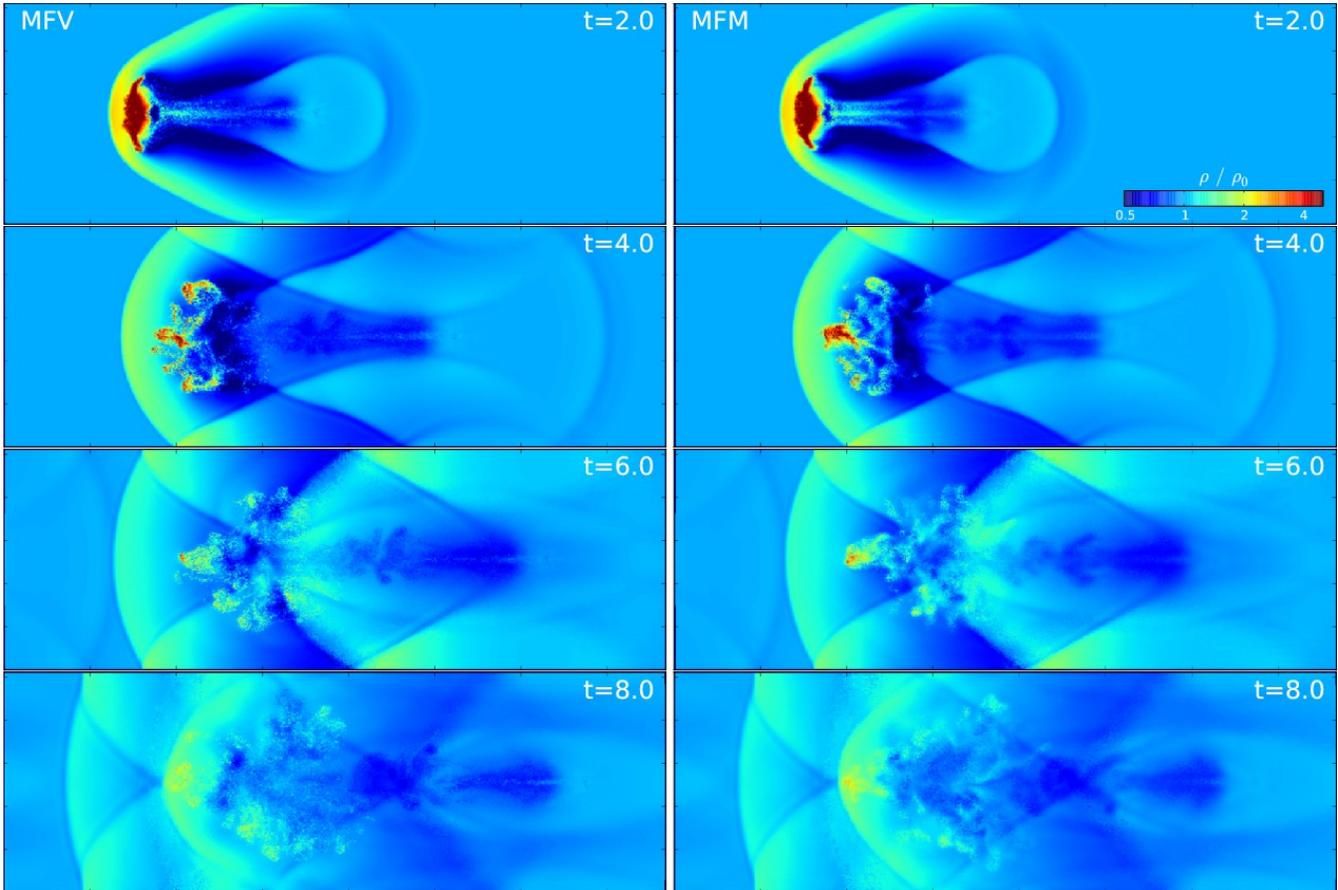
We next consider tests of driven, isothermal turbulence in a periodic box, in both the super-sonic and sub-sonic limits. This tests the numerical accuracy, convergence, shock-capturing, stability, and effective resolution of different methods, in a context directly relevant for almost all astrophysical problems.

The turbulent driving routines are implemented here in an identical manner to Bauer & Springel (2012). Briefly, a periodic box of unit length  $L = 1$ , density  $\rho = 1$ , sound speed  $c_s = 1$ , and isothermal equation of state  $\gamma = 1$  is stirred via the usual method in e.g. Schmidt et al. (2008); Federrath et al. (2008); Price & Federrath (2010), where a small range of modes corresponding to wavelengths between  $1/2 - 1$  times the box size are driven in Fourier space as an Ornstein-Uhlenbeck process, with the compressive part of the acceleration projected out via a Helmholtz decomposition in Fourier space so that the driving is purely incompressible/solenoidal (most appropriate for sub-sonic turbulence). We use identical parameters to Bauer & Springel (2012), Table 4 for the driving, and consider two cases: a sub-sonic case where the driving is set such that the box maintains a quasi-steady-state rms Mach number  $\mathcal{M} \sim 0.3$  and a super-sonic case with rms  $\mathcal{M} \sim 8.4$ .

First, we consider the sub-sonic case, since Bauer & Springel (2012) and others have noted this is more challenging for methods like SPH. Fig. 26 shows an image of the turbulent velocity, density, and vorticity/enstrophy fields, after the turbulence has reached a steady state ( $t \gtrsim 5$ ); the image is based on a tri-linear interpolation of the particle field values from the nearest neighbors to a slice at the midplane of the  $z$ -axis. This can be compared to the similar Fig. 4 in Bauer & Springel (2012), which compares moving-mesh (AREPO), fixed-grid, and GADGET-2 SPH results for the same setup. Fig. 27 compares the different methods quantitatively; we measure the velocity power spectra (following exactly the power-spectrum definition in Bauer & Springel 2012 for all methods) and show them as a function of methodology and resolution.

The results here from our MFV and MFM methods are very similar to the moving-mesh and stationary grid methods (both visually and quantitatively).<sup>32</sup> In particular, we note the striking amount of small-scale structure which can be seen in the vorticity and velocity fields (Fig. 26), and similarity in the predictions for the power spectrum (Fig. 27). MFV and MFM methods give essentially indistinguishable results here, because the density gradients and associated mass fluxes in the MFV method are very weak. For the same reason, Bauer & Springel (2012) found very little difference between moving-mesh and stationary-grid methods. In the power spectrum, all of these methods exhibit a similar “bottleneck” (the well-known feature whereby the deficit of physical viscosity leads to some excess power on scales just above the dissipation range)

<sup>32</sup> The larger apparent differences seen in the sub-sonic turbulence in the KH problem (§ 5.4.1) clearly relate to advection of the contact discontinuities and strong density gradients, not the maintenance of vorticity. The noise seen in the Gresho problem (§ 5.2.3) in the particle-based methods as compared to the moving-mesh methods does not seem to be a problem here; it is small compared to the net circulation of the vortices, and furthermore they do not typically survive as long as the test problem there is run.



**Figure 24.** The “blob” test (§ 5.4.3). We plot density (in units of the initial ambient density of the background  $\rho_0$ ) in a 2D slice ( $0 < y < 6000$ ,  $0 < x < 2000$ ,  $z = 0$ ) through the blob center, for the MFV and MFM runs at different code times (labeled). An initially dense, cold spherical cloud in pressure equilibrium is hit by a wind tunnel (moving left-to-right). The wind-cloud collision generates a bow shock and rapidly disrupts the cloud via RT and KH instabilities at the interface. We see good agreement between MFM/MFV methods; the cloud is rapidly “shredded” in both, and shocks are sharply-captured.

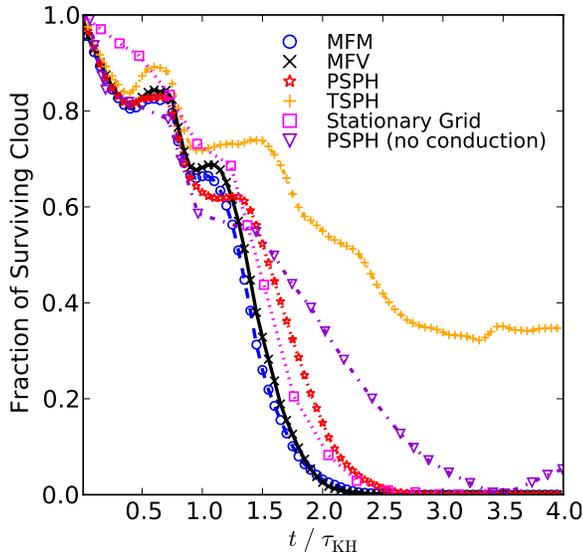
with a dropoff in power on the smallest scales. The only differences appear on very small scales near the Nyquist frequency (twice the inter-particle spacing, i.e. on scales smaller than the “bottleneck”); the particle and moving-mesh methods show some upturn of power here, but as pointed out by Price (2012a), this is dependent on how one defines the power spectrum and interpolates values for the Fourier transform (if we, for example, interpolate the particle-valued velocities onto a regular lattice, then perform the FFT, the feature goes away and the MFM and MFV methods look like the stationary grid result down to the Nyquist frequency). In any case, this all occurs below the scale where the cascade is no longer captured, so is not physically meaningful. All of these methods also show similar convergence; Fig. 27 shows this explicitly for MFV and moving-mesh methods but the results are again identical for MFM and stationary grids. Increasing the resolution directly translates to a larger dynamic range in the cascade; if we retain scales where the numerical result remains within a factor of 2 of the “expected” power for a Kolmogorov cascade, then for a  $N^3$  simulation, the power can be followed down to  $k \sim 2\pi N/5$  (or, equivalently, the methods can meaningfully define some vorticity for structures as small as 5 elements – either particles or cells – across), or an “effective” Reynolds number of  $\text{Re} \sim (L_{\text{box}}/L_{\text{diss}})^{4/3} \sim 0.1 N^{4/3}$ .

As expected, SPH performs less well here. Small-scale structure is lost in both TSPH and PSPH, owing to low-order gradient errors introducing noise and artificial viscosity not perfectly vanishing. In TSPH, the artificial viscosity is high everywhere, so

there is almost no inertial range, and convergence is very slow. An extensive study of TSPH on this problem is presented in Bauer & Springel (2012); our conclusions are consistent with theirs. As shown by Price (2012a) and Hopkins (2013), an artificial viscosity switch improves the performance of SPH greatly, and even allows us to see some convergence, but at all resolutions we study the cascade in PSPH is still truncated compared to the non-SPH methods (by a factor  $\sim 4$ ).

In Fig. 28, we repeat these experiments but now with highly super-sonic turbulence (rms  $\mathcal{M} \sim 8.4$ ). Consistent with many previous studies, we find smaller differences between SPH and all other methods (see e.g. Kitsionas et al. 2009; Price & Federrath 2010; Bauer & Springel 2012). The dynamic range of the velocity and density power is similar to the sub-sonic case, though an inertial range is less well-defined. Since the power on small scales in super-sonic turbulence is dominated by shocks and discontinuities (Burgers 1973), the essential property of the methods is that they can stably capture strong shocks and advection; in general, as long as the methods are conservative and numerically stable (true of all the methods here, although not of many finite-difference type methods), they do reasonably well in this limit.

We have also repeated our experiments with compressively-driven turbulence, and find similar systematic differences between methods (with overall properties consistent with those well-known from previous studies; Schmidt et al. 2009; Federrath et al. 2010); stationary-grid methods there perform slightly poorer since larger

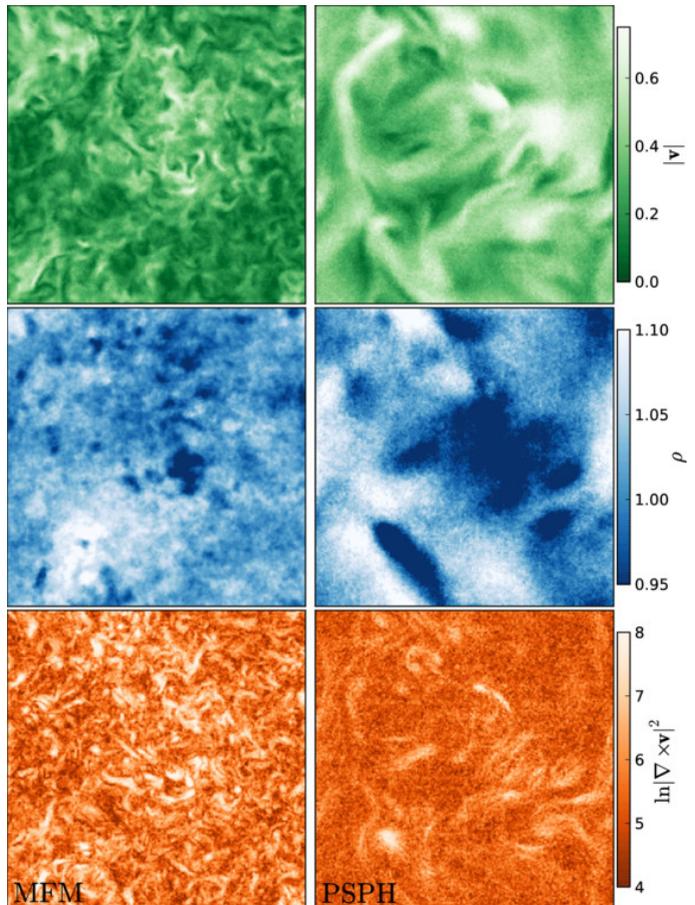


**Figure 25.** Quantitative decay of the “blob” in Fig. 24. We plot the total mass of cold, dense gas (normalized to the initial cloud mass) at each simulation time (normalized to the KH timescale  $\tau_{\text{KH}} = 2$ ). Here, the grid result is from ENZO, an AMR code, but ATHENA agrees well, as do the MFM and MFV methods. In TSPH, surface tension effects and the suppression of mixing instabilities prevent the destruction of the cloud. In PSPH, most of these effects are eliminated so the cloud is much more well-mixed. However, without artificial conductivity, a “tail” of particles remain low-entropy and dense because there is no mechanism for generation of mixing entropy.

density gradients must be advected. And we have repeated our experiments in 2D, with the turbulence driven on small scales, to verify that we indeed recover the expected inverse cascade; our conclusions regarding the relative performance of different methods are identical, with all methods recovering a slightly larger inertial range. The 1D analogue (Burgers turbulence) is essentially just the randomly-driven version of the interacting blastwave problem.

