

# SPHS: smoothed particle hydrodynamics with a higher order dissipation switch

J. I. Read<sup>1,2★</sup> and T. Hayfield<sup>1,3</sup>

<sup>1</sup>*Institute for Astronomy, Department of Physics, ETH Zürich, Wolfgang-Pauli-Strasse 16, CH-8093 Zürich, Switzerland*

<sup>2</sup>*Department of Physics and Astronomy, University of Leicester, University Road, Leicester LE1 7RH*

<sup>3</sup>*Max-Planck-Institut für Astronomie, Königstuhl 17, D-69117 Heidelberg, Germany*

Accepted 2012 February 24. Received 2012 February 23; in original form 2011 August 12

## ABSTRACT

We present a novel implementation of smoothed particle hydrodynamics that uses the spatial derivative of the velocity divergence as a higher order dissipation switch. Our switch – which is second order accurate – detects flow convergence *before* it occurs. If particle trajectories are going to cross, we switch on the usual SPH artificial viscosity, as well as conservative dissipation in all advected fluid quantities (e.g. the entropy). The viscosity and dissipation terms (that are numerical errors) are designed to ensure that all fluid quantities remain single valued as particles approach one another, to respect conservation laws, and to vanish on a given physical scale as the resolution is increased. SPHS alleviates a number of known problems with ‘classic’ SPH, successfully resolving mixing, and recovering numerical convergence with increasing resolution. An additional key advantage is that – treating the particle mass similarly to the entropy – we are able to use multimass particles, giving significantly improved control over the refinement strategy. We present a wide range of code tests including the Sod shock tube, Sedov–Taylor blast wave, Kelvin–Helmholtz Instability, the ‘blob test’ and some convergence tests. Our method performs well on all tests, giving good agreement with analytic expectations.

**Key words:** hydrodynamics – instabilities – methods: numerical.

## 1 INTRODUCTION

Smoothed particle hydrodynamics (SPH) is now widely used in almost all areas of theoretical astrophysics (Gingold & Monaghan 1977; Lucy 1977; Monaghan 1992). Its popularity has been largely driven by its Lagrangian nature that makes it manifestly Galilean invariant and geometry-free; its ease of implementation and the fact that it couples naturally to tree-gravity solvers that are currently the most efficient method for solving gravity (Monaghan 1992; Dehnen 2000; Price 2005; Rosswog 2009; Springel 2010b; Dehnen & Read 2011).

There are many different flavours of SPH used in the literature reflecting the above broad range of applications. The most common – that we shall call ‘classic’ SPH – is the fully conservative SPH implemented in the standard release of the GADGET-2 code (Springel & Hernquist 2002; Springel 2005).<sup>1</sup> Although classic SPH remains

a powerful numerical tool for solving the fluid equations, it suffers from slow numerical convergence (Springel 2010b), and a spurious surface tension at phase boundaries that inhibits fluid mixing (see e.g. Morris 1996a; Dills 1999; Ritchie & Thomas 2001; Agertz et al. 2007; Price 2008; Wadsley, Veeravalli & Couchman 2008; Read, Hayfield & Agertz 2010, hereafter RHA10; Springel 2010a).

In recent work, we demonstrated that mixing in classic SPH fails for two distinct reasons (RHA10). The first is a leading order error in the momentum equation, previously identified by Morris (1996b) and Dills (1999), that we called  $|E_0|$ . This can grow by orders of magnitude at flow boundaries, delaying the onset of instabilities. The second is a pressure discontinuity at flow boundaries, previously identified by Ritchie & Thomas (2001), Price (2008) and Wadsley et al. (2008), that we called the local mixing instability (LMI).<sup>2</sup> This leads to a large force error which manifests as a spurious surface tension. Both problems must be solved in order for mixing between fluids of different density or entropy to proceed correctly (see also Sections 3 and 4 in this paper).

\*E-mail: justin.inglis.read@gmail.com

<sup>1</sup> Slightly different implementations of this algorithm are also used in the literature, for example in the GASOLINE code (Wadsley, Stadel & Quinn 2004) and the HYDRA code (Couchman, Thomas & Pearce 1995). These are sufficiently similar to also be called ‘classic’ SPH.

<sup>2</sup> It is an instability since, even if we start in pressure equilibrium, an infinitesimal perturbation will cause a pressure discontinuity to form.

In RHA10, we presented some simple proof-of-concept solutions to both of these problems. We cured the LMI by using a weighted density estimate first proposed by Ritchie & Thomas (2001), and we showed that  $|\mathbf{E}_0|$  can be made arbitrarily small by brute-force so long as the method is stable to large neighbour number (this required introducing some new kernels). However, our resulting optimized smoothed particle hydrodynamics (OSPH) method required a neighbour number that scales linearly with the density contrast on the kernel scale. The OSPH pressure estimator is also biased in regions of the flow where entropy gradients are large. This leads to poor performance in strong blast wave tests (we demonstrate this in Appendix A).

The above problems with SPH have led to a welcome proliferation of new Lagrangian or pseudo-Lagrangian techniques in the literature, including a moving mesh (Springel 2010a), flux-based particle methods (Gaburov & Nitadori 2011), SPH using a Riemann solver (Inutsuka 2002; Cha, Inutsuka & Nayakshin 2010; Murante et al. 2011) and SPH using a Voronoi tessellation for the densities (Heß & Springel 2010). It has also led to an exploration of improved flavours of SPH that add additional dissipation terms to mitigate the surface tension effect, and use switches to reduce the dissipation away from flow boundaries and shocks<sup>3</sup> (e.g. Price 2008, 2012; Kawata et al. 2009; Cullen & Dehnen 2010; Rosswog 2010).

In this paper, we present a new flavour of SPH – SPHS<sup>4</sup> – that has the mixing performance of OSPH, but does not introduce prohibitive numerical cost. As in OSPH, we use a larger than normal neighbour number with a correspondingly higher order and stable kernel to reduce the force errors. However, instead of the expensive OSPH pressure estimator, we introduce a higher order dissipation switch to ensure that all fluid quantities are smooth by construction. We show that these simple changes to the SPH algorithm lead to converged results with increasing resolution, and excellent performance across a wide range of code tests. Our dissipation switch also allows us to successfully use multimass SPH particles.

This paper is organized as follows. In Sections 2–4, we present the SPHS method. In Section 5, we discuss our time-step criteria and multisteping scheme. In Section 6, we describe our implementation of SPHS in the GADGET-2 code (Springel 2005). In Section 7, we present a suite of tests for our new method that demonstrate that it can successfully model shocks, boundary instabilities and shear flows. We also check that it conserves momentum, energy and mass and discuss the numerical performance of the code. Finally, in Section 8, we present our conclusions.

## 2 THE SPHS EQUATIONS OF MOTION

In this paper, we consider solving the Euler equations in the absence of sinks or sources in the Lagrangian ‘entropy form’ (Springel & Hernquist 2002):

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v}, \quad (1)$$

$$\frac{d\mathbf{v}}{dt} = -\frac{\nabla P}{\rho}, \quad (2)$$

$$A = \text{const.}, \quad (3)$$

<sup>3</sup> Actually, some of these SPH flavours have been in use in the literature for quite some time (see e.g. Morris & Monaghan 1997).

<sup>4</sup> The second ‘S’ in SPHS stands for Switch.

closed by the ideal gas equation of state:

$$P = A(s)\rho^\gamma, \quad (4)$$

where  $\gamma, \rho, v$  and  $A$  are the adiabatic index, density, velocity and specific ‘entropy function’ of the flow, respectively. The function  $A(s)$  is a monotonic function of the specific entropy  $s$ . For adiabatic flow in the absence of sinks or sources,  $A$  is conserved. Thus equation (3) implicitly solves the energy equation. If required, the specific internal energy can be calculated from  $A$  and  $\rho$  as

$$u = \frac{A\rho^{\gamma-1}}{\gamma-1}. \quad (5)$$

Note that often  $A$  is referred to as the ‘entropy’ when really it is a monotonic function of the specific entropy. From here on we will adopt this convention also.

We use the discrete form of the above equations as in RHA10:

$$\rho_i = \sum_j^N m_j W_{ij}(|\mathbf{r}_{ij}|, h_i), \quad (6)$$

$$\frac{d\mathbf{v}_i}{dt} = -\sum_j^N \frac{m_j}{\rho_i \rho_j} [P_i + P_j] \nabla_i \bar{W}_{ij}, \quad (7)$$

$$P_i = A_i \rho_i^\gamma, \quad (8)$$

where  $m_i$  is the mass of particle  $i$ ,  $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ ,  $\bar{W}_{ij} = (1/2) [W_{ij}(h_i) + W_{ij}(h_j)]$  and  $W$  is a symmetric kernel that obeys the normalization condition:

$$\int_V W(|\mathbf{r} - \mathbf{r}'|, h) d^3 r' = 1, \quad (9)$$

and the property (for smoothing length  $h$ ):

$$\lim_{h \rightarrow 0} W(|\mathbf{r} - \mathbf{r}'|, h) = \delta(|\mathbf{r} - \mathbf{r}'|). \quad (10)$$

Note that we do not explicitly solve the continuity equation nor the energy equation. The continuity equation is implicitly solved by equation (6) since its time derivative satisfies a discrete form of equation (1) (see e.g. Price 2005). The energy equation is implicitly solved by advecting the entropy function  $A_i = \text{const.}$  along with the particles (Springel & Hernquist 2002).

We use a variable smoothing length  $h_i$  as in Springel & Hernquist (2002) that is adjusted to obey the following constraint equation:

$$\frac{4\pi}{3} h_i^3 n_i = N_n \quad \text{with} \quad n_i = \sum_j^N W_{ij}, \quad (11)$$

where  $N_n$  is the typical neighbour number (the number of particles inside the smoothing kernel,  $W$ ). The above constraint equation gives fixed mass inside the kernel if particle masses are all equal.