## 5.5 Tests with Self-Gravity

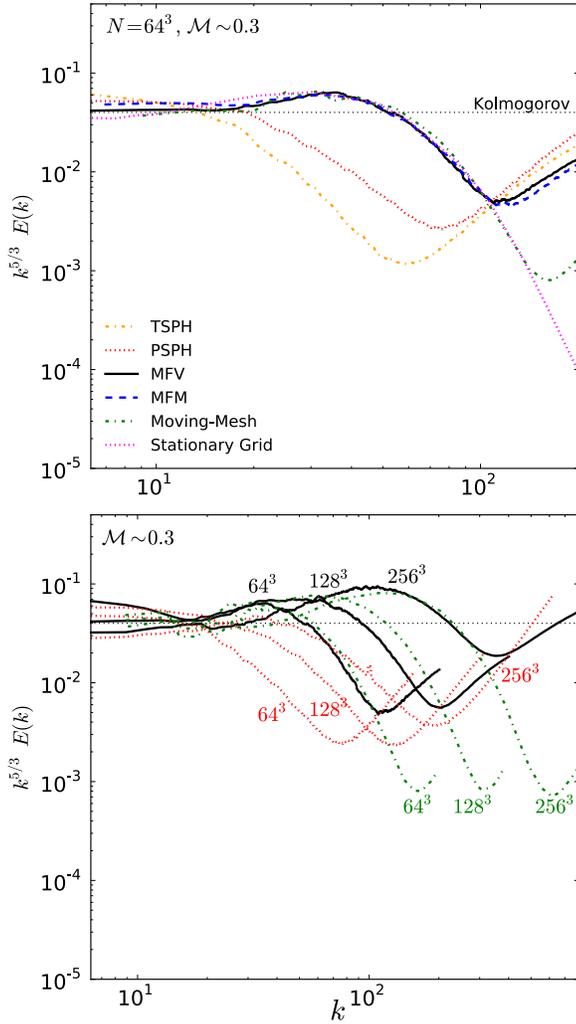
Now we consider several tests involving self-gravity and hydrodynamic forces on gas. Recall, the N-body gravity algorithm here is essentially identical to that in GADGET and AREPO, modulo well-tested improvements and optimizations, and this has been tested in a huge variety of situations (see e.g. Springel 2005; Springel et al. 2005b; Hayward et al. 2011; Vogelsberger et al. 2012; Scannapieco et al. 2012; Kim et al. 2013, and references therein). 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 Kim et al. 2013, 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.



**Figure 26.** Driven sub-sonic turbulence (§ 5.4.4), with our MFM (left) and PSPH (right) methods. We show the velocity (top), density (middle), and enstrophy (bottom), in 2D slices through the center of the 3D box. The resolution is  $256^3$ , and the time is chosen so the turbulence has reached quasi-steady-state with rms Mach number  $\mathcal{M} \sim 0.3$ . The MFV and MFM results are nearly identical, and closely resemble stationary-grid and moving-mesh results (compare Fig. 4 in Bauer & Springel 2012); note in particular the fine structure in vorticity which is captured. SPH tends to smear out some of the small-scale structure.

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

We begin with the simple but very relevant test problem from Evrard (1988), which is commonly used to test SPH codes (Hernquist & Katz 1989; Davé et al. 1997; Springel et al. 2001; Wadsley et al. 2004), 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 code

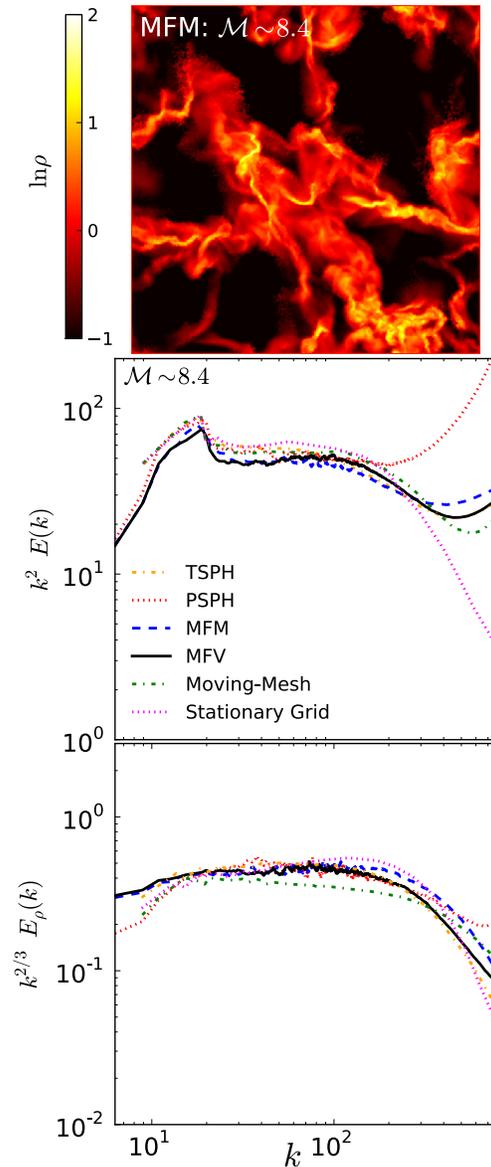


**Figure 27.** Compensated velocity power spectra for the driven, sub-sonic turbulence in Fig. 26. Dotted line shows the Kolmogorov  $E(k) \propto k^{-5/3}$  law. *Top:* Different methods at low ( $64^3$ ) resolution. MFM, MFV, moving-mesh, and stationary-grid methods are essentially identical down to the grid-scale, even including the bottleneck regime. The differences very close to the grid scale are not physically meaningful. TSPH fails to capture much power at all, on scales between the grid and driving scale. PSPH fares better, but still suppresses power in the velocity and vorticity fields on intermediate scales, compared to other methods (owing to noise in the gradient estimators). *Bottom:* MFV, moving-mesh, and PSPH spectra vs. resolution. MFV and moving-mesh (also MFM and stationary-grid) methods remain identical at higher resolution; these methods show good convergence. The dynamic range of the power captured in PSPH does increase with resolution, but more slowly.

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

Following Springel (2010), we show in Fig. 29 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 MFM 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 ex-

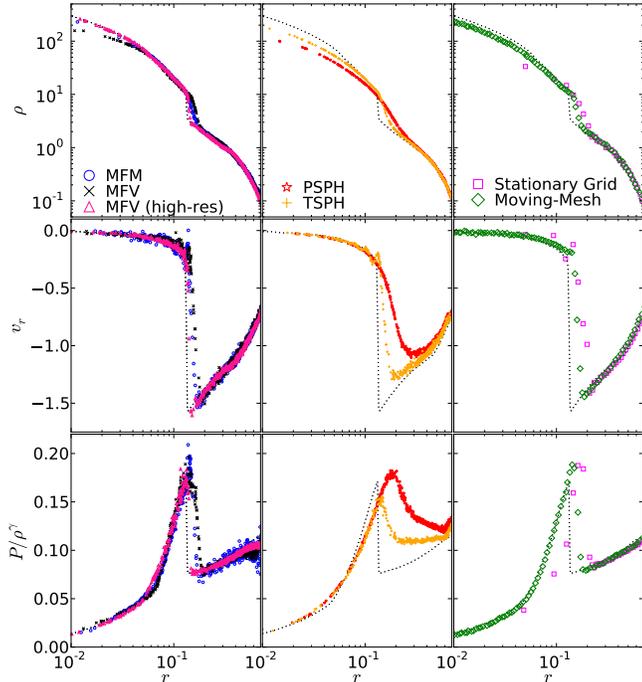


**Figure 28.** As Figs. 26-27, but for super-sonic turbulence (Mach  $\mathcal{M} \sim 8.4$ ). We show the logarithmic density field (*top*), velocity power spectrum (*middle*), and linear density power spectrum (*bottom*), across methods at resolution of  $256^3$ . Here, the differences between methods are smaller.

pected. All the methods capture the key qualitative features of the problem, but with important differences.

The MFM, MFV, and moving-mesh results are similar. MFM 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 MFM 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 MFM; for the same with moving meshes, see Springel 2010, Fig. 41).

SPH captures the key behaviors, but with much more severe smoothing of the shock. In particular the entropy jump is flattened



**Figure 29.** Evrard test (§ 5.5.1); the collapse of a spherical, self-gravitating polytrope. We show the radial profile of density (*top*), velocity (*middle*), and entropy (*bottom*), in low-resolution ( $30^3$ ) runs (all particles are shown). The collapse converts potential energy to kinetic, which sets up a strong shock with a virialized internal structure. At high resolution, the methods all converge to the exact solution (dotted); we demonstrate this with a high-res ( $128^3$ ) MFV run. At low resolution MFM, MFV, and moving-mesh results are similar, with the former two more noisy (and exhibiting some post-shock ringing), but all leading the high-resolution shock location. Stationary grids poorly-resolve the shock interior, suppressing the internal entropy and density, because of lack of adaptive resolution. SPH smooths the shock front much more noticeably, especially P-SPH (because of the larger kernel size and added conduction terms).

and spread over nearly  $\sim 1$  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.

Fixed-grids produce the least-accurate result in the shocked (interior) region.<sup>33</sup> 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

<sup>33</sup> 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 Springel (2010) 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.

than a factor of  $\sim 2$ , the result is not improved (see e.g. Fig. 12 in Bryan et al. 2014, 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.

### 5.5.2 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 useful both as 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 Zel’dovich (1970): 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 unperturbed 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  $x$ -direction), and temperature are then

$$x(q, z) = q - \frac{1+z_c}{1+z} \frac{\sin(kq)}{k} \quad (54)$$

$$\rho(q, z) = \frac{\rho_0}{1 - \frac{1+z_c}{1+z} \cos(kq)} \quad (55)$$

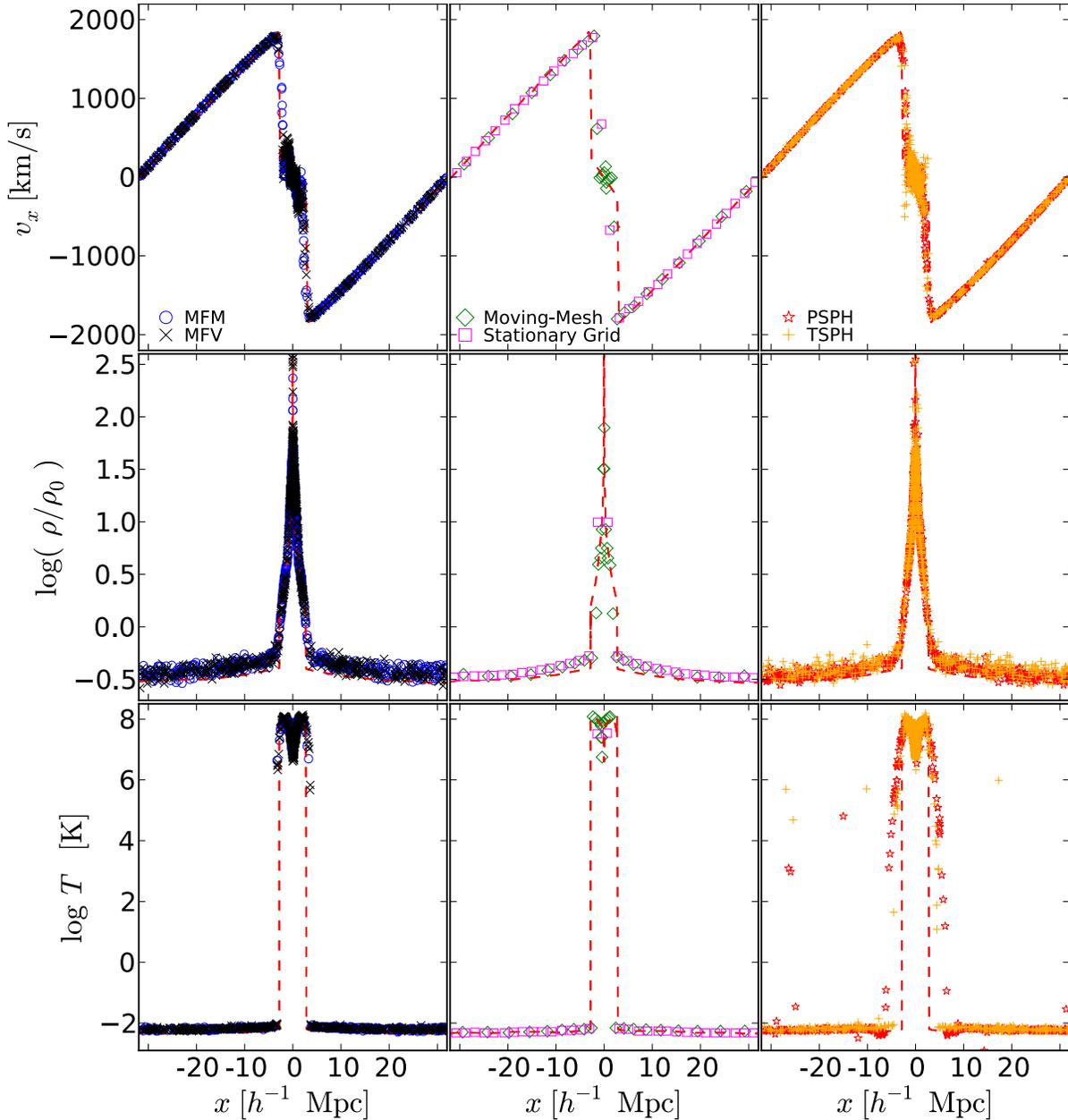
$$v_{\text{pec}}(x, z) = -H_0 \frac{1+z_c}{\sqrt{1+z}} \frac{\sin(kq)}{k} \quad (56)$$

$$T(x, z) = T_i \left[ \left( \frac{1+z}{1+z_i} \right) \frac{\rho(x, z)}{\rho_0} \right]^{2/3} \quad (57)$$

with  $k = 2\pi/\lambda$  the wavenumber of the perturbation (wavelength  $\lambda$ ),  $\rho_0$  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 Bryan et al. (1995); Trac & Pen (2004), we set  $\lambda = 64 h^{-1} \text{Mpc}$  and  $z_c = 1$ , and start the simulations at an initial redshift  $z_i = 100$  (in the linear regime) with  $T_i = 100 \text{K}$  (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. 30 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



**Figure 30.** The Zeldovich pancake (§ 5.5.2). A density perturbation is initialized along the  $x$ -axis (in 3D space) at high redshift in an expanding, baryonic Einstein-de Sitter universe; it grows until collapsing into a caustic and shocking at redshift  $z_c = 1$ . We plot  $x$ -velocity (*top*), density (*middle*), and temperature (*bottom*) at  $z = 0$ , as a function of  $x$  position, at  $32^3$  resolution (the appearance of more elements in the non-mesh results is only because we use a glass IC, instead of a lattice, so the  $x$ -coordinates of particles spaced in other dimensions do not exactly overlap). Dashed red line shows a much higher-resolution (8192) 1D PPM calculation, which should be close to exact. All methods capture the key dynamics. Non-moving meshes under-resolve the shock interior at fixed element number (true even in AMR, in 3D, because the method does not allow anisotropic cells). SPH captures the shock and adiabatic evolution with no special treatment, but smooths the shock significantly and allows some particle-interpenetration (seen in  $v_x$ ) due to imperfect application of artificial viscosity. MFV/MFV methods are similar to each other: there is noise in the low-density  $\rho$ -field, from small inhomogeneities in the glass ICs which are amplified cosmologically; but the interior shock structure, and steep shock jump, are well-captured. Moving-meshes are similar; less noisy but also less well-resolved in the shock center (vs. MFV/MFV) because of the mesh regularization procedure (see text).

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  $\sim 100$  relative to the particle-based methods.<sup>34</sup> The moving mesh does not suffer from

this problem so captures some of the structure and obtains a factor

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  $\sim 1000$  in the 3D version of this problem) to achieve the same accuracy as the moving-mesh result (see e.g. Fig. 13 in Bryan et al. 2014).

<sup>34</sup> In AMR methods, the outward jumps can be better captured with more

$\sim 10$  higher density jump, but this is still over-smoothed by a factor of  $\sim 10$  relative to the MFM, MFV, and SPH methods.<sup>35</sup>

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  $x \sim \pm 5h^{-1}$  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 MFM 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 MFM 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 or explicit thermal energy evolution (see § D). *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 temperature 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 MFM 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 § D).

Second, the code must be able to deal with an extremely anisotropic geometry: the fluid is compressed enormously (factor  $\sim 1000$ ) along the  $x$  axis but not the other two axes. In stationary-

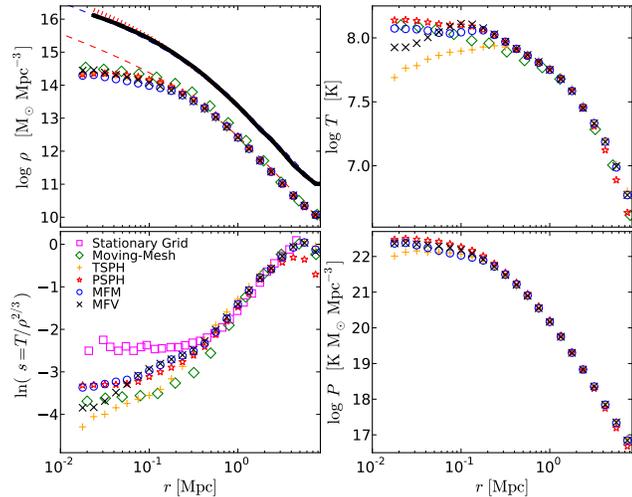
meshes (including AMR), since the cells are always “regular” (usually cubical), this leads to a practical loss of resolution – obviously non-AMR methods lose resolution when the fluid is compressed, but AMR methods which would try and “refine” near  $x \sim 0$  in this problem (i.e. around/within the shock) are forced to refine in the  $y$  and  $z$  directions simultaneously. So to capture a factor  $\sim 10$  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 regularization” 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  $\sim 10$  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  $N_{\text{NGB}}$  neighbors lie exactly along a line in the  $x$ -direction! To handle this, the adaptive checks described in § A 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 § A. This, however, produces very small differences in the final solution. Note that all these problems are artificially masked (and we can make all methods appear much more accurate) if one studies a 1D version of the test problem.

### 5.5.3 The Santa Barbara Cluster: Cosmological Hydrostatic Equilibrium, Inflow, & Entropy Noise

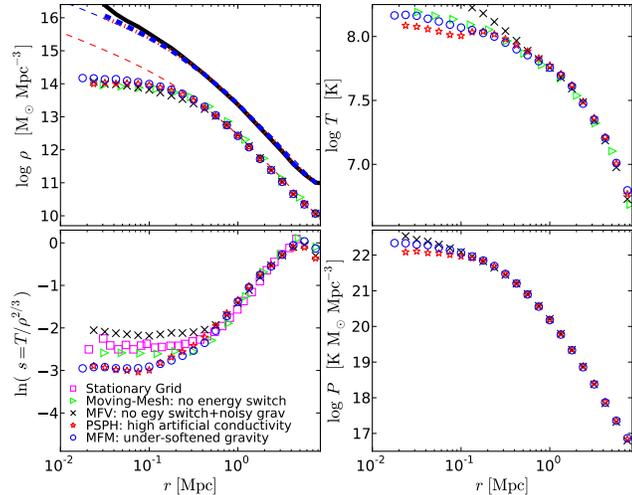
We next consider the “Santa Barbara Cluster” from the comparison project in Frenk et al. (1999). This is a standard reference test problem for which many codes have been compared. It is a “zoom-in” simulation in which a low-resolution cosmological background contains a higher-resolution Lagrangian region 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  $64h^{-1}$  Mpc is initialized at redshift  $z = 49$  ( $a = 1/(1+z) = 0.02$ ), in a flat Universe with dark matter density  $\Omega_{\text{DM}} = 0.9$ , baryonic  $\Omega_b = 0.1$ , Hubble constant  $H_0 = 100 h \text{ km s}^{-1} \text{ Mpc}^{-1}$  with  $h = 0.5$ , and negligible initial gas temperature  $T = 100 \text{ K}$ . The gas is non-radiative (ideal) with  $\gamma = 5/3$ .

In Fig. 31, we show the (spherically mass-weighted average) radial profile of dark matter density, and gas density, temperature, pressure, and entropy, at  $z = 0$  (centered on the center-of-mass of the gas in the most massive system). The dark matter density we compare to an NFW profile with virial radius  $R_{\text{vir}} = 2.734h^{-1} \text{ Mpc}$ ,

<sup>35</sup> In fact, the moving-mesh and stationary-grid results here actually 2D, at  $32^2$  resolution, since that is what was provided by Springel (2010). Since the stationary-grid is not AMR, the results should be identical in the  $32^3$  case, except more expensive. For the moving-mesh case, if one forces the aspect ratios of cells to be regular and the same in both directions perpendicular to the  $x$ -axis, it should again be identical (just more expensive) in 3D, but as discussed below the mesh-deformation problems are more challenging in higher-dimensional versions of the problem. So this comparison may overestimate the accuracy of moving-meshes on this problem.



**Figure 31.** The Santa Barbara cluster (§ 5.5.3); a “zoom-in” simulation of the cosmological formation and collapse of a massive cluster, with collisionless dark matter and non-radiative gas. We plot radially-averaged profiles at  $z = 0$ . *Top Left:* Gas (points) & dark matter (thick lines) density. Dashed lines compare the best-fit NFW profile (blue) and it rescaled by the Universal baryon fraction (red). *Top Right:* Temperature. *Bottom Left:* Entropy. *Bottom Right:* Pressure. All methods agree well on the dark-matter structure, and reasonably well on the gas-pressure profile (determined by hydrostatic equilibrium vs. gravity). The important differences are in central entropy/temperature. Stationary grids (here, from the AMR code RAMSES) produce high-entropy “cores.” TSPH predicts a nearly power-law entropy decline. Moving-mesh, MFM, MFV, and PSPH methods all produce intermediate cases: some “core” but at a much weaker level than grid codes (closer to TSPH). TSPH appears lowest due to its suppression of fluid mixing, grids highest due to their tendency to over-mix.



**Figure 32.** Santa Barbara cluster as Fig. 31, but comparing the runs in our large parameter survey of  $\sim 250$  test runs which most closely reproduce the grid-code results with the moving-mesh, MFM, MFV, and PSPH methods. Turning off the energy-entropy switch (i.e. allowing spurious heating from Riemann solver errors) in moving meshes, using too small a gravitational force softening in the MFM/MFV Riemann methods (so shot noise in the gravitational potential is translated into small shocks and entropy production), or forcing large artificial conductivity values in PSPH, all enable us to roughly reproduce the grid-code results. However all produce clearly unphysical artifacts in this and other test problems. We conclude that the high entropy cores in AMR codes are almost certainly over-estimated.

and concentration  $c = 7.5$ , which provides a reasonably good fit to all our simulations. Note that all the methods here, and indeed the other methods in the literature and even the original survey of methods in Frenk et al. (1999), agree fairly well on the dark matter profile and, in turn, the gas pressure profile (because the pressure gradient must balance gravity, which is primarily set by the dark matter profile). The methods also all agree well on the gas density/temperature/entropy profiles outside the cluster center ( $\gtrsim 0.2 h^{-1}$  Mpc).

The largest differences between methods reflect what Frenk et al. (1999) 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. The difference is discussed at length in Springel (2010, § 9.3 therein); briefly, SPH conserves particle entropy accurately (unlike grid methods), but suppresses fluid mixing, hence mixing entropy when averaging over finite scales. Grid codes, on the other hand, over-mix and diffuse entropy, and are subject to spurious “grid heating” (noise in the gravitational field from collisionless particles producing weak shocks which heat the gas). The difference persists even in modern, high-resolution comparisons: note that the state-of-the-art AMR result here from RAMSES (see also Power et al. 2013) is very similar to other AMR codes like ENZO and the original Frenk et al. (1999) fixed-grid results, and the TSPH result here is very similar to the Frenk et al. (1999) SPH results and GADGET-2.

Interestingly, the moving-mesh, MFM, MFV, and PSPH results lie generally between the TSPH and stationary-grid result, but somewhat closer to TSPH. The largest central entropy among these methods is predicted by PSPH, actually, but we have shown that this method tends to over-diffuse entropy compared to MFM, MFV, and moving meshes. The MFM and MFV predictions agree well; interestingly, the moving-mesh result from AREPO is slightly closer to the TSPH result in entropy, but to MFM and MFV in temperature.

To investigate this further, we have re-run an extensive suite of simulations of the cluster IC using GIZMO:  $> 50$  high-resolution ( $128^3$ ) runs and  $> 200$  low-resolution ( $64^3$ ) runs, in which we have systematically varied numerical aspects of the method like the choice of Riemann solver, slope-limiter, order of the reconstruction, gravitational softening (relative to the inter-particle separation), Courant factor/timestep criteria, energy/entropy switches, gravitational force accuracy, and (in SPH) artificial viscosity and conductivity parameters. The result of this extensive survey strongly supports the conclusions from Springel (2010). Fig. 32 illustrates this with a few representative simulations: we show that we can reproduce the stationary-grid results *if* we artificially enhance the numerical diffusion and/or gravitational “noise” in each method. For example, Springel (2010) show that if they disable the energy/entropy switch used to suppress artificial heating of adiabatic flows with high bulk Mach number, they obtain a result very similar to the stationary grid; however this numerical method is clearly wrong, since it gives a seriously incorrect solution for the (analytically known) temperature of the IGM in the early Universe and produces too much entropy on other tests (e.g. the Zeldovich pancake).

Similarly, in both our MFM and MFV methods, if we use a very strong slope limiter or a lower-order method (greatly increasing the numerical diffusion in other test problems), we can reproduce the stationary-grid result (with similar errors in the adiabatic phase). Alternatively, we can under-soften gravity for the dark matter – i.e. reduce the gravitational softening for the dark matter to a value smaller than the inter-particle separation in the cluster center

at this resolution – in which case the noise seeded by individual particle motions is greatly enhanced. This leads to “jostling” of the particles, which in the Riemann solution produces entropy, and again leads to a result similar to stationary grids. In this case, even though the softening is clearly poorly-chosen, since the early Universe is more smooth, the early (adiabatic) phase of expansion is still captured correctly. However the late-time dark matter profile at small radii is corrupted by N-body transfer of energy from dark matter to gas particles; again indicating this solution is clearly incorrect. In TSPH, these sources of noise are suppressed by the particle-based entropy conservation (i.e. we might get a more correct answer in these limits, but for the wrong reasons). But in PSPH, the addition of an artificial conductivity term means the same noise sources lead to similar effects. Alternatively, in PSPH, we can simply choose to enhance the numerical entropy diffusion by making the artificial conductivity coefficient much larger: we show the results increasing this by a factor of  $\sim 5$ , which leads to reasonable agreement with the stationary-grid results. However, this leads to seriously excessive diffusion in nearly every other test problem we consider (where PSPH was already one of the most diffusive methods). In short, essentially every run with parameters that give good results on the other test problems leads to an answer similar to those in Fig. 31 – i.e. much lower central entropies compared to stationary-grid codes; while every parameter choice we consider which gives good agreement with the stationary-grid codes on this test problem leads to a serious problem in some other test.

We therefore echo the conclusions of Springel (2010). While the “exact” correct solution to the SB cluster central entropy problem remains unclear, it almost certainly lies between the results from stationary-grid/AMR codes (which over-mix, predicting too much entropy owing to advection errors and spurious “gravitational heating”) and traditional SPH codes (which under-mix).

#### 5.5.4 Isolated Galaxy Disks: Modeling Complex ISM Physics

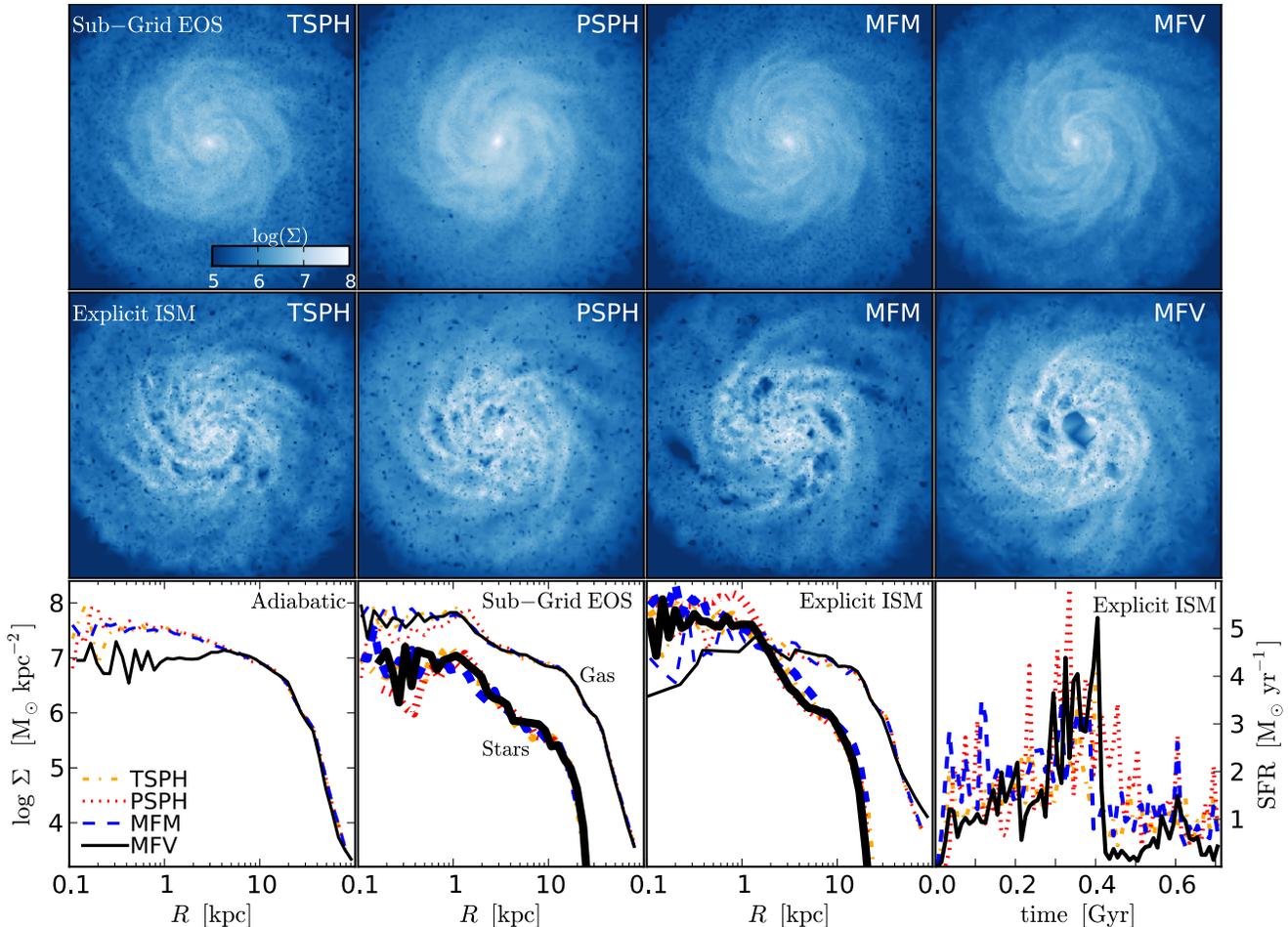
We now consider a more practical “realistic” problem – evolving a Milky Way-like galactic disk, with stars, gas, and dark matter. This is not so much a test problem (in that there is no known “correct” solution), as it is a means to check whether the methods here are useable on real, complicated problems that involve a wide range of physics including highly non-linear, chaotic processes like stellar feedback and star formation. For this problem, we will invoke a wide range of additional physics beyond just gravity and hydrodynamics; because there is no implementation of these physics in the moving-mesh or stationary-grid codes to which we compare (and, as we show, the choice of physics included dominates the solution), we restrict our comparison to the methods we can run within GIZMO.