The above equations of motion manifestly conserve momentum, mass and entropy. They do not manifestly conserve energy, but the energy conservation is still extremely good as we will show in Section 7.7. A fully conservative form of SPH can be constructed by replacing equation (7) with equation (B1) (see Appendix B and Springel & Hernquist 2002). However, as shown in RHA10, this leads to a larger truncation error in the momentum equation. In Appendix B, we show that – for the test problems presented in this paper – the fully conservative form gives only a modest improvement in energy conservation while introducing significantly more diffusion for multiphase test problems. For this reason, we use the above set of equations as our default choice for SPHS.

So far, the above equations are very similar to classic SPH and thus will suffer from both the  $|\mathbf{E}_0|$  error and the LMI problems

described in Section 1. We now address each of these problems in turn in Sections 3 and 4.

### 3 ERRORS AND CONVERGENCE

The first problem with classic SPH is the  $|E_0|$  error in the momentum equation (RHA10). While this is minimized by using equation (7), the error is still present in SPHS. To see this, we can Taylor expand  $P_j$  about  $P_i$  in equation (7) to obtain<sup>5</sup>

$$\frac{dv_i}{dt} \simeq -\frac{2P_i}{h_i \rho_i} \sum_j \frac{m_j}{\rho_j} \nabla_i^x \bar{W}_{ij} - \frac{(\mathbf{V}_i \nabla_i) P_i}{\rho_i} + O(h), \quad (12)$$

where  $\mathbf{V}_i$  is a matrix that approximates the identity matrix, and  $\nabla_i^x = h_i \nabla$  is a dimensionless gradient operator. The left-hand term in equation (12) defines the dimensionless  $E_{0,i}$  error:

$$E_{0,i} = 2 \sum_j \frac{m_j}{\rho_j} \nabla_i^x \bar{W}_{ij}. \quad (13)$$

Taking the limit of infinite kernel sampling (and equating  $m_j/\rho_j$  with a volume element  $dV$ ), we see that  $E_{0,i}$  is a discrete approximation to the volume integral:

$$E_{0,i} \simeq 2 \int_V dV \nabla^x W = \mathbf{0} \quad (14)$$

which is vector zero because  $\nabla^x W$  is antisymmetric.

Although  $E_{0,i}$  should be approximately zero, it is problematic because it appears in equation (12) at order  $h_i^{-1}$ . Formal convergence then requires that  $E_{0,i}$  shrinks faster than  $h_i$ . This can be tricky to ensure and depends intimately upon the choice of kernel  $W$  employed. A popular choice is the cubic spline (CS) kernel:

$$W = \frac{8}{\pi h^3} \begin{cases} 1 - 6x^2 + 6x^3 & 0 < x \leq \frac{1}{2}, \\ 2(1-x)^3 & \frac{1}{2} < x \leq 1, \\ 0 & \text{otherwise,} \end{cases} \quad (15)$$

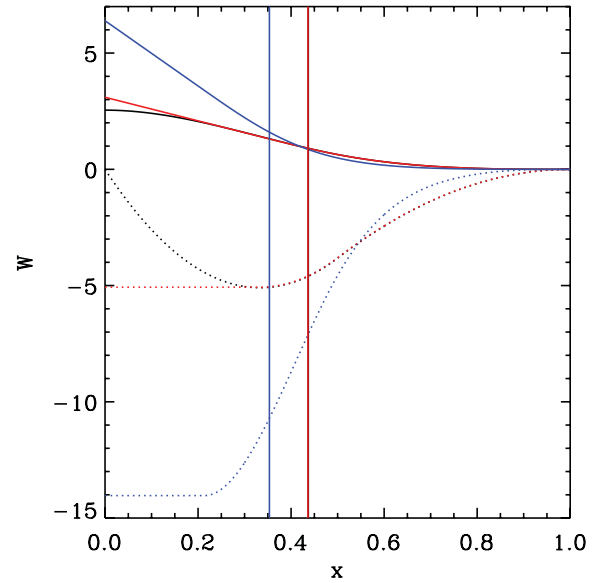
where  $x = r/h$  and  $h$  defines the kernel edge *not* its resolving power.<sup>6</sup> (The two are not the same as can be readily understood by considering a Gaussian kernel. This has an infinite edge, but a resolving power given by approximately its scale length.)

Now, it is tempting to increase the kernel sampling simply by stretching  $h$  for the CS kernel. However, this is a bad idea for two reasons. First, it introduces bias into the density estimate, spoiling convergence. Secondly, the CS kernel is not stable to large neighbour number. As  $h$  is increased, the particles clump on the kernel scale and the sampling is not significantly improved. For these reasons, in RHA10 we proposed a new class of kernels that can be used to achieve convergence. The lowest order of these was the CT kernel:

$$W = \frac{N}{h^3} \begin{cases} (-12\alpha + 18\alpha^2)x + \beta & 0 < x \leq \alpha, \\ 1 - 6x^2 + 6x^3 & \alpha < x \leq \frac{1}{2}, \\ 2(1-x)^3 & \frac{1}{2} < x \leq 1, \\ 0 & \text{otherwise,} \end{cases} \quad (16)$$

<sup>5</sup> Note that this assumes that the pressures are smooth and therefore differentiable. In classic SPH, this is not guaranteed. However, in SPHS, we add dissipation terms to ensure that this is the case. We discuss these in detail in Section 4.

<sup>6</sup> In many SPH papers,  $h$  is defined as a kernel scale length rather than the compact support size. In this paper,  $h$  always refers to the edge of the kernel where  $W(r, h) = 0$ .



**Figure 1.** The CS (black), CT (red) and HOCT4 (blue) kernels and their first derivatives (dotted lines) as a function of dimensionless length  $x = r/h$ , where  $h$  is the compact support size of the kernel. The vertical lines mark the half-mass radii for each kernel. For the CS and CT kernels the half-mass radii overlap on this plot.

where  $\beta = 1 + 6\alpha^2 - 12\alpha^3$ ,  $N = 8/[\pi(6.4\alpha^5 - 16\alpha^6 + 1)]$  and  $\alpha = 1/3$ . This has spatial resolution similar to the CS kernel but is stable to larger neighbour numbers. The next highest order was the HOCT4 kernel:

$$W = \frac{N}{h^3} \begin{cases} Px + Q & 0 < x \leq \kappa, \\ (1-x)^4 + (\alpha-x)^4 + (\beta-x)^4 & \kappa < x \leq \beta, \\ (1-x)^4 + (\alpha-x)^4 & \beta < x \leq \alpha, \\ (1-x)^4 & x \leq 1, \\ 0 & \text{otherwise,} \end{cases} \quad (17)$$

with  $N = 6.515$ ,  $P = -2.15$ ,  $Q = 0.981$ ,  $\alpha = 0.75$ ,  $\beta = 0.5$  and  $\kappa = 0.214$ . The CS, CT and HOCT4 kernels and their first derivatives are shown in Fig. 1.

We demonstrated in RHA10 that the HOCT4 kernel is stable for 442 neighbours on a lattice, while having a similar spatial resolution to the CT or CS kernel with 128 neighbours. This can be partially understood just from the half-mass radii of these two kernels (Fig. 1). If we assume that the resolution scale is the half-mass radius, then 442 neighbours for the HOCT4 kernel is equivalent to  $\sim 240$  neighbours for the CS kernel. However, in RHA10, we assess the spatial resolution more carefully by comparing instead the ability for these two kernels to correctly resolve the sound speed of linear waves. This is what leads to the conclusion that the HOCT4 kernel with 442 neighbours has similar resolving power to the CS with 128. Compared to ‘classic’ SPH with 42 neighbours, the HOCT4 kernel with 442 neighbours has, therefore, a poorer spatial resolution of only a factor of  $\sim (128/42)^{1/3} \sim 1.5$ . We will use the HOCT4 kernel with 442 neighbours for our default SPHS scheme. The CT kernel with 128 neighbours will be used only for convergence testing.

As pointed out by Price (2012), the CT and HOCT4 kernels have slightly larger density error than the more standard CS kernel. However, the effect is small (see Table 1). For glass particle distributions with 128 neighbours, the CT kernel gives a density error  $\sim 2$  per cent larger than the CS kernel, while the HOCT4 kernel with 442

**Table 1.** Density errors for a selection of SPH kernels applied to a constant density box. The columns give the following: kernel type; neighbour number; lattice configuration (glass or simple cubic) and the median/5 per cent/95 per cent recovered density to two significant figures (the true density is  $\rho_{\text{true}} = 1.00$ ).

Kernel	$N_n$	Lattice	$\rho$ (5 per cent)	$\rho$ (median)	$\rho$ (95 per cent)
CS	32	Simple	1.00	1.00	1.00
CS	128	Simple	1.00	1.00	1.00
CS	32	Glass	1.02	1.01	1.01
CS	128	Glass	1.01	1.00	0.996
CT	32	Simple	1.07	1.07	1.07
CT	128	Simple	1.02	1.02	1.02
CT	32	Glass	1.08	1.07	1.06
CT	128	Glass	1.02	1.02	1.01
HOCT4	442	Simple	1.01	1.01	1.01
HOCT4	442	Glass	1.02	1.01	1.00

neighbours is only  $\sim 1$  per cent worse.<sup>7</sup> (Note that the error can be very large if too few neighbours are used: higher order kernels require more neighbours to be adequately sampled.)

In summary, formal convergence in SPH is somewhat subtle; it requires several important criteria to be satisfied:

- (i) increased particle number,  $N$ ;
- (ii) increased neighbour number,  $N_n$  to ensure that  $\mathbf{E}_{0,i}$  shrinks faster than  $h_i$ ;
- (iii) a higher order kernel to maintain spatial resolution and
- (iv) a kernel that is stable to clumping/banding for the above choice of  $N_n$ .

The CT kernel with 128 neighbours and the HOCT4 kernel with 442 neighbours thus provide a convergent kernel pair that satisfy the above criteria. We will demonstrate this in Section 7.3. (Note that the above is simply what is required *formally*. It may be that for a given numerical problem  $\mathbf{E}_{0,i}$  shrinks faster than  $h_i$  without any need to raise the neighbour number. This cannot be guaranteed in general, however.)