The initial galaxy has a bulge, stellar and gaseous disk, halo, and central black hole. They are initialized in equilibrium following Springel et al. (2005a) so that in the absence of cooling, star formation, and feedback there are no significant transients. The galaxy has baryonic mass  $M_{\text{bar}} = 7.1 \times 10^{10} M_{\odot}$  and halo mass  $M_{\text{halo}} = 1.6 \times 10^{12} M_{\odot}$  (concentration  $c = 12$ ), black hole mass  $3 \times 10^6 M_{\odot}$ , a Hernquist (1990) profile bulge with mass  $m_b = 1.5 \times 10^{10} M_{\odot}$ , and exponential stellar ( $m_d = 4.7 \times 10^{10} M_{\odot}$ ) and gas disks ( $m_g = 0.9 \times 10^{10} M_{\odot}$ ) with scale-lengths  $h_d = 3.0$  and  $h_g = 6.0$  kpc, respectively. The gas disk is initially vertically pressure supported with scale-height  $z_0 = 0.3$  kpc, and the stellar disk scale height and velocity dispersion is such that the Toomre  $Q = 1$  everywhere. The disk is evolved with vacuum boundary conditions (i.e. in isolation, non-cosmologically) for 0.5 Gyr (a couple of galactic orbits at 10kpc, but  $\sim 100$  orbits near our resolution

$\sim 0.1$  kpc!). We intentionally focus on a low-resolution example, where differences between methods will be maximized: we use  $3.5e4$ ,  $5.0e4$ ,  $2.0e4$ , and  $1.0e4$  particles for the initial gas disk, halo, stellar disk, and bulge.

We consider three different physics modules, of varying complexity. First, an “adiabatic” model, pure hydro+gravity. Here all stars, dark matter, and black holes are collisionless, and the gas obeys a  $\gamma = 5/3$  equation of state. The test is similar our our Keplerian disk, but for a self-gravitating, thick, three-dimensional gas+stellar disk. All of the methods produce very similar results. The disks develop spiral structure, but do not transfer much angular momentum over this time. At 0.5 Gyr, the TSPH, PSPH, and MFM results are nearly identical (and all within  $\sim 25\%$  of the initial surface density profile at this time, showing the disk is quite stable). The MFV result is somewhat different, with the central density depleted and outermost density enhanced owing to a slow outward diffusion of angular momentum. This has to do with the small angular momentum advection errors associated with the mass fluxes between particles, as in § 5.2.4. The central part of the disk at  $\sim 0.1$  kpc has executed  $\sim 100$  orbits by this time, so any preservation of the disk at all is remarkable! We expect, based on § 5.2.4, that the MFM method can continue to preserve angular momentum accurately even at such late times; what is more surprising is that the SPH methods show little transfer as well. Recall, in the Keplerian disk problem, the degradation of the disk in SPH was caused by the viscous instability. Here, two effects strongly suppress this. First, the gas is much hotter and more strongly pressure-supported, especially in the center, where it reaches  $h/R \sim 1$ ; so the fractional effect of erroneous viscous forces is much smaller. Second, the disk is relatively gas-poor ( $\sim 5 - 10\%$  gas in the central regions), so the collisionless stellar disk actually dominates the dynamics and the gas disk is stabilized by the mutual interaction with this collisionless component. As a result, the angular momentum errors from pressure forces become negligible, and the only angular momentum errors that build up are those from advection; hence the small effect still visible in the MFV method. This is a serious concern, still, for fixed-grid codes, where advection errors are much larger than in Lagrangian codes.

The next model we consider uses the Springel & Hernquist (2003a) sub-grid model for a multi-phase interstellar medium (ISM) and star formation. This has been used in a wide range of previous and current work on cosmological galaxy formation (e.g. Springel & Hernquist 2003b; Robertson et al. 2004; Di Matteo et al. 2005; Hopkins et al. 2006, 2008a,b; Narayanan et al. 2006; Vogelsberger et al. 2013). In these models, rather than attempt to resolve the ISM structure or feedback explicitly, the ISM is parameterized by an “effective equation of state (EOS)” at high densities ( $n \gtrsim 0.1 \text{ cm}^{-3}$ ), with an adjustable law which turns gas into stars at a fixed efficiency of  $\sim 1\%$  per gas dynamical time (tuned to be similar to the observed Schmidt-Kennicutt relation). Gas at lower densities follows a standard atomic cooling curve from Katz et al. (1996). Here we see similar spiral structure to the pure adiabatic case; the effective EOS is quite “stiff” so keeps the gas in the disk smooth. By 0.5 Gyr about  $\sim 20\%$  of the gas has turned into stars, within the disk radius ( $\sim 10$  kpc) where the density meets the threshold value above. Here, both the stellar and gas mass profiles agree very well across all methods. There is still a small angular momentum diffusion in the MFV result at large radii, but at small radii we see that the effects are swamped by the effect of slightly enhanced gravitational instability in the disk (it is not quite as “stiff” as  $\gamma = 5/3$  here) leading to gas inflows into the center along the spiral arms which enhances the gas mass within  $\sim 1$  kpc.



**Figure 33.** Isolated Milky Way-like galaxy disk, with stars, gas, and dark matter, evolved for several orbits. The initial conditions and physics are identical in each case, only the hydrodynamic method is varied. *Top:* Projected disk gas density at  $t = 0.7$  Gyr ( $\sim 5$  orbits at the effective radius  $\sim 5$  kpc,  $\sim 150$  orbits at  $\sim 0.1$  kpc), in a box 70 kpc on a side. The Springel & Hernquist (2003a) sub-grid model for star formation and the ISM (treating the ISM with an “effective equation of state” determined via stellar feedback) is used. *Middle:* Same, but using the Hopkins et al. (2013b) physics models, which explicitly treat low-temperature cooling, star formation, and stellar feedback via SNe, radiation pressure, photo-heating, and stellar winds, leading to a multi-phase ISM. *Bottom:* Mass profiles and SFR in the simulations. We plot the projected (face-on) surface density profiles of gas (thin) and stars formed during the simulation (thick) in runs using only adiabatic hydrodynamics+gravity (*left*; no star formation here); using the Springel & Hernquist (2003a) sub-grid ISM treatment (*center-left*); using the Hopkins et al. (2013b) explicit treatment of the ISM and star formation (*center-right*). Finally, for the explicit ISM case, we plot the SFR vs. time for each method (*right*). All the methods agree well in morphology, star formation history, and disk angular momentum evolution, with weak second-order differences discussed in the text. The problem is clearly dominated by the input physics rather than numerical methods.

The third model we compare is the “explicit feedback” model used in the FIRE (Feedback In Realistic Environments) simulations, described in a suite of papers (Hopkins et al. 2011, 2013b; Narayanan & Hopkins 2012; van de Voort et al. 2014). Briefly, in these simulations the multi-phase ISM and stellar feedback are treated explicitly: gas can cool to  $< 100$  K via fine-structure and molecular cooling (Hopkins et al. 2012b), and star formation occurs in dense regions above a threshold  $n > 10 \text{ cm}^{-3}$ , which are also molecular (self-shielding), and locally self-gravitating (see Hopkins et al. 2013d). The energy, momentum, mass, and metal fluxes from various feedback mechanisms are followed explicitly according to standard stellar evolution models; this includes radiation pressure in the UV and IR (see Hopkins et al. 2011), supernovae types I & II, stellar winds (O-star and AGB), and photo-ionization and photo-electric heating (Hopkins et al. 2012d). The combination of these physics lead naturally to a self-regulating, multi-phase ISM (Hopkins et al. 2012a, 2013c) with strong galactic outflows (Hopkins et al. 2012c, 2013a). We see that the resulting ISM structure in this case shows a more clumpy morphology, as expected,

with large GMC complexes and bubbles produced via overlapping SNe explosions. The methods differ in detail, but these differences are consistent with being essentially stochastic – the interaction of feedback and the ISM is highly chaotic, so we do not expect exact agreement here. The mass profiles are similar in all cases; the central kpc of the galaxy rapidly turns  $\sim 50\%$  of its mass into stars, while the outer regions form stars slowly. The gas densities at  $> 10$  kpc are elevated by the presence of galactic winds and fountains, which increase the gas mass at large radii considerably. For these models, we also plot the star formation histories. Here we see considerable short-timescale variability, which again relates to the chaotic nature of local star formation and feedback, but the qualitative properties are quite similar: in all cases, there is a mini-burst from a nuclear-bar induced ring which builds up gas at  $\sim 1$  kpc and turns into stars at  $t \sim 0.2 - 0.3$  Gyr, after which the system relaxes again. Remarkably, despite the extremely non-linear nature of the physics included, the different numerical methods here produce similar results.

## 6 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 are 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 MFM 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 Riemann 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 ( $N_{\text{NGB}} = 32$ ), PSPH ( $N_{\text{NGB}} = 200$ ), MFM, 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 ( $N_{\text{NGB}} = 128$ ), MFM, 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  $\sim N_{\text{NGB}}^{3/2}$ . 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 MFM and MFV methods are somewhat faster on the test problems we have directly compared. The gravity solvers are nearly identical and a Riemann 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 MFM 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 Riemann 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 stencil, 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 can reduce the timesteps by factors of  $\sim 100 - 1000$  in some of the problems we consider here, compared to the MFM 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 MFM 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 MFM 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, MFM, and MFV methods, and much higher for AREPO and AMR methods.

## 7 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. Rather, they are sub-classes of Lagrangian, meshless, finite-volume Godunov-type methods; in a crude sense, like a moving mesh code “without the mesh.”

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, systematic tests of these methods compared to SPH, moving mesh, and stationary-grid (AMR) methods, and argue they are at least competitive with these methods on all test problems, and appear to capture many of the advantages of both SPH and AMR methods while avoiding many of their disadvantages. More work will be needed, of course, to determine the ultimate utility of these methods, but the results here are promising.

The two new methods here exhibit smaller, but significant, differences between each other. The MFM method exhibits slightly reduced noise, and superior angular momentum conservation, compared to the MFV method; MFM also has the advantage of conserved particle masses, which is very useful for tracing the his-

tory of fluid elements and for simulations with complicated self-gravitating interactions (e.g. galaxy and star formation), and reduces the “gravitational heating” errors in problems like the Santa Barbara cluster. However, this comes at the cost of being slightly more diffusive, and necessarily spreading contact discontinuities over a larger fraction of the kernel width, so that shocks and phase boundaries in e.g. the KH or RT instabilities are captured less-sharply.

### 7.1 Comparison to SPH

Both methods we propose avoid many known problems with SPH methods, and as a result give more accurate results in the tests we consider. Even in the “modern” SPH,<sup>36</sup> potentially important issues arise with noise, artificial diffusion, fluid mixing, and subsonic 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 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  $\sim N_{\text{NGB}}^{1/2}$ , depending on the choice of kernels).

Our methods eliminate the need for artificial dissipation terms and so – despite the use of a Riemann solver – are substantially *less* diffusive than even the high-order modern SPH switches/schemes. They conserve angular momentum 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 much 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 zeroth and first-order errors of SPH, *while remaining fully conservative*. This means, most importantly, the methods converge at fixed neighbor number. We are therefore able to obtain much higher accuracy with  $\sim 32$  neighbors than SPH with  $\sim 400$  neighbors, on most problems we consider. And as noted in § 6, at fixed neighbor and particle number there is little significant performance difference between SPH and our new methods.

SPH may still have some advantages in 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 Riemann solver. It is computationally an incredibly simple method. It trivially handles free surfaces with no 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.

<sup>36</sup> “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 switches for artificial viscosity & conduction.

### 7.2 Comparison to AMR

Our new methods also avoid many disadvantages of stationary (non-moving) grid methods, for certain classes of problems. In grid methods advection errors are large when the fluid moves with respect to the grid, the errors depend on the bulk velocity (solutions often degrade when the fluid moves), angular momentum is not conserved (unless the grid is designed around a particular geometry), spurious “grid alignment” and “carbuncle” instabilities can appear, and coupling to N-body gravity solvers is generally 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 in moving flows. It also leads to dramatically reduced diffusion in any problems involving non-grid aligned motion. The new methods are Lagrangian and errors are independent of velocity, so they 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 can rapidly diffuse away self-gravitating clouds or structures moving relative to the grid. As we and Springel (2010) show, this is also important for fluid mixing instabilities: the velocity dependence 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; effectively, the simulation resolution is downgraded). There is also no “grid alignment” effect so the carbuncle instability does not appear, 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 cartesian grid codes, gas in a rotating disk loses angular momentum and the orbits break down completely in just a couple orbits, even with  $> 10^7$  resolution elements in the disk.<sup>37</sup>

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

<sup>37</sup> Of course, all of these errors in grid codes (and SPH codes) are resolution-dependent; the methods do formally converge, so they can be reduced by increasing resolution. However, for any practical problem the resolution cannot be infinite so we do care about accuracy at fixed resolution. Moreover, for many problems, the convergence is slow, so formal convergence with some methods may be unattainable. For example, it is well-known that in cartesian 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  $\lesssim 3$  orbits (see Hahn et al. 2010, for an example in RAMSES). 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. Of course, errors can also be reduced by choosing grids with specially designed geometries for a specific problem, but this cannot be generalized to all cases.

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); it is not obvious, in particular, if constrained-transport methods can be applied. Grid codes, especially fixed (non-adaptive, non-moving) regular (locally orthogonal) meshes minimize certain forms of numerical noise (“grid 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, or other problems where launching of even weak waves sourced by numerical errors could corrupt the desired behavior, 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, albeit not all, of these advantages. However, in an AMR scheme, one major additional 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).

### 7.3 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 appear at least competitive with one another. However there are some differences already evident in our comparisons with AREPO.<sup>38</sup>

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  $\sim 0.01$ ).

On the other hand, the symmetry and angular momentum conservation in our new methods – particularly for gas in gravitational orbits (e.g. disks) – may be somewhat 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 MFM and MFV methods arises because we map to spherical kernel functions partitioning the volume. This means angular momentum can be well-defined and conserved. In a moving mesh, any irregular (non-symmetric) mesh shape means that the total cell angular momentum cannot be defined at higher than second-order quadrature & integration accuracy (see e.g. Duffell & MacFadyen 2012); although we stress that moving-mesh methods still have some advantage over cartesian-grid codes and (highly-viscous) traditional SPH methods. Similarly, the equations of motion in

<sup>38</sup> 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 some tests (Noh, Sedov, interacting blastwaves) relative to the analytic result. We suspect this owes to either a slightly too-aggressive adaptive timestepping or application of the entropy-energy switch, since we find both of these effects can reproduce this error in our own MFM and MFV calculations. The latter effect has been resolved in more recent applications of AREPO (V. Springel, private communication). In some problems, we see reduced post-shock ringing/noise and wall-heating 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 than to the method itself.

the meshless methods here are manifestly symmetry-maintaining, whereas in moving-mesh approaches the discontinuous nature of cell splitting/merging needed to deal with irregular shapes may lead to symmetry-breaking “mesh-bending” instabilities (see Springel 2010).

### 7.4 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 Riemann 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 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.<sup>39</sup>

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). Gaburov & Nitadori (2011), for example, show how to generalize this to third-order PPM-like method. Based on their and our own 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.

It would be particularly useful to explore more accurate, higher-order quadrature rules for the volume partition (evaluating  $V_i \equiv \int \psi_i(\mathbf{x}) d^3\mathbf{x} \approx \omega(\mathbf{x}_i)^{-1}$ ). As we argued above, in many tests, the non-exact nature of our discretized quadrature rule leads to noise which is avoided in moving-mesh and static grid codes; if this can be eliminated, it would represent a considerable improvement in the method.

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. Gaburov & Nitadori (2011) 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 Duffell & MacFadyen 2011 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.<sup>40</sup>

<sup>39</sup> For example, we have chosen a simple, commonly-used kernel from the SPH literature. However, the kernel function here has a different meaning from that in SPH and is freed from some of the restrictions of SPH kernels. So studies based on SPH kernels should be revisited, with a more appropriate literature being that on kernel estimation of least-squares field gradients.

<sup>40</sup> As noted in § 1, a public version of the GIZMO code is available at <http://www.tapir.caltech.edu/~phopkins/Site/>

## ACKNOWLEDGMENTS

We thank Volker Springel, Romain Teyssier, Lars Hernquist, Jim Stone, Dusan Keres, Paul Duffell, James Wadsley, Oscar Agertz, Thorsten Naab, Richard Bower, Matthieu Schaller, Nick Gnedin, Andrey Kravtsov, Desika Narayanan, Ji-Hoon Kim, Andrew Wetzel, Eliot Quataert, Paul Torrey, Norm Murray, Claude-Andre Faucher-Giguere, and many others for conversations motivating, clarifying, and guiding the development of this work. I must express my immense debt and gratitude to the many people in the numerical development community who have made their codes and methods public, particularly Volker Springel, Jim Stone, and Romain Teyssier, whose excellent codes were used throughout this project. Partial support for PFH was provided by the Gordon and Betty Moore Foundation through Grant #776 to the Caltech Moore Center for Theoretical Cosmology and Physics, and the Alfred P. Sloan Foundation through a Sloan Research Fellowship #BR2014-022. The simulations here were run on the Stampede supercomputer at TACC through NSF XSEDE allocation #TG-AST130039.

## REFERENCES

- Abel, T. 2011, *MNRAS*, 413, 271
- Agertz, O., et al. 2007, *MNRAS*, 380, 963
- Balsara, D. S. 1989, Ph.D. thesis, Univ. Illinois at Urbana-Champaign
- Balsara, D. S. 2004, *ApJS*, 151, 149
- Barth, T., & Jespersen, D. 1989, The design and application of upwind schemes on unstructured meshes (*American Institute of Aeronautics and Astronautics*), 366+
- Bauer, A., & Springel, V. 2012, *MNRAS*, 423, 3102
- Berger, M. J., & Colella, P. 1989, *Journal of Computational Physics*, 82, 64
- Bilotta, G., Russo, G., Héroult, A., & Negro, C. D. 2011, *Annals of Geophysics*, 54
- Brandenburg, A., & Dobler, W. 2002, *Computer Physics Communications*, 147, 471
- Bryan, G. L., Norman, M. L., Stone, J. M., Cen, R., & Ostriker, J. P. 1995, *Computer Physics Communications*, 89, 149
- Bryan, G. L., et al. 2014, *ApJS*, 211, 19
- Burgers, J. 1973, *The nonlinear diffusion equation: asymptotic solutions and statistical problems* (D. Reidel Pub. Co.)
- Byerly, Z. D., Adelstein-Lelbach, B., Tohline, J. E., & Marcelllo, D. C. 2014, *ApJS*, 212, 23
- Cartwright, A., Stamatellos, D., & Whitworth, A. P. 2009, *MNRAS*, 395, 2373
- Cha, S.-H., Inutsuka, S.-I., & Nayakshin, S. 2010, *MNRAS*, 403, 1165
- Cha, S.-H., & Whitworth, A. P. 2003, *MNRAS*, 340, 73
- Clarke, D. A. 2010, *ApJS*, 187, 119
- Colella, P. 1990, *Journal of Computational Physics*, 87, 171
- Cullen, L., & Dehnen, W. 2010, *MNRAS*, 408, 669
- Cunningham, A. J., Frank, A., Varnière, P., Mitran, S., & Jones, T. W. 2009, *ApJS*, 182, 519
- Davé, R., Dubinski, J., & Hernquist, L. 1997, *New Astronomy*, 2, 277
- Davis, S. F. 1984, *Journal of Computational Physics*, 56, 65
- Dehnen, W., & Aly, H. 2012, *MNRAS*, 425, 1068
- Di Matteo, T., Springel, V., & Hernquist, L. 2005, *Nature*, 433, 604
- Dilts, G. A. 1999, *International Journal for Numerical Methods in Engineering*, 44, 1115
- Duffell, P. C., & MacFadyen, A. I. 2011, *ApJS*, 197, 15
- . 2012, *ApJ*, 755, 7
- . 2014, *MNRAS*, in press, arxiv:1407.7300
- Durier, F., & Dalla Vecchia, C. 2012, *MNRAS*, 419, 465
- Evrard, A. E. 1988, *MNRAS*, 235, 911
- Federrath, C., Klessen, R. S., & Schmidt, W. 2008, *ApJL*, 688, L79
- Federrath, C., Roman-Duval, J., Klessen, R. S., Schmidt, W., & Mac Low, M.-M. 2010, *A&A*, 512, A81+
- Frenk, C. S., et al. 1999, *ApJ*, 525, 554
- Fromang, S., Hennebelle, P., & Teyssier, R. 2006, *A&A*, 457, 371
- Gaburov, E., Johansen, A., & Levin, Y. 2012, *ApJ*, 758, 103
- Gaburov, E., & Nitadori, K. 2011, *MNRAS*, 414, 129
- García-Senz, D., Cabezón, R. M., & Escartín, J. A. 2012, *A&A*, 538, A9
- Gingold, R. A., & Monaghan, J. J. 1977, *MNRAS*, 181, 375
- . 1983, *MNRAS*, 204, 715
- Gnedin, N. Y. 1995, *ApJS*, 97, 231
- Gresho, P. M., & Chan, S. T. 1990, *International Journal for Numerical Methods in Fluids*, 11, 621
- Hahn, O., Teyssier, R., & Carollo, C. M. 2010, *MNRAS*, 405, 274
- Hayward, C. C., et al. 2011, *MNRAS*, in preparation
- Heitmann, K., et al. 2008, *Computational Science and Discovery*, 1, 015003
- Hernquist, L. 1990, *ApJ*, 356, 359
- Hernquist, L., & Katz, N. 1989, *ApJS*, 70, 419
- Heß, S., & Springel, V. 2010, *MNRAS*, 406, 2289
- Hopkins, P. F. 2013, *MNRAS*, 428, 2840
- Hopkins, P. F., Cox, T. J., Kereš, D., & Hernquist, L. 2008a, *ApJS*, 175, 390
- Hopkins, P. F., Hernquist, L., Cox, T. J., Di Matteo, T., Robertson, B., & Springel, V. 2006, *ApJS*, 163, 1
- Hopkins, P. F., Hernquist, L., Cox, T. J., Dutta, S. N., & Rothberg, B. 2008b, *ApJ*, 679, 156
- Hopkins, P. F., Keres, D., Murray, N., Hernquist, L., Narayanan, D., & Hayward, C. C. 2013a, *MNRAS*, in press, arXiv:1301.0841
- Hopkins, P. F., Keres, D., Murray, N., Quataert, E., & Hernquist, L. 2012a, *MNRAS*, 427, 968
- Hopkins, P. F., Keres, D., Onorbe, J., Faucher-Giguere, C.-A., Quataert, E., Murray, N., & Bullock, J. S. 2013b, *MNRAS*, in press, arXiv:1311.2073
- Hopkins, P. F., Kereš, D., & Murray, N. 2013c, *MNRAS*, 432, 2639
- Hopkins, P. F., Narayanan, D., & Murray, N. 2013d, *MNRAS*, in press, arXiv:1303.0285
- Hopkins, P. F., Narayanan, D., Quataert, E., & Murray, N. 2012b, *MNRAS*, in press, arXiv:1209.0459
- Hopkins, P. F., Quataert, E., & Murray, N. 2011, *MNRAS*, 417, 950
- . 2012c, *MNRAS*, 421, 3522
- . 2012d, *MNRAS*, 421, 3488
- Hu, C.-Y., Naab, T., Walch, S., Moster, B. P., & Oser, L. 2014, *MNRAS*(arXiv:1402.1788)
- Imaeda, Y., & Inutsuka, S.-i. 2002, *ApJ*, 569, 501
- Inutsuka, S.-I. 2002, *Journal of Computational Physics*, 179, 238
- Katz, N., Weinberg, D. H., & Hernquist, L. 1996, *ApJS*, 105, 19
- Kim, J.-h., et al. 2013, *ApJ*, in press, arXiv:1308.2669

GIZMO.html

Users are encouraged to modify and extend the capabilities of this code; the development version of the code is available upon request from the author.

- Kitsionas, S., et al. 2009, *A&A*, 508, 541
- Koren, B., & van der Maarel, H. T. M. 1993, in *Lecture Notes in Physics*, Berlin Springer Verlag, Vol. 414, *Numerical Methods in Fluid Dynamics*, ed. M. Napolitano & F. Sabetta, 110–114
- Kravtsov, A. V., Klypin, A. A., & Khokhlov, A. M. 1997, *ApJS*, 111, 73
- Kuhnert, J. 2003, in *Lecture Notes in Computational Science and Engineering*, Vol. 26, *Meshfree Methods for Partial Differential Equations*, ed. M. Griebel & M. Schweitzer (Springer Berlin Heidelberg), 239–249
- Lanson, N., & Vila, J.-P. 2008a, *SIAM J. Numer. Anal.*, 46, 1912
- . 2008b, *SIAM J. Numer. Anal.*, 46, 1935
- LeVeque, R. J. 1998, *Journal of Computational Physics*, 146, 346
- Liska, R., & Wendroff, B. 2003, *SIAM Journal on Scientific Computing*, 25, 995
- Liu, M., Xie, W., & Liu, G. 2005, *Applied Mathematical Modelling*, 29, 1252
- Liu, W. K., Jun, S., & Zhang, Y. F. 1995, *International Journal for Numerical Methods in Fluids*, 20, 1081
- Lucy, L. B. 1977, *AJ*, 82, 1013
- Luo, H., Baum, J. D., & Löhner, R. 2008, *Journal of Computational Physics*, 227, 8875
- Lyubarskij, Y. E., Postnov, K. A., & Prokhorov, M. E. 1994, *MNRAS*, 266, 583
- Maddison, S. T., Murray, J. R., & Monaghan, J. J. 1996, *Publications of the Astronomical Society of Australia*, 13, 66
- Maron, J. L., & Howes, G. G. 2003, *ApJ*, 595, 564
- Maron, J. L., McNally, C. P., & Mac Low, M.-M. 2012, *ApJS*, 200, 6
- Marri, S., & White, S. D. M. 2003, *MNRAS*, 345, 561
- Mavriplis, D. J. 1997, *Annual Review of Fluid Mechanics*, 29, 473
- McNally, C. P., Lyra, W., & Passy, J.-C. 2012, *ApJS*, 201, 18
- Mignone, A., Bodo, G., Massaglia, S., Matsakos, T., Tesileanu, O., Zanni, C., & Ferrari, A. 2007, *ApJS*, 170, 228
- Miyoshi, T., & Kusano, K. 2005, *Journal of Computational Physics*, 208, 315
- Mocz, P., Vogelsberger, M., Sijacki, D., Pakmor, R., & Hernquist, L. 2014, *MNRAS*, 437, 397
- Monaghan, J. 1997a, *Journal of Computational Physics*, 138, 801
- Monaghan, J. J. 1997b, *Journal of Computational Physics*, 136, 298
- . 2002, *MNRAS*, 335, 843
- Morris, J. P. 1996, *Publications of the Astronomical Society of Australia*, 13, 97
- Morris, J. P., & Monaghan, J. J. 1997, *Journal of Computational Physics*, 136, 41
- Muñoz, D. J., Springel, V., Marcus, R., Vogelsberger, M., & Hernquist, L. 2013, *MNRAS*, 428, 254
- Müller, E., & Steinmetz, M. 1995, *Computer Physics Communications*, 89, 45
- Murante, G., Borgani, S., Brunino, R., & Cha, S.-H. 2011, *MNRAS*, in press arXiv:1105.1344
- Murphy, J. W., & Burrows, A. 2008, *ApJS*, 179, 209
- Narayanan, D., Cox, T. J., Robertson, B., Davé, R., Di Matteo, T., Hernquist, L., Hopkins, P., Kulesa, C., & Walker, C. K. 2006, *ApJL*, 642, L107
- Narayanan, D., & Hopkins, P. 2012, *MNRAS*, in press, arXiv:1210.2724
- Noh, W. F. 1987, *Journal of Computational Physics*, 72, 78
- Oñate, E., Idelsohn, S., Zienkiewicz, O. C., & Taylor, R. L. 1996, *International Journal for Numerical Methods in Engineering*, 39, 3839
- Okamoto, T., Jenkins, A., Eke, V. R., Quilis, V., & Frenk, C. S. 2003, *MNRAS*, 345, 429
- O’Shea, B. W., Bryan, G., Bordner, J., Norman, M. L., Abel, T., Harkness, R., & Kritsuk, A. 2004, eprint, arXiv:astro-ph/0403044
- O’Shea, B. W., Nagamine, K., Springel, V., Hernquist, L., & Norman, M. L. 2005, *ApJS*, 160, 1
- Pakmor, R., Bauer, A., & Springel, V. 2011, *MNRAS*, 418, 1392
- Peery, K., & Imlay, S. 1988, *Blunt-body flow simulations (American Institute of Aeronautics and Astronautics)*, 2904+
- Pen, U.-L. 1998, *ApJS*, 115, 19
- Power, C., Navarro, J. F., Jenkins, A., Frenk, C. S., White, S. D. M., Springel, V., Stadel, J., & Quinn, T. 2003, *MNRAS*, 338, 14
- Power, C., Read, J. I., & Hobbs, A. 2013, *MNRAS*, in press, arXiv:1307.0668
- Price, D. J. 2008, *Journal of Computational Physics*, 227, 10040
- . 2012a, *MNRAS*, 420, L33
- . 2012b, *Journal of Computational Physics*, 231, 759
- Price, D. J., & Federrath, C. 2010, *MNRAS*, 406, 1659
- Price, D. J., & Monaghan, J. J. 2007, *MNRAS*, 374, 1347
- Rasio, F. A., & Shapiro, S. L. 1991, *ApJ*, 377, 559
- Read, J. I., & Hayfield, T. 2012, *MNRAS*, 422, 3037
- Read, J. I., Hayfield, T., & Agertz, O. 2010, *MNRAS*, 405, 1513
- Ritchie, B. W., & Thomas, P. A. 2001, *MNRAS*, 323, 743
- Robertson, B., Yoshida, N., Springel, V., & Hernquist, L. 2004, *ApJ*, 606, 32
- Roe, P. L. 1981, *Journal of Computational Physics*, 43, 357
- . 1986, *Annual Review of Fluid Mechanics*, 18, 337
- Rosswog, S. 2009, *New Astronomy Reviews*, 53, 78
- . 2014, arXiv e-print: 1405.6034
- Ryu, D., Ostriker, J. P., Kang, H., & Cen, R. 1993, *ApJ*, 414, 1
- Saitoh, T. R., & Makino, J. 2009, *ApJL*, 697, L99
- . 2013, *ApJ*, 768, 44
- Scannapieco, C., et al. 2012, *MNRAS*, 423, 1726
- Schmidt, W., Federrath, C., Hupp, M., Kern, S., & Niemeyer, J. C. 2009, *A&A*, 494, 127
- Schmidt, W., Federrath, C., & Klessen, R. 2008, *Physical Review Letters*, 101, 194505
- Shen, S., Wadsley, J., & Stinson, G. 2010, *MNRAS*, 407, 1581
- Sijacki, D., Vogelsberger, M., Keres, D., Springel, V., & Hernquist, L. 2012, *MNRAS*, 424, 2999
- Springel, V. 2005, *MNRAS*, 364, 1105
- Springel, V. 2010, *MNRAS*, 401, 791
- Springel, V. 2010, *ARA&A*, 48, 391
- Springel, V., Di Matteo, T., & Hernquist, L. 2005a, *MNRAS*, 361, 776
- Springel, V., & Hernquist, L. 2002, *MNRAS*, 333, 649
- . 2003a, *MNRAS*, 339, 289
- . 2003b, *MNRAS*, 339, 312
- Springel, V., White, S. D. M., Tormen, G., & Kauffmann, G. 2001, *MNRAS*, 328, 726
- Springel, V., et al. 2005b, *Nature*, 435, 629
- Steinmetz, M., & White, S. D. M. 1997, *MNRAS*, 288, 545
- Stone, J. M., Gardiner, T. A., Teuben, P., Hawley, J. F., & Simon, J. B. 2008, *ApJS*, 178, 137
- Stone, J. M., & Norman, M. L. 1992, *ApJS*, 80, 753
- Sweby, P. 1984, *SIAM Journal on Numerical Analysis*, 21, 995
- Tasker, E. J., & Bryan, G. L. 2008, *ApJ*, 673, 810
- Teyssier, R. 2002, *A&A*, 385, 337
- Tiwari, S., & Kuhnert, J. 2003, in *Lecture Notes in Computa-*