The above convergence criteria – constructed simply to ensure that  $\mathbf{E}_{0,i}$  shrinks faster than  $h_i$  – seem rather laborious. Given that we know a priori what  $\mathbf{E}_{0,i}$  is for each particle, it is tempting to simply factor it out of the momentum equation as follows:

$$\frac{d\mathbf{v}_i}{dt} = - \underbrace{\sum_j^N \frac{m_j}{\rho_i \rho_j} [P_j - P_i] \nabla_i \bar{W}_{ij}}_{\text{Morris equation}} - \frac{P_i}{h_i \rho_i} \mathbf{E}_{0,i}. \quad (18)$$

The left-hand term in equation (18) – that we shall call the *Morris equation* (Morris 1995) – is now a higher order momentum equation. It gives zero force for constant pressure by construction, unlike equation (7). And it should give much simpler convergence – no longer requiring increased neighbour number, or careful kernel choice (indeed we will demonstrate this in Section 7.3). Such higher order momentum equations have been discussed several times in the literature before (e.g. Morris 1995; Oger et al. 2007). Recently, Abel (2011) have proposed an interesting variant similar to the Morris equation but with  $\rho_i \rightarrow \rho_j$ . This produces a manifestly smoother

<sup>7</sup> It is not clear, given these results, why Price (2012) argue that the density error is prohibitive for the CT kernel. Most likely it is because the error is very large when only 32 neighbours are used. For larger neighbour numbers, however, the performance of the CT and HOCT4 kernels is acceptable.

force that has improved stability properties as compared to the Morris equation.

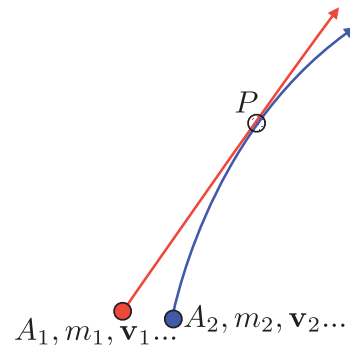
While higher order momentum equations are appealing, their advantages come at a price. As pointed out in RHA10, notice that the Morris equation is *symmetric* in  $i$  and  $j$  inside the sum. This means that momentum is no longer conserved between particle pairs. This lack of manifest momentum conservation becomes a problem in strong shocks (Morris 1996a). We will discuss subtracted- $\mathbf{E}_{0,i}$  momentum equations and their potential for creating higher order SPH-like methods in a separate paper (Hayfield & Read, in preparation).

#### 4 CONVERGENT FLOW AND DISSIPATION

The second problem with SPH is dealing with flow convergence – a problem common to any Lagrangian scheme. SPH can be thought of as both a Monte Carlo method and a method of characteristics. It is a Monte Carlo method because a finite number of discrete particles are used to initially sample the fluid. However, from this moment onwards it a method of characteristics: the particles move along streamlines in the flow. The problem is that the particles represent large unresolved patches of the fluid. Unlike real infinitesimal points in a fluid flow, SPH particles can approach one another, as shown in Fig. 2. This leads to multivalued fluid quantities at the crossing point: multivalued momentum, entropy, mass and any other fluid quantity that is advected with the particles. The only quantity that is not multivalued is the density since this is calculated by smoothing over a particles' nearest neighbours (see equation 6).

The problem of particles approaching one another was realized very early on in the development of SPH, and led to the introduction of artificial viscosity. This acts to make the momentum between particles single valued as they approach one another, while maintaining energy and momentum conservation. However, less appreciated in the literature is the need for similar dissipation terms in all other advected fluid quantities. This was recently highlighted by Price (2008). For example, if two particles approach one another with very different entropy, their pressures will become multivalued. This leads to a spurious repulsive force between the particles which inhibits mixing. In RHA10, we referred to this as the LMI. However, it can be thought of as a more general problem of multivalued fluid quantities arising when particles approach one another.

There are two possible solutions to deal with multivalued pressures in SPH. In RHA10, we used an idea from Ritchie & Thomas



**Figure 2.** A schematic representation of two SPH particles approaching one another in a convergent flow. The two particles carry discretely different advected quantities with them: entropy,  $A_{1,2}$ , mass,  $m_{1,2}$ , velocity,  $\mathbf{v}_{1,2}$ , etc. Apart from their densities that are manifestly smooth (cf. equation 6), all other fluid quantities become multivalued at point  $P$ . Thus, we must detect when this situation is going to occur and add dissipation terms in *all* fluid quantities to ensure that they remain single valued throughout the flow.



(2001) to manifestly smooth the pressures by using a modified ‘RT’ density estimator:

$$\rho_i = \sum_j^N \left( \frac{A_j}{A_i} \right)^{1/\gamma} m_j \bar{W}_{ij} \quad (19)$$

which gives (via equation 8)

$$P_i = \left[ \sum_j^N A_j^{1/\gamma} m_j \bar{W}_{ij} \right]^\gamma. \quad (20)$$

The above density estimator, combined with equations (6) and (7), defines the OSPH method derived in RHA10.

The ‘RT’ density estimator has the nice feature that it avoids multivalued pressures *by construction*. However, there is an associated cost. Consider the situation of large entropy contrasts on the kernel scale. Particles with  $A_i \gg A_j$  will contribute essentially zero weight, reducing the effective kernel sampling. To maintain a constant  $|E_0|$  error, we must then scale the neighbour number proportional to the entropy contrast on the kernel scale. This becomes prohibitively expensive for astrophysically important applications like strong blast waves. Here, OSPH gives significantly poorer performance than SPH for the same numerical cost. We demonstrate this in Appendix A.

For the above reasons, in this paper we take an approach more similar to Price (2008), but with a key difference. Price (2008) presented dissipation switches designed to detect (and correct) multivalued pressures. However, once pressures are multivalued it is already too late. As demonstrated recently by Valcke et al. (2010), once pressure blips form at flow boundaries, they cause pressure waves to propagate throughout the fluid. These damp the growth of surface instabilities and cause errors to propagate throughout the flow. To avoid this problem, we must detect when particles will approach one another *in advance*. We can then act to ensure that all fluid quantities (not just the pressure) will be single valued by the time the particles reach one another. This is the strategy we adopt here.

To detect when particles will cross, we require an accurate flow convergence detector. We take an approach similar to Cullen & Dehnen (2010). Cullen & Dehnen (2010) came up with the novel idea of using the time derivative of the velocity divergence to detect flow convergence in advance. They then switch on artificial viscosity to prevent particle interpenetration. We use a similar idea, but consider instead the *spatial derivative* of the velocity divergence. As we will show, this has the advantage that we obtain an excellent estimate of the flow divergence and curl for free.

Cullen & Dehnen (2010) focus only on the artificial viscosity. Here, we use the same switch not just for the artificial viscosity, but for *all* artificial dissipation terms. (Recall that we require one of these for each advected fluid quantity.)

We have some freedom in how to construct the flow convergence detector and the dissipation terms. However, both must satisfy a number of constraints in order for the scheme to produce converged results with increasing resolution:

- (i) the switch must detect flow convergence *before it occurs*;
- (ii) the switch must be sufficiently robust (i.e. high order) as to not trigger randomly due to particle noise;
- (iii) the dissipation terms must respect conservation laws;
- (iv) the dissipation terms must shrink on a given physical scale with increasing resolution and
- (v) the dissipation terms must not generate spurious pressure waves that propagate through the fluid.

These criteria guide our choices for the switch and the artificial dissipation terms that we describe in the following two subsections. The last point, in particular, is important. It is no good if our dissipation terms introduce more problems than they solve. They should act to make fluid quantities single valued wherever particles approach one another. But they should do this in a manner that respects conservation laws, is convergent, and does not lead to problems elsewhere in the flow.

#### 4.1 A higher order convergence detector

We first describe our higher order flow convergence detector. Local flow convergence occurs wherever the velocity divergence is negative. This suggests that we should switch on dissipation terms if  $\nabla \cdot \mathbf{v}_i < 0$  for a given particle. However, if we set the magnitude of the dissipation also using  $\nabla \cdot \mathbf{v}_i$ , then the dissipation will only switch on once the flow is converging, not before. To detect flow convergence in advance, we use instead the *spatial derivative* of  $\nabla \cdot \mathbf{v}_i$  for the magnitude of our dissipation parameter  $\alpha_{loc,i}$ . This leads to the following dimensionless dissipation switch:

$$\alpha_{loc,i} = \begin{cases} \frac{h_i^2 |\nabla(\nabla \cdot \mathbf{v}_i)|}{h_i^2 |\nabla(\nabla \cdot \mathbf{v}_i)| + h_i |\nabla \cdot \mathbf{v}_i| + n_s c_s} \alpha_{max} & \nabla \cdot \mathbf{v}_i < 0, \\ 0 & \text{otherwise,} \end{cases} \quad (21)$$

where  $\alpha_{loc,i}$  describes the amount of dissipation for a given particle in the range  $[0, \alpha_{max} = 1]$  and  $n_s = 0.05$  is a ‘noise’ parameter that determines the magnitude of velocity fluctuations that trigger the switch. Equation (21) turns on dissipation if  $\nabla \cdot \mathbf{v}_i < 0$  (convergent flow) and if the magnitude of the spatial derivative of  $\nabla \cdot \mathbf{v}_i$  is large as compared to the local divergence (i.e. if the flow is going to converge).

In principle, the maximum dissipation parameter  $\alpha_{max}$  can be different for each fluid quantity. Our default in this paper is to use  $\alpha_{max} = 1$  for all fluid variables. We investigate the sensitivity of SPHS to  $\alpha_{max}$  in Appendix C.

As in Cullen & Dehnen (2010), we set the local dissipation to the above value instantaneously if  $\alpha_i < \alpha_{loc,i}$ ,

$$\alpha_i = \alpha_{loc,i}, \quad \alpha_i < \alpha_{loc,i}, \quad (22)$$

otherwise  $\alpha_i$  smoothly decays back to zero:

$$\begin{aligned} \dot{\alpha}_i &= (\alpha_{loc,i} - \alpha_i) / \tau_i, & \alpha_{min} < \alpha_{loc,i} < \alpha_i, \\ \dot{\alpha}_i &= (\alpha_{min} - \alpha_i) / \tau_i, & \alpha_{min} > \alpha_{loc,i}, \end{aligned} \quad (23)$$

where  $\tau_i = h_i / v_{max,i}$  is the time-scale for the decay, and  $v_{max,i}$  is the maximum signal velocity (Springel 2005):

$$v_{max,i} = \max_j [v_{sig,ij}], \quad (24)$$

with

$$v_{sig,ij} = c_i + c_j - 3w_{ij}, \quad (25)$$

where  $w_{ij} = (\mathbf{v}_{ij} \cdot \mathbf{r}_{ij}) / |\mathbf{r}_{ij}|$ , and  $c_i$  is the local sound speed at particle  $i$ .

The parameter  $\alpha_{min} = 0.2$  ensures that the dissipation parameter decays all the way back to zero once particles are no longer converging.

#### 4.2 A higher order gradient estimator

Our dissipation switch (equation 21) requires a good estimate of both the first and second derivatives of the velocity field. A noisy estimator will cause the limiter to trigger unnecessarily, leading

to an overly diffusive method.<sup>8</sup> To achieve good quality gradients, we fit a second order polynomial to each of the fluid variables as in Maron & Howes (2003). The first and second derivatives then follow from the coefficients of the polynomial fit. The full 3D algorithm is given in Appendix D. Here we present a 1D version to illustrate the idea.

We assume that a fluid variable,  $q_i$ , can be locally represented by a smooth second-order polynomial:

$$q_i = a_{0,i} + a_{1,i}x_{ij} + a_{2,i}x_{ij}^2 + O(h^3), \quad (26)$$

where  $x_{ij} = r_{ij}/h_i$ .

To determine the coefficients  $a_{n,i}$ , we then consider the matrix equation  $\mathbf{M}\mathbf{a} = \mathbf{q}$ :

$$\begin{aligned} & \left[ \sum_j^N m_j W_{ij} \begin{pmatrix} 1 & x_{ij} & x_{ij}^2 \\ x_{ij} & x_{ij}^2 & x_{ij}^3 \\ x_{ij}^2 & x_{ij}^3 & x_{ij}^4 \end{pmatrix} \right] \begin{pmatrix} a_{0,i} \\ a_{1,i} \\ a_{2,i} \end{pmatrix} \\ & = \sum_j^N m_j W_{ij} \begin{pmatrix} q_j \\ q_j x_{ij} \\ q_j x_{ij}^2 \end{pmatrix}. \end{aligned} \quad (27)$$

The matrix  $\mathbf{M}$  and the vector  $\mathbf{q}$  contain weighted moments that can be calculated in the usual way by summing over each particle's nearest neighbours. The vector  $\mathbf{a}$  is then calculated by solving for the inverse of  $\mathbf{M}$ . The particle gradients at the position of the particle ( $x_{ij} = 0$ ) then follow from  $q'_i(0) = a_{1,i}$  and  $q''_i(0) = 2a_{2,i}$ .

The above straightforwardly generalizes to 3D and to vector fluid variables. For scalar variables in 3D we must solve a  $10 \times 10$  matrix inverse to obtain a 10 coefficient fit (see Appendix D):

$$\begin{aligned} q_{ij} = & a_{0,i} + a_{1,i}x_{ij} + a_{2,i}y_{ij} + a_{3,i}z_{ij} + a_{4,i}x_{ij}^2 + a_{5,i}y_{ij}^2 \\ & + a_{6,i}z_{ij}^2 + a_{7,i}x_{ij}y_{ij} + a_{8,i}x_{ij}z_{ij} + a_{9,i}y_{ij}z_{ij} \\ & + O(h^3), \end{aligned} \quad (28)$$

where  $\mathbf{x}_{ij} = \mathbf{r}_{ij}/h_i = [x_{ij}, y_{ij}, z_{ij}]$ .

Note that Maron & Howes (2003) use these higher order gradients to actually move the fluid. This makes the method non-conservative, leading to problems in strong shocks. In SPHS, we use these gradients instead to conservatively maintain fluid smoothness.

Our dissipation switch manifestly satisfies our criteria (i) and (ii) outlined above. It detects flow convergence in advance, and it is accurate since it is based on a second order accurate expansion of the velocity field.

Note that a second-order polynomial is the lowest order that we could fit in order to obtain a second derivative. In principle, we could fit a third- or fourth-order polynomial thus further increasing the accuracy of the switch. However, this comes at quite significant cost. At third order, the size of the moment matrix increases from  $10 \times 10$  to  $20 \times 20$  and requires an additional 40 sums over the particles to be calculated and stored. Secondly, for the higher order moments to actually help, the neighbour number should be increased. Otherwise noise in the third moments could make the higher order gradient estimator poorer than the second-order estimate. For these reasons, we stick to the second-order scheme in this paper.

<sup>8</sup> Indeed, Rosswog (2010) recently advanced the idea of using higher order gradients for their dissipation switch. They used a first-order accurate gradient of the pressure, whereas we use the gradient of the velocity divergence (which is a second derivative of the velocity field).

### 4.3 The dissipation terms

As discussed at the start of this section, we require a dissipation term for each advected fluid quantity. These will then be switched on (using the switch described in Section 4.1) if the flow is converging. In this subsection, we describe the conservative dissipation terms that we use for the momentum (artificial viscosity; Section 4.3.1), entropy (artificial thermal conductivity; Section 4.3.2) and mass (artificial mass dissipation;<sup>9</sup> Section 4.3.3). These dissipation terms (described in equations 29 to 40) are all added together in our SPHS scheme.

#### 4.3.1 Artificial viscosity

We start with the familiar artificial viscosity. Here, we use the form as in Monaghan (1997) and Springel (2005):

$$\dot{\mathbf{v}}_{\text{diss},i} = - \sum_j^N m_j \Pi_{ij} \nabla_i \bar{W}_{ij}, \quad (29)$$

where

$$\Pi_{ij} = \begin{cases} -\frac{\bar{\alpha}_{ij}}{2} \frac{v_{\text{sig},ij} w_{ij}}{\bar{\rho}_{ij}} & \text{if } \mathbf{v}_{ij} \cdot \mathbf{r}_{ij} < 0, \\ 0 & \text{otherwise,} \end{cases} \quad (30)$$

where  $\bar{\alpha}_{ij} = (1/2) [\alpha_i + \alpha_j]$ , and  $v_{\text{sig},ij}$  and  $w_{ij}$  are defined by equation (25). This must then generate entropy to ensure energy conservation:

$$\dot{A}_{\text{diss},i} = -\frac{1}{2} \frac{\gamma - 1}{\rho_i^{\gamma-1}} \sum_j^N m_j \bar{\alpha}_{ij} \Pi_{ij} \mathbf{v}_{ij} \cdot \nabla_i \bar{W}_{ij}. \quad (31)$$

In addition, we use a Balsara-like switch to limit viscosity in shear flows (Balsara 1989). As in Cullen & Dehnen (2010), we apply this to our viscosity parameter  $\alpha_i$ , rather than directly to equation (31). This is mathematically identical, but means that  $\alpha_i$  represents the true viscosity of the flow. Thus, we multiply  $\alpha_i$  by a suppression function given by

$$f_{\text{Balsara},i} = \frac{|\nabla \cdot \mathbf{v}|_i}{|\nabla \cdot \mathbf{v}|_i + |\nabla \wedge \mathbf{v}|_i + 0.0001c_i/h_i}, \quad (32)$$

where  $c_i$  is the sound speed for particle  $i$ . Equation (32) is identical to the usual Balsara switch, except that we use the higher order gradients derived in Appendix D to derive the divergence and curl of the velocity field.

Equations (29) and (31) satisfy our dissipation criteria (ii)–(v) outlined at the start of Section 4. They respect energy and momentum conservation by construction; they act only on the kernel scale (and thus the viscosity will reduce at a given physical scale as the resolution is increased); and they introduce a numerical error only locally.

#### 4.3.2 Entropy dissipation

For our dissipation in the entropy function variable  $A_i$ , we choose a form that explicitly conserves energy, similar to that proposed in Price (2008):

$$\dot{A}_{\text{diss},i} = \sum_j^N \frac{m_j}{\bar{\rho}_{ij}} \bar{\alpha}_{ij} v_{\text{sig},ij}^p L_{ij} \left[ A_i - A_j \left( \frac{\rho_j}{\rho_i} \right)^{\gamma-1} \right] K_{ij}, \quad (33)$$

<sup>9</sup> This is only required if there are multimass particles.

where  $\bar{\rho}_{ij} = [\rho_i + \rho_j]/2$  is the symmetrized density,  $K_{ij} = \hat{r}_{ij} \cdot \nabla_i W_{ij}$  is a symmetric smoothing kernel,  $L_{ij}$  is a pressure limiter (of which more in a moment) and  $v_{\text{sig},ij}^p$  is similar to the signal velocity, but defined to be positive definite:

$$v_{\text{sig},ij}^p = \begin{cases} c_i + c_j - 3w_{ij} & \text{if } 3w_{ij} < (c_i + c_j), \\ 0 & \text{otherwise.} \end{cases} \quad (34)$$

This modified signal velocity is chosen to give more dissipation to approaching particle pairs than receding particle pairs. However, unlike the viscosity where the dissipation is fully suppressed for receding pairs (cf. equation 30), we find that receding pairs still require some small entropy dissipation. This is because, while neighbouring particles can have discretely different velocities without serious repercussion (so long as they are not approaching one another), discretely different entropies inside the kernel will drive spurious pressure waves that affect the numerical solution everywhere.