- tional Science and Engineering, Vol. 26, Meshfree Methods for Partial Differential Equations, ed. M. Griebel & M. Schweitzer (Springer Berlin Heidelberg), 373–387
- Toro, E. F. 1997, Riemann solvers and numerical methods for fluid dynamics : a practical introduction (Berlin, New York: Springer)
- . 1999, Riemann solvers and numerical methods for fluid dynamics: a practical introduction, 2nd edn. (Springer Berlin Heidelberg), xix + 624
- Trac, H., & Pen, U.-L. 2004, *New Astronomy*, 9, 443
- Truelove, J. K., Klein, R. I., McKee, C. F., Holliman, II, J. H., Howell, L. H., & Greenough, J. A. 1997, *ApJL*, 489, L179+
- Truelove, J. K., Klein, R. I., McKee, C. F., Holliman, II, J. H., Howell, L. H., Greenough, J. A., & Woods, D. T. 1998, *ApJ*, 495, 821
- Valdarnini, R. 2012, *A&A*, 546, A45
- van de Voort, F., Quataert, E., Hopkins, P. F., Keres, D., & Faucher-Giguere, C.-A. 2014, arXiv, in press, 1407.7039
- Van Leer, B. 1977, *Journal of Computational Physics*, 23, 276
- van Leer, B. 1984, *SIAM Journal on Scientific and Statistical Computing*, 5, 1
- Vogelsberger, M., Genel, S., Sijacki, D., Torrey, P., Springel, V., & Hernquist, L. 2013, *MNRAS*, in press, arXiv:1305.2913
- Vogelsberger, M., Sijacki, D., Keres, D., Springel, V., & Hernquist, L. 2012, *MNRAS*, 425, 3024
- Wadsley, J. W., Stadel, J., & Quinn, T. 2004, *New Astronomy*, 9, 137
- Wadsley, J. W., Veeravalli, G., & Couchman, H. M. P. 2008, *MNRAS*, 387, 427
- Whitehurst, R. 1995, *MNRAS*, 277, 655
- Woodward, P., & Colella, P. 1984, *Journal of Computational Physics*, 54, 115
- Xu, G. 1997, *MNRAS*, 288, 903
- Zel'dovich, Y. B. 1970, *A&A*, 5, 84
- Zhang, Y., Anninos, P., Norman, M. L., & Meiksin, A. 1997, *ApJ*, 485, 496
- Zingale, M., et al. 2002, *ApJS*, 143, 539

## APPENDIX A: DEALING WITH PATHOLOGICAL PARTICLE CONFIGURATIONS

In general, our matrix-based methods for solving the least-squares particle-centered gradients (see § 2.2) are very robust, and can deal with arbitrary configurations of particles within the kernel (for example, the proof that the method exactly recovers linear gradients is trivial and independent of the particle spatial locations within the kernel).

However, in all quasi-Lagrangian methods, there is some possibility that the mesh or particle distribution becomes severely irregular in a way that requires careful consideration (or else errors may increase, and/or the method may crash). In this case, the proof above makes an implicit assumption – that the matrix in Eq. 14  $\mathbf{E}_i^{\alpha\beta} \equiv \sum_j (\mathbf{x}_j - \mathbf{x}_i)^\alpha (\mathbf{x}_j - \mathbf{x}_i)^\beta \psi_j(\mathbf{x}_i)$  is non-singular. Consider, for example, the following pathological case. Since the kernel is compact, there are a finite number  $N_{\text{NGB}}$  of particles inside it; it is conceivable that all  $N_{\text{NGB}}$  particles lie exactly along one axis (in a 3D simulation). In this case  $\mathbf{E}_i$  will be singular, and the gradients in the perpendicular directions will be undefined. This is physically correct, after all, since in this configuration there is no information on these directions! This is directly analogous to the case in moving mesh codes, when a cell becomes highly deformed so has a very large axis ratio in one direction (leading to divergences and ill-defined gradients).

Such situations are very rare, and clearly pathological, but they can occur in highly non-linear, large simulations (like cosmological simulations) and we must implement some method to deal with them. More likely, we will have situations which are “close to” singular (e.g. the particles are all on one axis to within some deviation  $|\epsilon| \ll h$ ), in which case the method is formally accurate (the matrix is invertible and stable), but the numerical “noise” can be very large (since the inferred gradients become dominated by small offsets of the particles positions).

Fortunately, there is a well-studied means to properly define “pathological” here, which is given by the condition number  $N_{\text{cond}}$  of the (weighted) position moments matrix. That matrix is just  $\mathbf{E}_i$  (Eq. 14), and the condition number is:

$$N_{\text{cond},i} \equiv \nu^{-1} \left[ \left\| \mathbf{E}_i^{-1} \right\| \cdot \left\| \mathbf{E}_i \right\| \right]^{1/2} \quad (\text{A1})$$

$$\left\| \mathbf{E}_i \right\| \equiv \sum_{\alpha=1}^{\nu} \sum_{\beta=1}^{\nu} \left| \mathbf{E}_i^{\alpha\beta} \right|^2 \quad (\text{A2})$$

where  $\nu$  is the number of dimensions. It is easy to verify that for a truly singular matrix  $\mathbf{E}_i$ ,  $N_{\text{cond}} \rightarrow \infty$ ; at the opposite extreme, if  $\mathbf{E}_i$  were the most “perfectly invertible” matrix (the identity matrix),  $N_{\text{cond}} = 1$ .

For any configuration of particles, we can measure  $N_{\text{cond}}$ ; the problem then reduces to how to deal with large  $N_{\text{cond}} \gg 1$ . There are many possible choices. In moving-mesh codes, the usual approach is to “re-mesh” (split the cells until they have regular aspect ratios; see Springel 2010); the analogue in particle-based codes is to split particles, inserting new particles in the directions which are under-sampled in some regular fashion (see e.g. Maron et al. 2012). We can do this (see § E). Unfortunately, these are highly diffusive operations which introduce their own lower-order errors; moreover, most of the time in the realistic cases we study here, the pathology is transient (it is a random coincident alignment of particles, with well-sampled particles waiting “just outside” the kernel, rather than something systematic and persistent). So in most cases the problem can be addressed without adding errors and diffusion by simply extending the particle search until particles are found in the under-

sampled directions and  $N_{\text{cond}}$  is reduced. We therefore adopt the following approach: if  $N_{\text{cond}}$  exceeds some critical  $N_{\text{cond}}^{\text{crit}} \gg 1$ , then we iteratively expand the kernel (increase  $N_{\text{NGB}}^{\text{eff}}$ ) in small increments until we reduce  $N_{\text{cond}}$  below

$$N_{\text{cond}} \leq N_{\text{cond}}^{\text{crit}} \text{MAX} \left( 1, \alpha_{\text{cn}} \left[ 1 - \left( \frac{N_{\text{NGB}}^{\text{eff}}}{N_{\text{NGB}}^0} \right)^2 \right] \right) \quad (\text{A3})$$

where  $\alpha_{\text{cn}} \approx 10$  and  $N_{\text{cond}}^{\text{crit}} \approx 100 - 1000$  are set by our own experiments (we find this does the best job of simultaneously minimizing errors and diffusion while stabilizing the code),  $N_{\text{NGB}}^0$  is the “default” number of neighbors, and the second term exists only to prevent  $N_{\text{NGB}}^{\text{eff}}$  from running away if, indeed, it cannot find a reduction in  $N_{\text{cond}}$  with a reasonable augmentation to the neighbor number.

In the extremely rare cases where this cannot reduce  $N_{\text{cond}}$  below some threshold, say  $\sim 10 N_{\text{cond}}^{\text{crit}}$ , we simply have the code issue a warning and proceed by replacing the gradient estimators (in both the standard gradient estimation and the equation of motion or definition of the “effective face” areas for the Riemann problem) for that particle and timestep with the standard SPH gradient estimators, so

$$(\nabla q)_i \approx (\nabla q)_i^{\text{SPH}} \equiv \sum_j \frac{1}{\omega_j} q_j \nabla_i W_{ij}(h_i) \quad (\text{A4})$$

These gradient estimators have low-order errors; however, they are stable in irregular/pathological particle configurations. For example, for the case above (all particles aligned in one axis), this will simply return a gradient of zero in the perpendicular directions. We find that using this method, instead of particle splitting, in these extreme cases, is sufficient to restore stability and produces still less diffusion than particle splitting. However, we stress that this is *extremely* rare, occurring only once (for a small number of timesteps around the central caustic in the Zeldovich problem when the analytic density diverges) in all the tests we run.

## APPENDIX B: ON THE SLOPE-LIMITING PROCEDURE FOR UNSTRUCTURED, MESHLESS RIEMANN 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 requires some slope or flux-limiting 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. Barth & Jespersen 1989). In performing the face reconstruction of some arbitrary quantity  $\phi^i$  for particle  $i$ , we replace the “true” (matrix-evaluated) gradient  $\nabla \phi_{\text{true}}^i$  with an effective (slope-limited) gradient  $\nabla \phi_{\text{lim}}^i$ :

$$\nabla \phi_{\text{lim}}^i = \alpha_i \nabla \phi_{\text{true}}^i \quad (\text{B1})$$

where

$$\alpha_i \equiv \text{MIN} \left[ 1, \beta_i \text{MIN} \left( \frac{\phi_{ij,\text{ngb}}^{\text{max}} - \phi_i}{\phi_{ij,\text{mid}}^{\text{max}} - \phi_i}, \frac{\phi_i - \phi_{ij,\text{ngb}}^{\text{min}}}{\phi_i - \phi_{ij,\text{mid}}^{\text{min}}} \right) \right] \quad (\text{B2})$$

where  $\phi_{ij,\text{ngb}}^{\text{max}}$  and  $\phi_{ij,\text{ngb}}^{\text{min}}$  are the maximum and minimum values of  $\phi_j$  among all neighbors  $j$  of the particle  $i$ , and  $\phi_{ij,\text{mid}}^{\text{max}}$ ,  $\phi_{ij,\text{mid}}^{\text{min}}$  are the

maximum and minimum values (over all pairs  $ij$  of the  $j$  neighbors of  $i$ ) of  $\phi$  re-constructed on the “ $i$  side” of the interface between particles  $i$  and  $j$  (i.e.  $\phi_{ij,\text{mid}}^{\text{max}} = \text{MAX}[\phi_i + \nabla\phi_{\text{true}} \cdot (\mathbf{x}_{\text{face},ij} - \mathbf{x}_i)]$ ).

As noted by Balsara (2004), the constant  $\beta$  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  $\beta$  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 § A, 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

$$\beta_i = \text{MAX}[\beta_{\text{min}}, \beta_{\text{max}} \text{MIN}(1, N_{\text{cond}}^{\text{crit}}/N_{\text{cond}}^i)] \quad (\text{B3})$$

with  $\beta_{\text{min}} = 1$ ,  $\beta_{\text{max}} = 2$ . We find that  $\beta_{\text{min}} < 1$  does not much improve stability, but does begin to introduce noticeable diffusion of discontinuities, while  $\beta_{\text{max}} > 2$  does not much improve accuracy and leads can lead to problems with stability in very strong interacting shocks (though for most other problems,  $\beta_{\text{max}} = 4$  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  $\phi_i - \phi_{ij,\text{mid}}^{\text{min}}$  and  $\phi_{ij,\text{mid}}^{\text{max}} - \phi_i$  by the value  $|\nabla\phi_{\text{true}} \cdot |\mathbf{x}_{\text{face},ij} - \mathbf{x}_i|^{\text{max}}$  (where  $|\mathbf{x}_{\text{face},ij} - \mathbf{x}_i|$  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  $|\mathbf{x}_{\text{face},ij} - \mathbf{x}_i|^{\text{max}}$  can be directly calculated, but given our other definitions is well-approximated by 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. This sort of issue has been seen before, especially for unstructured point distribution (see e.g. Mocz et al. 2014).