In fact, the above explains why adding some small entropy dissipation is preferable to doing nothing at all. The right amount of entropy dissipation will ensure smooth pressures and keep errors local. But the key is getting the ‘right amount’. If we are not careful, our dissipation terms can actually drive pressure waves and do more harm than good. To avoid this, we introduce a pressure limiter:

$$L_{ij} = \frac{|P_i - P_j|}{P_i + P_j}. \quad (35)$$

Note that, unlike the dissipation prescription presented in Price (2008), equation (33) poses no problem for simulations involving gravity. In hydrostatic equilibrium the entropy dissipation will vanish since the flow is non-converging and  $\bar{\alpha}_{ij} = 0$ .

Equation (33) satisfies our dissipation criteria (iii)–(v) outlined at the start of Section 4. It respects energy conservation by construction, acts only on the kernel scale and – through the pressure limiter – does not propagate errors non-locally.

### 4.3.3 Mass dissipation (for multimass applications)

Multimass SPH particles are very useful since they allow interesting regions of the flow to be simulated at significantly higher resolution (e.g. Monaghan & Varnas 1988; Meglicki, Wickramasinghe & Bicknell 1993). However, classic SPH runs into difficulties once particle masses are allowed to vary (see e.g. Ott & Schnetter 2003). The problems occur because, like the entropy, the particle masses are advected along with the particles. When particles approach one another, the masses become multivalued, driving a pressure wave at the mass interface. The problem is less severe than for the entropies because the masses are smoothed over in the equation of state (cf. equations 6 and 8). Nonetheless, large density contrasts realized with multimass particles are problematic.

Some approaches to multimass SPH have been proposed in the literature. Ott & Schnetter (2003) suggest adapting the density estimate to ensure smooth pressures by construction – an approach very similar to the multiphase SPH proposed by Ritchie & Thomas (2001). Kitsionas & Whitworth (2002) suggest increasing the neighbour number at the mass interface. This acts to smooth the pressure blip. However, for reasonable neighbour number increases there can still be significant noise/error introduced at the boundary.

A key advantage of our approach is that we can treat any advected fluid quantity in the same manner as the entropy, above. This includes the particle masses, which allows us to consider a multimass SPH scheme that does not require raising the neighbour number at boundaries, or introducing a new density estimator. Treating the

mass similarly to the entropy, above, we introduce a conservative pairwise mass dissipation:

$$\dot{m}_{\text{diss},i} = \sum_j \frac{\bar{m}_{ij}}{\bar{\rho}_{ij}} \bar{\alpha}_{ij} v_{\text{sig},ij}^p L_{ij} [m_i - m_j] K_{ij}, \quad (36)$$

where  $\bar{m}_{ij} = [m_i + m_j]/2$  is the symmetrized mass. Note that this symmetrized mass appears *only* in equation (36), and not in the other dissipation terms. This difference follows from the fact that equation (36) must respect mass conservation, while equations (33) and (31) – that describe the evolution of the *specific* entropy – must respect energy conservation (see Appendix E for further details).

As for the artificial viscosity, we must then add correction terms to ensure momentum and energy conservation. There is actually some freedom in how we choose to do this (see Appendix E). A simple approach is to ensure that each particle individually conserves its energy and momentum:

$$\frac{d(m_i \mathbf{v}_i)}{dt} = \dot{m}_i \mathbf{v}_i + m_i \dot{\mathbf{v}}_i = 0, \quad (37)$$

$$\frac{dE_i}{dt} = \dot{m}_i u_i + m_i \dot{u}_i + \frac{1}{2} \dot{m}_i \mathbf{v}_i \cdot \mathbf{v}_i + m_i \mathbf{v}_i \cdot \dot{\mathbf{v}}_i, \quad (38)$$

where we recall that  $u_i$  is the specific internal energy. Substituting equation (37) into equation (38) then gives the correction terms for each particle:

$$\dot{\mathbf{v}}_{\text{diss},i} = -\frac{\dot{m}_{\text{diss},i}}{m_i} \mathbf{v}_i, \quad (39)$$

$$\dot{A}_{\text{diss},i} = \frac{1}{2} \frac{\gamma - 1}{\rho_i^{\gamma-1}} \frac{\dot{m}_{\text{diss},i}}{m_i} [\mathbf{v}_i \cdot \mathbf{v}_i] - \frac{\dot{m}_{\text{diss},i}}{m_i} A_i. \quad (40)$$

We will use the above correction terms throughout this paper. (We derive a general class of correction terms in Appendix E; these may lead to even better results in some situations, but we leave this as an investigation for future work.) It is clear that equations (36), (39) and (40) satisfy our criteria (iii)–(v) outlined at the start of Section 4. For equal mass particle applications,  $\dot{m}_{\text{diss},i} = 0$  by construction and equations (36), (39) and (40) have no effect.

A final concern is that adding mass dissipation will affect our solution of the continuity equation. Taking the time derivative of equation (6), we have that

$$\dot{\rho}_i = \frac{d}{dt} \sum_j m_j W_{ij} = \sum_j \dot{m}_j W_{ij} + \sum_j m_j \mathbf{v}_{ij} \cdot \nabla W_{ij}. \quad (41)$$

The right-hand term is the familiar SPH continuity equation; the left-hand term is a correction factor that accounts for mass dissipation. Thus, by using the familiar SPH density sum (equation 6), we *automatically* include the mass dissipation correction to the continuity equation. However, we may still worry whether equation (41) tends towards equation (1) in the limit of infinite resolution. Substituting for  $\dot{m}_j = \dot{m}_{\text{diss},j}$  in equation (41), we have

$$\sum_j \dot{m}_j W_{ij} = \sum_{j,k} Q_{jk} (m_j - m_k) K_{jk} W_{ij} \simeq 0, \quad (42)$$

where  $Q_{ij} = Q_{ji} = \bar{m}_{ij} / \bar{\rho}_{ij} \bar{\alpha}_{ij} v_{\text{sig},ij}^p L_{ij}$ , and the equation is very nearly vanishing since the sum is almost perfectly antisymmetric in the indices  $j, k$  (the antisymmetry is broken by  $W_{ij}$ ). In the continuum limit, the above term is exactly zero and so equation (41) does indeed tend towards equation (1) with increasing resolution.

## 5 TIME-STEPPING

For our time-step control, we use individual particle time-steps ordered on a hierarchy of rungs in powers of two, as in Springel (2005). Particles are placed on rungs using a Courant-like condition:

$$\Delta t_i = C \frac{h_i}{v_{\max,i}}, \quad (43)$$

where  $C = 0.2$  is the Courant factor. We use this same fixed Courant factor for all tests presented in this paper.

In addition to the standard time-step criteria above, we introduce a constraint similar to that in Saitoh & Makino (2009) to ensure that neighbouring particles do not differ in their time-steps by more than a factor of 4. The algorithm we use is based on code kindly provided by Muldrew & Pearce (private communication).

## 6 IMPLEMENTATION

We implemented our method in the `GADGET-2` code (Springel 2005). `GADGET-2` is a massively parallel `TREE-SPH` code originally designed to model galaxy collisions, but now adapted for cosmological, hydrodynamic, magnetohydrodynamic and many other applications. `SPHS` acts as a new hydro module within `GADGET-2`, replacing the standard `SPH` parts of the code. We refer the reader to the original `GADGET-2` paper for details of the gravity solver (Springel 2005).

The `SPHS` hydro module like `GADGET-2` requires two loops over the particles. In the first loop, the densities are calculated (iterating to ensure equation 11 is satisfied). At the same time, we calculate the polynomial fluid gradients (for the dissipation switch; Section 4). In the second loop, the hydrodynamic forces are evaluated along with the dissipation terms. Some speed comparisons between our current implementation of `SPHS` and classic `GADGET-2` `SPH` are given in Section 7.8.

## 7 CODE TESTS

In this section, we present a suite of code tests designed to challenge the `SPHS` method. In Section 7.2, we use the Sod shock tube test to examine shocks in `SPHS` both with and without multimass particles, and to assess the rate of convergence in `SPHS`. In Section 7.3, we use the ‘Gresho’ vortex test to examine convergence in shear flows in `SPHS`, and the role of numerical viscosity. In Section 7.4, we use a strong Sedov–Taylor blast wave test to see how well `SPHS` performs in the presence of extreme entropy contrasts. In Section 7.5, we use a Kelvin–Helmholtz (KH) instability test with density contrast 1:8 – both with and without multimass particles – to examine mixing in `SPHS`. Finally, in Section 7.6, we use the ‘blob’ test – a 1:10 density ratio gas sphere in a wind tunnel to assess how `SPHS` performs in more complex flow situations where shocks and mixing combine.

### 7.1 Simulation labelling convention

In the following subsections, we run a broad range of simulations both in our new hydrodynamics code `SPHS`, and in ‘classic’ `SPH` (the version of `SPH` that is in the public release version of the `GADGET-2` code, and that is described in Springel 2005). To avoid confusion, we use the following naming convention for these simulations:

SPHX-KKNNx,

where X refers to the flavour of `SPH`: ‘classic’ `SPH` (`SPH`), or our new code (`SPHS`); KK refers to the kernel used (`CS` – cubic spline;

equation 15), (`CT` – core triangle; equation 16), (`HCT` – high order core triangle; equation 17); NN refers to the neighbour number (42, 96, 128, 442) and x is reserved to describe special simulations: x = g means that the test was run using glass rather than lattice initial conditions; x = e means that the test was run using a higher order momentum equation (equation 18 with the  $E_0$  term subtracted) and x = m means that the test was run using multimass particles.