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  $\phi_{ij,\text{mid}}$  (the re-constructed value on the “ $i$  side”) as above, using the slope-limited gradients  $\nabla\phi_{\text{lim}}$ . We then apply a second pair-wise limiter to this, replacing our initial estimate  $\phi_{ij,\text{mid}}^0$  with a limited  $\phi'_{ij,\text{mid}}$  based on the values

of  $\phi_i$  and  $\phi_j$ :

$$\begin{aligned} \phi'_{ij,\text{mid}} &= \begin{cases} \phi_i & (\phi_i = \phi_j) \\ \text{MAX}(\phi_-, \text{MIN}[\bar{\phi}_{ij} + \delta_2, \phi_{ij,\text{mid}}^0]) & (\phi_i < \phi_j) \\ \text{MIN}(\phi_+, \text{MAX}[\bar{\phi}_{ij} - \delta_2, \phi_{ij,\text{mid}}^0]) & (\phi_i > \phi_j) \end{cases} \\ \phi_- &= \begin{cases} \phi_{\text{min}} - \delta_1 & (\text{SIGN}(\phi_{\text{min}} - \delta_1) = \text{SIGN}(\phi_{\text{min}})) \\ \frac{\phi_{\text{min}}}{1 + \delta_1/|\phi_{\text{min}}|} & (\text{SIGN}(\phi_{\text{min}} - \delta_1) \neq \text{SIGN}(\phi_{\text{min}})) \end{cases} \\ \phi_+ &= \begin{cases} \phi_{\text{max}} + \delta_1 & (\text{SIGN}(\phi_{\text{max}} + \delta_1) = \text{SIGN}(\phi_{\text{max}})) \\ \frac{\phi_{\text{max}}}{1 + \delta_1/|\phi_{\text{max}}|} & (\text{SIGN}(\phi_{\text{max}} + \delta_1) \neq \text{SIGN}(\phi_{\text{max}})) \end{cases} \\ \bar{\phi}_{ij} &\equiv \phi_i + \frac{|\mathbf{x}_{ij} - \mathbf{x}_i|}{|\mathbf{x}_j - \mathbf{x}_i|} (\phi_j - \phi_i) \\ \phi_{\text{min}} &\equiv \text{MIN}(\phi_i, \phi_j) \\ \phi_{\text{max}} &\equiv \text{MAX}(\phi_i, \phi_j) \\ \delta_1 &\equiv \psi_1 |\phi_i - \phi_j| \\ \delta_2 &\equiv \psi_2 |\phi_i - \phi_j| \end{aligned} \quad (\text{B4})$$

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  $\psi_1$  and  $\psi_2$ . Many popular slope limiters can be expressed as variations of these parameters: for example, the monotonized central (Van Leer 1977), minmod and superbee (Roe 1986), Koren & van der Maarel (1993), and Sweby (1984) 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  $\beta$ , 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  $\phi_{\text{lim}}(r) = \phi_{ij,\text{mid}} \psi(r)$  of  $r = (\phi_i - \phi_{i-1})/(\phi_{i+1} - \phi_i)$ , where following Cha et al. (2010) we take  $\phi_{i-1} = \phi_j$  and  $\phi_{i+1}$  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  $\psi = 2r$  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  $\phi_i$  and  $\phi_j$  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 numerical 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 C: THE CHOICE OF RIEMANN SOLVER

To actually solve the Riemann problem, there are many methods commonly used in the literature (see e.g. Toro 1997). We have performed some limited experiments ourselves, but have not rigorously explored the possible parameter space.

We have implemented an exact Riemann solver, following Toro (1997). This uses an iterative procedure to exactly solve the Riemann problem for gas described by a local polytropic index. Because the solution is numerical, we must define some numerical tolerance for the deviation between iterations when convergence is assumed; we set this to  $10^{-6}$  in the pressure at the contact state  $P_*$ . In general, using values as low as  $10^{-12}$  we see no improvements beyond this in tests here. Unfortunately, while this method should in principle always return the exact solution, in practice numerical errors mean that, very rarely, the procedure can numerically diverge or fail to converge in many iterations ( $> 1000$ ). It is also very expensive to use this for every inter-particle Riemann problem.

We therefore have also implemented a standard approximate HLLC Riemann solver (see Toro 1999; Miyoshi & Kusano 2005). This method is not exact, but it is accurate at the order we require (and exactly conservative); moreover such methods are extremely well-tested in the literature. HLLC solvers break the problem into a simple set of waves/fronts, and require some initial “guess” for certain wavespeeds; we have experimented with a few choices for this following Roe (1981); Gaburov & Nitadori (2011); Toro (1997). In general, even when we perform convergence tests, we see no measurable loss of accuracy using the HLLC solution as opposed to the exact solver. And the HLLC solver is much faster, and does not require a polytropic index, so can be trivially generalized to non-ideal equations of state. However, in rare examples, this can fail, because of bad estimates for the wavespeeds. This failure is usually assessed by checking whether the pressure returned is everywhere positive in the approximate solution.

The Riemann solution method therefore proceeds as follows. We begin with our usual piecewise-linear (second-order) reconstruction of left and right states. We then attempt the HLLC Riemann solver. Within the HLLC solver, we first check Roe-average (usually most accurate) wave-speed estimate; if this is bad (returns  $P_* \leq 0$ ), we check the simpler wave-speed estimate from Toro (1999) as used in Gaburov & Nitadori (2011) (their Eq. 34-36); if this is bad, we check the Rusanov or TVD Lax-Friedrich primitive-variable wave speed estimate. If no good solutions are found, we use the exact Riemann solver. If (in very rare cases) this fails to converge after 1000 iterations, we go back to the reconstruction step and use a piecewise-constant (first-order) reconstruction, then repeat the process of searching for solutions. If this fails, we print a warning and exit the code. However, in all our tests, we find we always obtain a valid solution so long as physically allowed values for input states are used.

### APPENDIX D: EXPLICIT THERMAL ENERGY EVOLUTION AND ENERGY-ENTROPY SWITCHES AT EXTREMELY HIGH MACH NUMBERS

When a Riemann solver is used in an exactly-conservative method, flows which are strongly kinetic-energy dominated (very cold and super-sonic in the frame in which the Riemann problem is solved) exhibit spurious heating in the adiabatic parts of the flow (Ryu et al. 1993; Bryan et al. 1995; Steinmetz & White 1997). This ultimately stems from the Riemann problem’s use of and conservation of total energy; if the Mach number is high ( $\sim 10^5$ , as in the Zeldovich problem we simulate below), then very small truncation errors (part in  $\sim 10^{10}$ ) appear in the thermal energy. This problem is discussed at length in Springel (2010) (§ 3.5); it is ubiquitous in cosmology in the early stages of structure formation (where the velocities from gravity produce extremely high Mach numbers), corrupting simulations unless some fix is applied.

We follow an approach similar to the Bryan et al. (1995) “dual energy formalism,” whereby we explicitly evolve the internal energy, in addition to total energy, and when the motion is sufficiently supersonic the temperature and pressure are set based on the results of this equation. Following Gaburov & Nitadori (2011), § 3.4, this amounts to explicitly evolving the internal energy  $U$  (or internal energy per unit mass  $u = U/m$  as

$$\frac{dU}{dt} = \frac{dE}{dt} - \mathbf{v} \cdot \frac{d\mathbf{P}}{dt} + \frac{\mathbf{v} \cdot \mathbf{v}}{2} \frac{dm}{dt} \quad (\text{D1})$$

where  $E = U + \mathbf{P} \cdot \mathbf{P}/(2m)$  is the “hydrodynamic total energy,”  $\mathbf{P} = m\mathbf{v}$  the momentum, and  $m$  the particle mass. Note that when this is done, total energy is no longer conserved to machine accuracy, but to the truncation error of the time-integration scheme. However, internal energy is evolved more accurately (otherwise, any errors in the solution are simply shifted into the internal energy). In fact, for every test problem here, we find this produces at least comparable accuracy to the explicitly energy-conserving formalism; and for flows with gravity, where the internal energy would otherwise be determined by the difference between two large numbers, it gives substantially improved accuracy and numerical stability, and actually *better* overall energy conservation (to  $\ll 1\%$  accuracy) in most cases. The reason for this counter-intuitive result is that when long-range forces like gravity are present, it is no longer possible to conserve total energy to machine precision in any case, because the long-range interactions cannot be perfectly pair-wise symmetric unless an explicit  $N^2$  (i.e. pair wise) method for gravity (with a single time-step) is adopted; this is impossibly expensive for anything but simulations with a tiny number of particles.

With this choice, most of the problems described above are solved. However, it is possible in the most extreme situations (like the Zeldovich problem) that the numerical convergence accuracy (part in  $\sim 10^8$ ) in the Riemann solver still leads to large errors in the thermal energy equation. We can in this case follow Ryu et al. (1993) and Springel (2010), and explicitly calculate the evolution of the system as if it were purely adiabatic in each timestep (see Springel 2010, § 3.5), with a switch to decide when this solution is used. Experimenting with this, we find that a Mach number switch is unnecessary and can create more problems than it solves (the same is true in AREPO; V. Springel, private communication). However, an energy-based switch is, in rare situations, useful. In each timestep, we determine the expected thermal energy  $E_{\text{therm}}$  based on the usual update; we compare this to the gravitational energy associated with motion across the particle size ( $\delta E_{\text{grav}} = m_i |\mathbf{a}_{\text{grav},i}| h_i$ ) and maximum kinetic energy  $\delta E_{\text{kin}}^{\text{max}}$  of all neighbor cells in the rest-frame of the current cell  $i$ . If  $E_{\text{therm}} < \alpha_{\text{kin}} (\delta E_{\text{kin}}^{\text{max}} + E_{\text{therm}})$ , or

$E_{\text{therm}} < \alpha_{\text{grav}} \delta E_{\text{grav}}$ , we use the entropy-based evolution. Because our method is Lagrangian (which minimizes these sorts of errors to begin with), and because of the energy evolution choice above, we can set  $\alpha_{\text{kin}}$  and  $\alpha_{\text{grav}}$  to very conservative (low) values, in the simulations here  $\approx 0.001$ , which means they are almost never triggered but manage to trap the extremely rare pathological cases encountered in some problems.

All of these are choices, and of course it is straightforward to use the method we propose without such switches (and the total energy evolution). However, we find negligible penalty and considerable advantages in this particular form of the method.

## APPENDIX E: PARTICLE SPLITTING AND MERGING IN GIZMO

In some problems, it may be necessary to split or merge particles, especially when mass fluxes between them are allowed (as in the MFV method here). For example, gas particles at the center vs. in the outskirts of a galactic disk may eventually (over many orbits) develop large (more than order-of-magnitude) differences in their masses. This is fine (and correct, given the nature of the method) if the gas flows are sufficiently smooth, but if galactic winds from say, SNe explosions suddenly expel mass from the center at high velocities, this will lead to particles with very different masses suddenly interacting. The hydro method is formally robust to this (although if the differences are large enough, truncation error in fluxes from one particle could lead to unphysical quantities in the other). However, if self-gravity is also included, this can produce unacceptably large  $N$ -body scattering effects.

To deal with this, we have implemented a simple particle splitting/merging algorithm in GIZMO, although we caution that it is not expected to be the most optimal possible algorithm. If a particle falls below a mass  $= \epsilon_{\text{min}} m_{\text{min}}(t_0)$  (where  $m_{\text{min}}(t_0)$  is the minimum mass over all particles in the initial conditions), and is the least massive particle currently among its entire neighbor list, it is merged with the second-least-massive particle among the neighbors. The merger is straightforward: the less-massive particle is deleted and conservation requires that the more massive particle inherits the summed mass, momentum, and energy (and their time rates-of-change). For quantities like the signal velocity and smoothing length, the larger of the two is chosen, but these will be re-initialized in the next active timestep. The updated particle is moved to the center-of-mass position of the pair. The merge operation is done only on timesteps where the neighbor/gravity tree is being reconstructed, so that no errors in gravity or neighbor searches are introduced. We adopt the somewhat ad-hoc choice  $\epsilon_{\text{min}} = 0.5$ .

Similarly, if a particle is above a mass  $= \epsilon_{\text{max}} m_{\text{max}}(t_0)$  (where  $m_{\text{max}}(t_0)$  is the maximum mass over all particles in the initial conditions), and is the most massive particle among its neighbors, it is split into two particles. Each particle has half the mass and inherits the specific (per-unit-mass) properties of the parent. This is straightforward; the ambiguity in particle splitting comes from the positions of the particles. They cannot be placed at identically the parent location, but must be separated by some small amount. However doing so in a way that does not seed fluctuations in the volumetric quantities is highly non-trivial. Here, we adopt a very simple prescription: the two particles are separated by the minimum of  $h_i/8$  or  $|r_{\text{near}}|/3$ , where  $|r_{\text{near}}|$  is the distance within the kernel to the closest neighbor particle. They are each moved this distance, in opposite directions along an axis perpendicular to the particle number density gradient (to minimize the perturbation to volumetric quantities).

We note that these operations are both noisy and diffusive, and we recommend against particle merging/splitting unless absolutely necessary. That said, we have run all the test problems in this paper with and without such splitting and find very little difference in almost every case (because very few particles would be eligible). However for at least one problem – the isolated disk with the full physics of stellar feedback from the FIRE models – we simply cannot run the problem using the MFV method without it crashing, if we do not invoke particle splitting and merging (the strong galactic winds led to exactly the  $N$ -body problems described above). The methods for splitting/merging merit serious, detailed examination in future work, as there are almost certainly ways to improve the simple algorithm we invoke here.

## APPENDIX F: THE SPH IMPLEMENTATION IN GIZMO

As discussed in the text, we can run GIZMO as an SPH code, if desired. We implement two “default” versions of SPH, and use them throughout the text, so we describe their properties here.

### F1 “Traditional” SPH (TSPH)

The “traditional” SPH (TSPH) implementation in GIZMO is particularly simple. As noted in the text, nearly everything in the code remains identical whether we run in SPH mode or one of our new modes. Here we outline the method insofar as it requires something distinct from our other methods.

The TSPH implementation falls within the general class of manifestly conservative, Lagrangian-derived SPH schemes outlined in Hopkins (2013). Specifically it is a “density-energy” scheme (where the internal energy is explicitly evolved). As shown therein, the choice of “density-energy” or “density-entropy” scheme (as in GADGET-2) gives essentially identical results *when a Lagrangian-derived scheme is used*, since both simultaneously conserve energy and entropy in global timesteps; we have explicitly confirmed this by comparison to a density-entropy formulation in the tests here. Since the choice is a matter of convenience, we find it more naturally aligns with our other methods, and allows a more flexible equation of state, to use the “density-energy” form.

We also determine the smoothing length in the same manner as our other methods (§ 2.6) based on the particle number density; in 3D, this means  $(4\pi/3) h_i^3 n_i = N_{\text{NGB}}$  (where  $n_i = \sum W(\mathbf{x}_j - \mathbf{x}_i, h_i)$ ). This corresponds to the choice  $\bar{x} = 1$  in Hopkins (2013), which we argue there provides the most stable and accurate results (as opposed to a “constant mass in kernel” or “constant energy in kernel” weighting). We use a cubic spline kernel with  $N_{\text{NGB}} = 4, 16, 32$  in 1, 2, 3 dimensions; this is the standard in most traditional SPH formulations.