### 7.2 Sod shock tube

The Sod shock tube test is a 1D tube on the interval  $[-0.5, 0.5]$  with a discontinuous change in properties at  $x = 0$  designed to generate a shock. The left state is described by  $\rho_l = 1.0$ ,  $P_l = 1.0$ ,  $v_l = 0$ , and the right state by  $\rho_r = 0.125$ ,  $P_r = 0.1$ ,  $v_r = 0$ , where  $\rho$ ,  $P$  and  $v$  are the density, pressure and velocity along the  $x$ -axis. We use an adiabatic equation of state with  $\gamma = 5/3$  and perform the test in 3D on the union of a  $32 \times 32 \times 400$  lattice on the left, with a  $16 \times 16 \times 200$  lattice on the right, giving a 1D resolution of  $N_{1D} = 600$  points. For the `SPHS` simulation, we set an initial dissipation parameter  $\alpha = 1$  over the initial pressure discontinuity ( $-0.05 < x < 0.05$ ) since this test starts with a shock. We use lattice ICs for this test aligned with the shock, identical to those presented in RHA10. In `SPH`, however, there is some freedom in how to lay down the particles. Cullen & Dehnen (2010), for example, use instead glass-like initial conditions. These are noisier than the simple cubic lattice we use here, but have no preferred direction. While the choice of initial condition can affect the results, it should not affect the difference in the results between `SPH` and `SPHS`. We will demonstrate this using glass and lattice ICs for the Gresho vortex test in Section 7.3.

The results at time  $t = 0.2$  are shown in Fig. 3 for `SPH` (red) and `SPHS` (black). The analytic solution is marked in blue. Notice that `SPHS` performs well on this test. In particular, the pressure blip at the shock, present in the `SPH` run (red), is almost completely gone. The `SPH` run, which used 96 neighbours, shows significantly more noise in the velocity distribution. This occurs due to symmetry breaking of the simple cubic lattice ICs and is reduced for glass-like initial conditions (Cullen & Dehnen 2010). It is also reduced by moving to higher order kernels that have larger neighbour number and are therefore correspondingly less noisy (e.g. Price 2012). This is why the noise is not present in the `SPHS` simulation.

In addition, we perform this same test using multimass particles in `SPHS`, where we sample the domain uniformly with a  $16 \times 16 \times 400$  lattice – the same resolution as the low-density phase in the single particle mass Sod test (Fig. 3, bottom panels). This is an extremely challenging test for `SPHS`. The initial conditions have a sharp jump in three fluid variables: entropy, pressure and mass. Nonetheless, the solution is in excellent agreement with the analytic result.

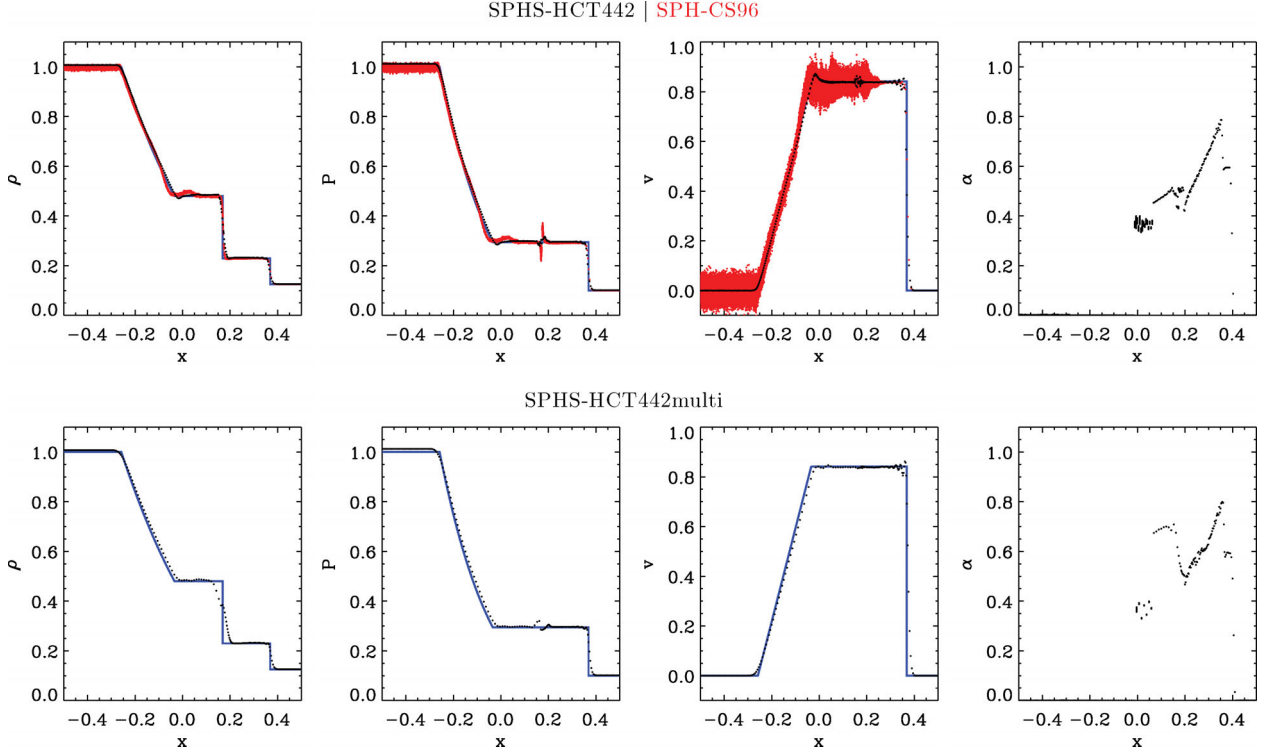
Finally, we perform a convergence study for the Sod shock tube test in Fig. 4. We follow Springel (2010b) and define our error measure as

$$L1(v_x) = \frac{1}{N_b} \sum_i^{N_b} |\bar{v}_{x,i} - \bar{v}_x(x_i)|, \quad (44)$$

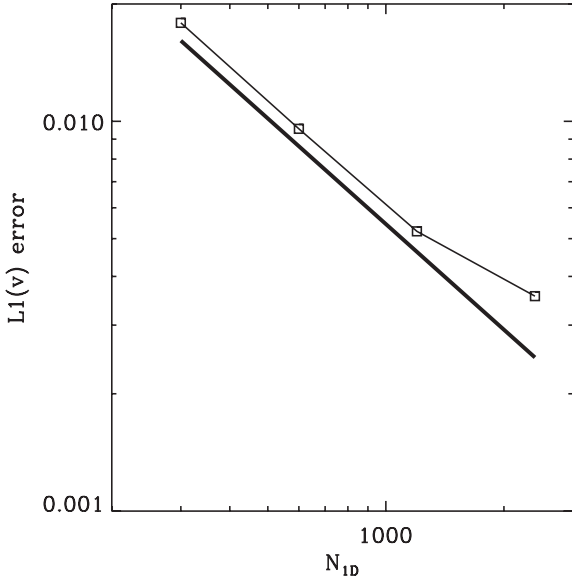
where  $N_b$  is the number of bins in  $x$ ,  $\bar{v}_{x,i}$  is the mean  $x$ -velocity in bin  $i$ , and  $\bar{v}_x(x_i)$  is the expected analytic mean velocity in bin  $i$ . We use a  $x$ -bin width of 0.01. This is small enough to capture the improvements with increasing resolution, but not so small as to oversample the lowest resolution run.

Springel (2010b) find that for shock tests in 2D, `SPH` performs worse than the optimal  $N_{1D}^{-1}$  scaling, giving something closer to  $N_{1D}^{-0.7}$ . Here we find that, by contrast, `SPHS` gives a near-optimal





**Figure 3.** 3D Sod shock tube test results at  $t = 0.2$  in SPHS (black) and SPH-CS96 (red). From left to right, the panels show the density, pressure, magnitude of  $v_x$  (the  $x$ -velocity component along the shock) and the dissipation switch  $\alpha$  (only relevant for the SPHS simulations). The blue line marks the analytic solution. Notice that the pressure blip at  $x \sim 0.2$  is almost fully removed in SPHS. The top panels show results for a single particle-mass simulation; the bottom for the same run with multimass particles in SPHS on a uniform particle grid.



**Figure 4.** Sod shock tube convergence test results. The  $x$ -axis gives the number of particles along the shock  $N_{\text{ID}}$  (Fig. 3, top, shows results for  $N_{\text{ID}} = 600$ ), the  $y$ -axis shows the binned velocity error in  $x$ -bins of width 0.01 and the thick black line shows a scaling of  $N_{\text{ID}}^{-0.9}$  (the best-possible scaling is  $N_{\text{ID}}^{-1}$ ).

scaling, going as very nearly  $N_{\text{ID}}^{-1}$ , except at the highest resolution (compare the black and thick black lines in Fig. 4). The slowing down of the convergence rate for the highest resolution simulation may be due to a fundamental convergence limit set by our neighbour

number (i.e. set by the magnitude of the  $E_0$  error; see Section 3). Unfortunately, we need to probe up to prohibitively high resolution to conclusively test this; we defer such tests to future work.

### 7.3 Gresho vortex test

We set up the Gresho vortex test similarly to Springel (2010b) (and see Gresho & Chan 1990). The test involves an  $N \times N \times N/8$  3D lattice of particles. A velocity and pressure field are applied to these to set up a stable vortex:

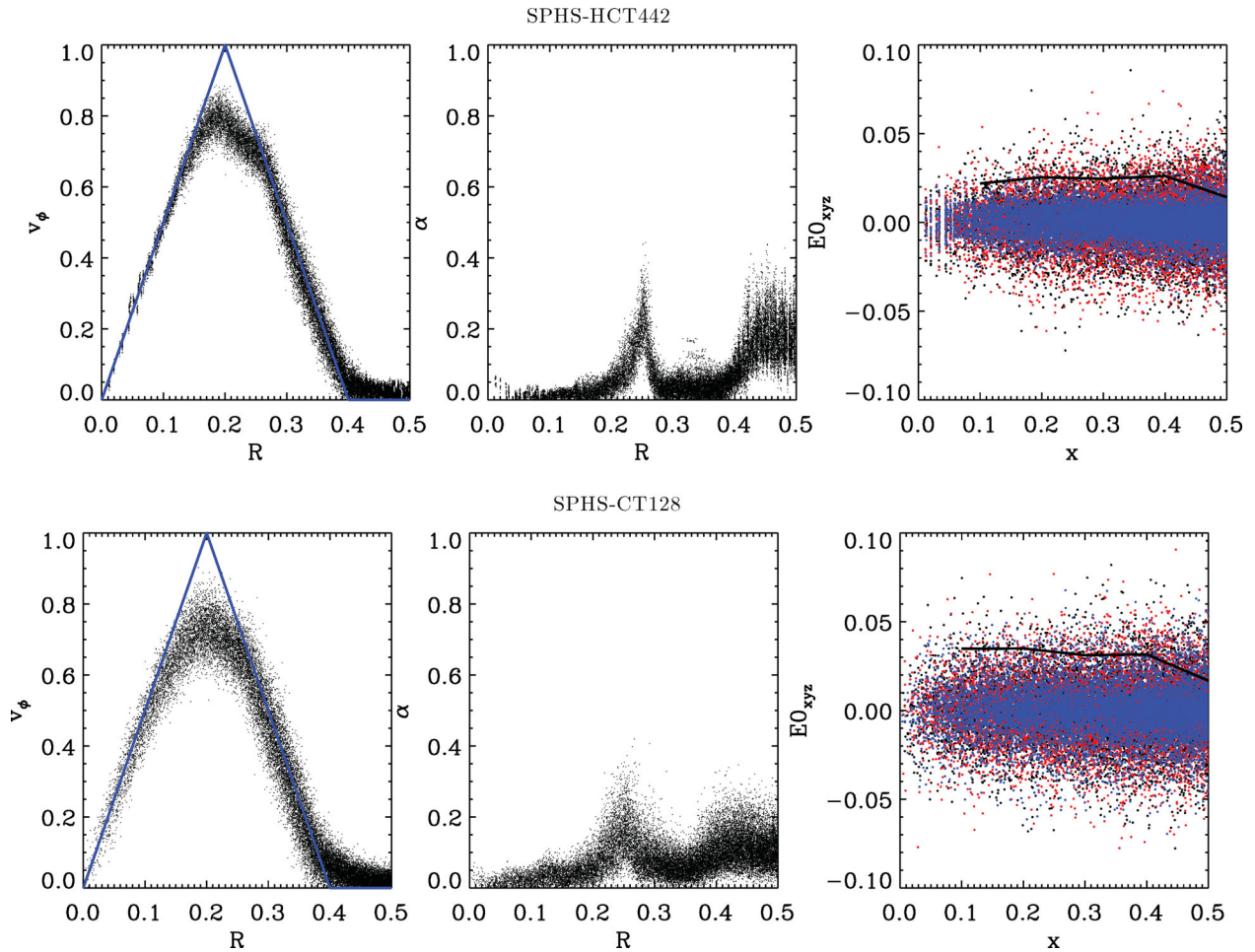
$$v_\phi(R) = \begin{cases} 5R & \text{for } 0 \leq R \leq 0.2, \\ 2 - 5R & \text{for } 0.2 \leq R \leq 0.4, \\ 0 & \text{for } R \geq 0.4, \end{cases} \quad (45)$$

$$P(R) = \begin{cases} 5 + \frac{25}{2}R^2 & \text{for } 0 \leq R \leq 0.2, \\ 9 + \frac{25}{2}R^2 & \text{for } 0.2 \leq R \leq 0.4, \\ -20R + 4 \ln(R/0.2) & \text{for } 0.2 \leq R \leq 0.4, \\ 3 + 4 \ln 2 & \text{for } R \geq 0.4, \end{cases} \quad (46)$$

where  $R = \sqrt{x^2 + y^2}$  and we set  $\rho = 1$  and  $\gamma = 5/3$ .

The above vortex should be stable over many rotations, but in practice will decay due to the numerical viscosity inherent in the scheme. As such, it is a useful test of the numerical viscosity generated in shear flows. Indeed, classic SPH performs poorly on this test converging very slowly to the wrong solution (Springel 2010b). Such rotating configurations are common in a wide class of astrophysical problems; it is important for numerical schemes to perform well on such tests.

The results for SPHS are given in Fig. 5. We show, from left to right, the rotational velocity profile of the vortex (black points) as

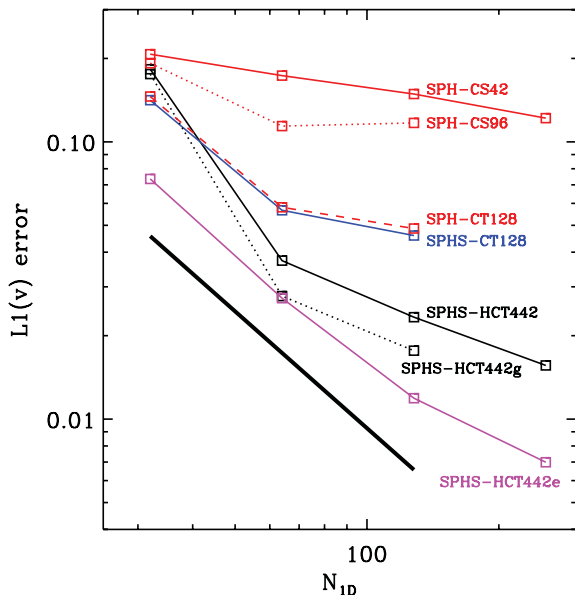


**Figure 5.** Gresho vortex test results. The top panels show results for the HOCT4 kernel with 442 neighbours; the bottom panels for the CT kernel with 128 neighbours. The left-hand panels show the  $v_\phi$  velocities of particles after a time  $t = 1$  for  $N = 64 \times 64 \times 8$  particles; the analytic solution is marked in blue. The middle panels show the dissipation parameter  $\alpha$  at the same time. The right-hand panels show the three  $E_0$  error components in black, red and blue. Overplotted is the mean binned error magnitude  $|E_0|$  (solid black line).

compared to the analytic solution (thick blue line); the dissipation parameter  $\alpha$  and the leading order error in the momentum equation  $E_0$ . We find significantly better performance in SPHS than was found by Springel (2010b) for SPH. The primary reason for this – surprisingly – is not the lower viscosity of the method. The average viscosity is lower in SPHS – in the range  $0.05 < \alpha < 0.3$  as compared to SPH that has constant  $\alpha = 1$  (see Fig. 5, middle panels). However, the real reason for the improvement is the improved *force accuracy*. In Fig. 5, the top three panels show the results for our default method (SPHS-HCT442), while the bottom three show results for SPHS using a lower neighbour number with the CT kernel (SPHS-CT128; see Section 7.1 for our simulation labelling convention). If anything, the viscosity is slightly *lower* for the SPHS-CT128 simulation, yet the results are worse, with increased noise and a bias in the rotational velocity for  $R \lesssim 0.2$ . The only difference between these two simulations is the neighbour number, and the associated  $E_0$  error. Indeed, in SPHS-HCT442, the  $E_0$  is lower than for the SPHS-CT128 simulation (see Fig. 5, right-hand panels).

In Fig. 6, we explore this further by presenting convergence tests for SPH and SPHS using varying neighbour number and kernel choice. We calculate the  $L1(v_\phi)$  error norm as in the Sod test (Section 7.2), using a bin size of  $\Delta R = 0.01$ . The thick black line on the plot marks the ideal scaling of  $N_{\text{ID}}^{-2}$  (ideal for a second order

method away from contact discontinuities). The red lines show results for classic SPH with 42 neighbours (solid line), 96 neighbours (dotted line) and 128 neighbours using the CT kernel (dashed line). Notice that SPH-CS42 converges very slowly with increasing resolution, with the error always larger than 10 per cent. Increasing the neighbour number to 96 neighbours helps at low resolution but gives diminishing returns with increasing resolution. This agrees with our results from RHA10, where we showed that in shear flows the  $E_0$  error improves only very slowly with increasing neighbour number for the CS kernel. This is because for neighbour number larger than  $\sim 40$ , the particles begin to clump preventing any significant improvement in the kernel sampling. By contrast, switching to the CT kernel with 128 neighbours – that is manifestly stable to particle clumping – gives a significant improvement in the convergence rate (dashed line). Now the error drops to  $\sim 5$  per cent for  $N_{\text{ID}} = 128$ . The solid blue line shows the result for SPHS using the CT kernel with 128 neighbours (SPHS-CT128). The results are only very slightly better than for classic SPH. This highlights that, for this test, it is the *force accuracy* that determines the rate of convergence, not the dissipation scheme. The solid black line shows the result for our default SPHS scheme: SPHS-HCT442. With 442 neighbours and a correspondingly higher order kernel, our method now converges on percent level accuracy for this test. The convergence



**Figure 6.** Gresho vortex convergence test results. The x-axis gives the number of particles along one side of the box  $N_{1D}$  (Fig. 5 shows results for  $N_{1D} = 64$ ); the y-axis shows the binned velocity error in  $R$ -bins of width 0.01. The different line colours show different SPH methods; the naming convention (marked) is as described in Section 7.1. Three simulations: SPHS-CS42 (‘classic’ SPH); SPHS-HCT442 (our default SPHS method) and SPHS-HCT442e (our default method using a higher order momentum equation) are tested to higher resolution ( $N_{1D} = 256$ ). The thick black line shows a scaling of  $N_{1D}^{-1.4}$  (the best-possible scaling is  $N_{1D}^{-2}$ ).

appears to be uninterrupted even at  $N_{1D} = 256$ , though the rate is perhaps slowing. The dotted black line shows the results for the same simulation but run using glass initial conditions. The error is slightly improved, but the rate of convergence is identical. This demonstrates that our results are not sensitive to the initial particle distribution. Finally, the solid magenta line shows results for our default SPHS scheme, but using a higher order momentum equation. For this simulation, we replaced equation (7), with the left-hand term in equation (18) – i.e. explicitly subtracting away the  $E_0$  error (SPHS-HCT442e). In this case, we should expect to see a steady convergence rate, without any need to further increase the neighbour number. Indeed, this is what is seen. The results are now significantly better than any of the other SPH methods. At the very

highest resolution ( $N_{1D} = 256$ ), however, there may still be some slowing in the convergence rate. It is possible that this is simply a fluctuation (running the test at even higher resolution to test this seems extravagant). Alternatively, it may be that at these resolutions the viscosity does start to play a role, slowing the convergence rate.