In TSPH the density is estimated by kernel-smoothing as:

$$\rho_i^{\text{TSPH}} \equiv \bar{\rho}_i = \sum_j m_j W(\mathbf{x}_i - \mathbf{x}_j, h_i) \quad (\text{F1})$$

the pressure is then determined from the density as  $P_i^{\text{TSPH}} = P(\bar{\rho}_i, u_i) = (\gamma - 1) \bar{\rho}_i u$  (for a polytropic equation of state).

Recall, we need to replace our flux calculations. The mass flux in SPH is identically zero. With the choices above, the momentum and internal energy fluxes derived from the particle Lagrangian (see

Hopkins 2013, Eq. 12-13 therein) are

$$\frac{d\mathbf{P}_i}{dt} = - \sum_j m_i m_j \left[ \frac{P_i}{\bar{\rho}_i^2} f_{i,j} \nabla_i W_{ij}(h_i) + \frac{P_j}{\bar{\rho}_j^2} f_{j,i} \nabla_i W_{ij}(h_j) \right] \quad (\text{F2})$$

$$\frac{dE}{dt} = \mathbf{v}_i \cdot \frac{d\mathbf{P}_i}{dt} - \sum_j m_i m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \left[ \frac{P_i}{\bar{\rho}_i^2} f_{i,j} \nabla_i W_{ij}(h_i) \right] \quad (\text{F3})$$

$$f_{i,j} = 1 - \frac{1}{m_j} \left( \frac{h_i}{n_i \nu} \frac{\partial \bar{\rho}_i}{\partial h_i} \right) \left[ 1 + \frac{h_i}{n_i \nu} \frac{\partial n_i}{\partial h_i} \right]^{-1} \quad (\text{F4})$$

$$\frac{\partial n_i}{\partial h_i} = - \sum_j \frac{1}{h_i} \left( \nu W_{ij}(h_i) + u_{ij} \frac{\partial W(u)}{\partial u} \Big|_{u=u_{ij}} \right) \quad (\text{F5})$$

$$\frac{\partial \bar{\rho}_i}{\partial h_i} = - \sum_j \frac{m_j}{h_i} \left( \nu W_{ij}(h_i) + u_{ij} \frac{\partial W(u)}{\partial u} \Big|_{u=u_{ij}} \right) \quad (\text{F6})$$

where we abbreviate  $W(\mathbf{x}_i - \mathbf{x}_j, h_k) = W_{ij}(h_k)$ ,  $\nu$  is the number of dimensions, and  $u_{ij} \equiv |\mathbf{x}_j - \mathbf{x}_i|/h_i$ .

We also require artificial diffusion terms in SPH, to handle shocks (the equations above only hold for adiabatic flows). In TSPH this is just artificial viscosity, using the Gingold & Monaghan (1983) prescription with a Balsara (1989) switch. This contributes an additional term to the equations of motion if and only if particles  $i$  and  $j$  are approaching, i.e.  $(\mathbf{v}_i - \mathbf{v}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j) < 0$ , in which case:

$$\frac{d\mathbf{P}_i}{dt} = \sum_j \alpha_{ij} \mu_{ij} (c_{ij} - 2\mu_{ij}) m_i m_j \frac{\nabla_i W_{ij}(h_i) + \nabla_i W_{ij}(h_j)}{\bar{\rho}_i + \bar{\rho}_j} \quad (\text{F7})$$

$$\frac{dE_i}{dt} = \frac{1}{2} (\mathbf{v}_i + \mathbf{v}_j) \cdot \frac{d\mathbf{P}_i}{dt} \quad (\text{F8})$$

$$\mu_{ij} = \frac{h_{ij} (\mathbf{v}_i - \mathbf{v}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j)}{|\mathbf{x}_i - \mathbf{x}_j|^2 + 0.0001 h_{ij}^2} \quad (\text{F9})$$

$$c_{ij} = \frac{c_{s,i} + c_{s,j}}{2}, \quad \alpha_{ij} = \frac{\alpha_i + \alpha_j}{2}, \quad h_{ij} = f_{\text{kernel}} \frac{h_i + h_j}{2} \quad (\text{F10})$$

$$\alpha_i = \frac{\alpha_{\text{av}} |(\nabla \cdot \mathbf{v})_i|}{|(\nabla \cdot \mathbf{v})_i| + |(\nabla \times \mathbf{v})_i| + 0.0001 c_{s,i} / (f_{\text{kernel}} h_i)} \quad (\text{F11})$$

where  $\alpha_{\text{av}} = 1$  is constant everywhere,  $f_{\text{kernel}}$  depends on the kernel shape but  $= 1/2$  for the cubic spline here, and the velocity gradients in the Balsara (1989) switch are determined by our standard (least-squares) gradient procedure.

Note that, by virtue of our desire to make this implementation as consistent as possible with the rest of GIZMO, there are already a number of subtle improvements of this method over SPH implementations like in GADGET-2 and GASOLINE. Our standard (least-squares) gradient estimators are used for predict steps and for quantities like the Balsara (1989) switch; these are substantially more accurate than the usual SPH gradient estimators (based on the kernel gradient). We use our manifestly-conservative adaptive timestepping scheme, instead of relying solely on integration accuracy. We include the neighbor and particle-approach-based timestep limiter, which prevents spurious particle inter-penetration in strong shocks. The smoothing length is based on particle number density (not mass density), reducing errors when there are particles of different masses in the same kernel. Gravity includes fully-conservative adaptive force softening. And we use a Lagrangian-derived density-energy formulation, which is necessary to prevent additional errors whenever the smoothing lengths vary in SPH; compared to methods like GASOLINE and SPHS, which use a non-Lagrangian (hence non-conservative) SPH equation of motion, the choice here at least ensures that entropy and energy are simultaneously conserved in adiabatic flows.

## F2 “Modern” SPH (PSPH)

Our “modern” SPH method in GIZMO builds on the TSPH method, using higher-order kernels, pressure-based formulations of the equations of motion, more accurate gradients, and higher-order switches for dissipation terms.

The method is a “pressure-energy” scheme, following Hopkins (2013), again with  $\bar{x} = 1$ ; so we follow internal energy, and determine  $h_i$  in the exact same manner. As noted above, “pressure-energy” and “pressure-entropy” schemes are essentially equivalent if Lagrangian-derived. However, in a pressure-entropy scheme, because the particle-entropies enter the pressure in a non-linear fashion, radiative cooling (if enabled) must be followed in a somewhat complicated iterative manner to ensure proper energy conservation is maintained; pressure-energy formulations avoid this.

To reduce the E0 errors, we follow standard practice and increase the number of neighbors to  $N_{\text{NGB}} = 128$  in 3D (our default, though we will vary this in the text). This cannot be done using the cubic spline kernel without suffering the pairing instability, so we go to a higher-order (in this case, quintic spline) kernel, as advocated in Dehnen & Aly (2012); we revert to the cubic spline when we run in PSPH mode with  $N_{\text{NGB}} = 32$ . The quintic spline is given by:

$$W(q, h_i) = \frac{3^7}{40\pi h_i^3} \times \begin{cases} (1-q)^5 - 6\left(\frac{2}{3}-q\right)^5 + 15\left(\frac{1}{3}-q\right)^5 & (0 \leq q < \frac{1}{3}) \\ (1-q)^5 - 6\left(\frac{2}{3}-q\right)^5 & (\frac{1}{3} \leq q < \frac{2}{3}) \\ (1-q)^5 & (\frac{2}{3} \leq q < 1) \\ 0 & (q \geq 1) \end{cases} \quad (\text{F12})$$

where  $q \equiv |\mathbf{x} - \mathbf{x}_i|/h_i$ . Note that we have also experimented with the Wendland kernels in Dehnen & Aly (2012); for the neighbor number here, both their experiments and ours find essentially identical behavior to the quintic spline kernel.

In PSPH, both the density and pressure are estimated by kernel smoothing:

$$\rho_i^{\text{PSPH}} \equiv \bar{\rho}_i = \sum_j m_j W_{ij}(h_i) \quad (\text{F13})$$

$$P_i^{\text{PSPH}} \equiv \bar{P}_i = \sum_j (\gamma - 1) m_j u_j W_{ij}(h_i) \quad (\text{F14})$$

The momentum and energy equations become

$$\begin{aligned} \frac{d\mathbf{P}_i}{dt} &= - \sum_{j=1}^N (\gamma - 1)^2 m_i m_j u_j \left[ \frac{f_{ij}}{\bar{P}_i} \nabla_i W_{ij}(h_i) + \frac{f_{ji}}{\bar{P}_j} \nabla_i W_{ij}(h_j) \right] \\ \frac{dE_i}{dt} &= \mathbf{v}_i \cdot \frac{d\mathbf{P}_i}{dt} - \sum_{j=1}^N (\gamma - 1)^2 m_i m_j u_j \frac{f_{ij}}{\bar{P}_i} (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W_{ij}(h_i) \\ f_{ij} &= 1 - \left( \frac{h_i}{\nu(\gamma - 1) \bar{n}_i m_j u_j} \frac{\partial \bar{P}_i}{\partial h_i} \right) \left[ 1 + \frac{h_i}{\nu n_i} \frac{\partial n_i}{\partial h_i} \right]^{-1} \\ \frac{\partial n_i}{\partial h_i} &= - \sum_j \frac{1}{h_i} \left( \nu W_{ij}(h_i) + u_{ij} \frac{\partial W(u)}{\partial u} \Big|_{u=u_{ij}} \right) \\ \frac{\partial \bar{P}_i}{\partial h_i} &= - \sum_j \frac{(\gamma - 1) m_j u_j}{h_i} \left( \nu W_{ij}(h_i) + u_{ij} \frac{\partial W(u)}{\partial u} \Big|_{u=u_{ij}} \right) \end{aligned} \quad (\text{F15})$$

We again require artificial diffusion terms. For the artificial viscosity, we use the higher-order switch from Cullen & Dehnen (2010), as updated in Hopkins et al. (2013b). Once again this

contributes if and only if particles  $i$  and  $j$  are approaching (i.e.  $(\mathbf{v}_i - \mathbf{v}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j) < 0$ ):

$$\begin{aligned} \frac{d\mathbf{P}_i}{dt} &= \sum_j \alpha_{ij} \mu_{ij} (c_{ij} - \beta_b \mu_{ij}) m_i m_j \frac{\nabla_i W_{ij}(h_i) + \nabla_i W_{ij}(h_j)}{\bar{\rho}_i + \bar{\rho}_j} \\ \frac{dE_i}{dt} &= \frac{1}{2} (\mathbf{v}_i + \mathbf{v}_j) \cdot \frac{d\mathbf{P}_i}{dt} \\ \mu_{ij} &= \frac{(\mathbf{v}_i - \mathbf{v}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j)}{|\mathbf{x}_i - \mathbf{x}_j|}, \quad c_{ij} = \frac{c_{s,i} + c_{s,j}}{2}, \quad \alpha_{ij} = \frac{\alpha_i + \alpha_j}{2} \\ \alpha_i &= \text{MAX} \left( \frac{|\beta_\xi \xi_i^4 (\nabla \cdot \mathbf{v})_i|^2 \alpha_{0,i}(t)}{|\beta_\xi \xi_i^4 (\nabla \cdot \mathbf{v})_i|^2 + \text{Trace}(\mathbf{S}_i \mathbf{S}_i^T)}, \alpha_{\min} \right) \\ \xi_i &\equiv 1 - \frac{1}{\bar{\rho}_i} \sum_j \text{SIGN}[(\nabla \cdot \mathbf{v})_j] m_j W_{ij}(h_i) \end{aligned} \quad (\text{F16})$$

where  $\alpha_{0,i}(t)$  is set for each particle each timestep by evaluating  $\alpha_{\text{tmp}}$ :

$$\begin{aligned} \alpha_{\text{tmp}} &= \begin{cases} 0 & ((d[\nabla \cdot \mathbf{v}]/dt)_i \geq 0, \text{ or } (\nabla \cdot \mathbf{v})_i \geq 0) \\ \frac{\alpha_{\max} |(d[\nabla \cdot \mathbf{v}]/dt)_i|}{|(d[\nabla \cdot \mathbf{v}]/dt)_i| + \beta_c c_{s,i}^2 / (f_{\text{kern}} h_i)^2} & (\text{otherwise}) \end{cases} \\ \alpha_{0,i}(t + \Delta t) &= \begin{cases} \alpha_{\text{tmp}} & (\alpha_{\text{tmp}} \geq \alpha_{0,i}(t)) \\ \alpha_{\text{tmp}} + (\alpha_{0,i}(t) - \alpha_{\text{tmp}}) e^{-\beta_d \Delta t |\mathbf{v}_{\text{sig},i}| / (2 f_{\text{kern}} h_i)} & \\ \alpha_{\text{tmp}} & (\alpha_{\text{tmp}} < \alpha_{0,i}(t)) \end{cases} \end{aligned} \quad (\text{F17})$$

where after considerable experimentation we find the best mix of accuracy and stability with  $\alpha_{\min} = 0.02$ ,  $\alpha_{\max} = 2$ ,  $\beta_c = 0.7$ ,  $\beta_d = 0.05$ ,  $\beta_\xi = 1$ ,  $\beta_b = 1$ ,  $\mathbf{S}$  is the shear tensor (constructed from our standard velocity derivatives as described in Cullen & Dehnen 2010),  $f_{\text{kern}} = 1/3$  for the quintic spline kernel, and  $(d[\nabla \cdot \mathbf{v}]/dt)_i$  is evaluated using the method in Cullen & Dehnen (2010), which is essentially the same as our least-squares gradient estimation here, applied to the acceleration as well as velocity to obtain the time derivative. Note that there are some very small modifications of this scheme from Cullen & Dehnen (2010); these are motivated by our experiments in Hopkins et al. (2013b) and the tests in Hu et al. (2014); they allow the viscosity to be reduced more rapidly in high-shear regions when the flows are complicated (leading to improvements in turbulence), better maintain stability in strong shocks by enforcing a finite  $\alpha_{\min}$  (necessary in some of our tests), and enhance the detection of weak shocks from high-redshift cosmological structure formation.

We also include an artificial conductivity term, following Price (2008) with the improvements in Read & Hayfield (2012). This enters just the energy equation between  $i$  and  $j$ , when  $\tilde{v}_s > 0$  where  $\tilde{v}_s \equiv c_{s,i} + c_{s,j} - 3(\mathbf{v}_i - \mathbf{v}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j) / |\mathbf{x}_i - \mathbf{x}_j|$ . We then have

$$\frac{dE_i}{dt} = \alpha_C \sum_j m_i m_j \alpha_{ij} \tilde{v}_s (u_i - u_j) \frac{|P_i - P_j|}{P_i + P_j} \frac{\nabla_i W_{ij}(h_i) + \nabla_i W_{ij}(h_j)}{\bar{\rho}_i + \bar{\rho}_j} \quad (\text{F18})$$

Here  $\alpha_{ij}$  is a similar switch to the above for artificial viscosity; in fact we find essentially identical results using the same switch for both (which means the conductivity only is applied, correctly, in crossing flows). And we set the global constant  $\alpha_C$  to a relatively conservative value  $\alpha_C \approx 0.25$ ; together with the limiters in the equation this leads to greatly reduced diffusion compared to some prescriptions for conductivity in the literature (e.g. Shen et al. 2010).