The most promising results for this particular test come from the SPHS-HCT442e method that uses a higher order momentum equation. Unfortunately, this same equation violates pairwise momentum conservation between particles and causes problems in strong shocks (Section 3; and see Morris 1996a). As such, we defer the investigation of such schemes to future work. We note here, however, that our default scheme – SPHS-HCT442 – still performs very well on this test, achieving percent level accuracy with increasing resolution and showing no indication that convergence is reaching a plateau, even at the highest resolution considered.

Our results suggest that in practice, we may be able to ensure that  $E_{0,i}$  shrinks faster than  $h_i$  without indefinitely increasing the neighbour number and kernel order (for a discussion of formal convergence criteria for SPH, see Section 3). The neighbour number and kernel order, however, control the *rate* of convergence.

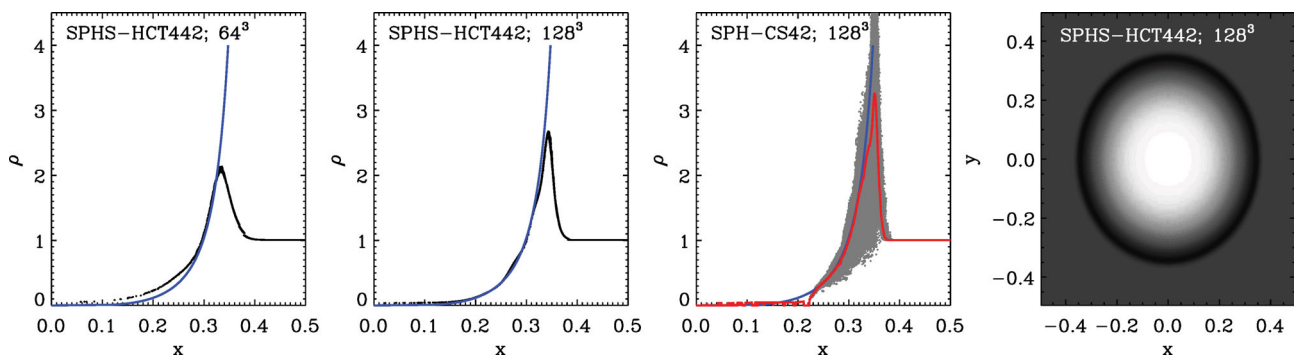
#### 7.4 Sedov–Taylor blast wave

We set up a Sedov–Taylor blast-wave test as in Springel & Hernquist (2002) using a uniform lattice of  $64^3$  particles with initial density  $\rho = 1$ . We inject an explosion energy  $E = 1$  into a central region  $r < 0.08$ . This corresponds to an initial entropy per central particle of  $A = 350$ . The remaining particles are assigned  $A = 0.05$ , giving an entropy contrast of  $\sim 7000$ . The analytic similarity solution to this problem is well known (see e.g. Landau & Lifshitz 1966), and gives a time evolution for the blast wave radius of

$$r(t) = 1.15 \left( \frac{Et^2}{\rho} \right)^{1/5} \quad (47)$$

for an adiabatic index of  $\gamma = 5/3$ .

The Sedov–Taylor test is particularly challenging for any hydrodynamical code because of the extreme entropy gradient in the initial conditions. The results for SPHS for  $N = 64^3$  and  $128^3$  particles are given in the left two panels of Fig. 7. As the resolution is increased, the results converge on the analytic solution shown in blue: the peak density of the shock increases, while the low-density tail better matches the analytic expectations. The blast wave is perfectly symmetric, as shown in the rightmost panel. For comparison,



**Figure 7.** Sedov–Taylor blast wave test results. Left three panels: the density profile of the gas at  $t = 0.05$  for  $N = 64^3$  and  $128^3$  particles for SPHS-HCT442, and for  $N = 128^3$  particles for SPH-CS42 (using the time-step limiter described in Section 5). The blue lines mark the analytic solution. For the SPHS simulations, the actual unbinned point particle densities are plotted in black; for the SPH simulation, they are plotted in grey. Notice the significantly larger noise for the SPH simulation. A mean binned profile, using a bin size of  $\Delta x = 0.001$  is overplotted in red. Rightmost panel: logarithmic density contours of the blast wave viewed from top at  $t = 0.05$  for SPHS-HCT442 with  $N = 128^3$  particles.

the results for classic SPH (with  $N = 128^3$ ) are shown in the third panel. Notice that the result is significantly more noisy (compare the grey dots with the black dots in the left two panels). The reduced noise in SPHS is partly due to the increased neighbour number, and partly due to the entropy dissipation (see for example similarly less noisy results for this test reported in Rosswog & Price 2007). The mean solution for SPH is, however, in good agreement with the analytic solution (compare the red and blue lines for the SPH-CS42 panel). Note that, for this test we had to use the time-step limiter described in Section 5 (and see Saitoh & Makino 2009), and so this simulation is not strictly speaking ‘classic’ SPH as we have defined it in this paper. Similar results can be obtained with classic SPH by using fixed time-steps and a sufficiently small Courant factor (equation 43). However, this is computationally very expensive.

One interesting aspect of the Sedov test is that it allows us to compare the spatial resolution in SPHS-HCT442 with classic SPH using 42 neighbours (SPH-CS42). Notice that the SPH-CS42 simulation resolves higher density in the shock. The unbinned particles reach densities up to  $\rho_{\max} \sim 4.5$  in simulation units, whereas our default scheme (SPH-HCT442) manages only  $\rho_{\max} = 2.7$ . We argued in Section 3 that the HOCT4 kernel with 442 neighbours should degrade the spatial resolution by a factor  $f \sim 1.5$  as compared to the CS kernel with 42 neighbours. Since the shock front for the Sedov test is one-dimensional, then we can expect a lower peak density in SPHS-HCT442 of a factor  $\sim f$ . This is almost exactly what is seen since  $\rho_{\max, \text{SPH}}/\rho_{\max, \text{SPHS}} = 1.67$ . In practice, however, the spatial resolution of the SPH simulation is not this good because of the increased noise. We ought to trust only the averaged solution that is shown in red. For this averaged density, the peak is significantly lower, with  $\rho_{\max} \sim 3.3$  – only a factor of 1.2 better than our default SPHS method. We conclude from this that SPHS-HCT442 does not significantly degrade the spatial resolution as compared to SPH-CS42, especially once the reduced noise of the method is taken into account.

### 7.5 Kelvin–Helmholtz test

We set up a 1:8 density contrast KH test in 3D as in RHA10. We used a periodic thin slab defined by  $x \in \{-0.5, 0.5\}$ ,  $y \in \{-0.5, 0.5\}$  and  $z \in \{-1/64, 1/64\}$ . The domain satisfied

$$\rho, T, v_x = \begin{cases} \rho_1, T_1, v_1, & |y| < 0.25, \\ \rho_2, T_2, v_2, & |y| > 0.25. \end{cases} \quad (48)$$

The density and temperature ratio were  $R_\rho = \rho_1/\rho_2 = T_2/T_1 = c_2^2/c_1^2$ , ensuring that the whole system was in pressure equilibrium. The two layers were given constant and opposing shearing velocities, with the low-density layer moving at a Mach number  $\mathcal{M}_2 = -v_2/c_2 \approx 0.11$  and the dense layer moving at  $\mathcal{M}_1 = \mathcal{M}_2\sqrt{R_\rho}$ . The density ratios considered in this work are small which assures a subsonic regime where the growth of instabilities can be treated using equation (51) (Vietri, Ferrara & Miniati 1997).

To trigger instabilities, velocity perturbations were imposed on the two boundaries of the form

$$v_y = \delta v_y [\sin(2\pi(x + \lambda/2)/\lambda) \exp(-(10(y - 0.25))^2) - \sin(2\pi x/\lambda) \exp(-(10(y + 0.25))^2)], \quad (49)$$

where the perturbation velocity  $\delta v_y/v = 1/8$  and  $\lambda = 0.5$  is the wavelength of the mode.

The linear growth rate of the KHI is given by (Chandrasekhar 1961)

$$w = k \frac{(\rho_1 \rho_2)^{1/2} v}{(\rho_1 + \rho_2)}, \quad (50)$$

where  $k = 2\pi/\lambda$  is the wavenumber of the instability,  $\rho_1$  and  $\rho_2$  are the densities of the respective layers and  $v = v_1 - v_2$  is the relative shear velocity. The characteristic growth time for the KHI is then

$$\tau_{\text{KH}} \equiv \frac{2\pi}{w} = \frac{(\rho_1 + \rho_2)\lambda}{(\rho_1 \rho_2)^{1/2} v}. \quad (51)$$

We set up two simulations to satisfy the set-up described above. An equal mass particle simulation with  $N = 2359\,296$ , and a multi-mass version with  $N = 524\,288$ . The latter simulation used a *uniform grid* of particles, with mass ratio 1:8 to describe the density step. To satisfy pressure equilibrium everywhere, the temperatures were adjusted at the boundary to be consistent with the SPH density step that is smooth (cf. equation 6).

The results of the test at times  $\tau_{\text{KH}} = 1, 2$  and 3 are shown in Fig. 8. The top row shows the results for classic SPH (SPH-CS42); the middle row shows the results for our default SPHS scheme (SPHS-HCT442), using single mass particles; the bottom row shows the results for SPHS-HCT442 using multimass particles. The SPH results are poor, with no mixing observed between the fluid layers, similarly to what has been reported in previous works (e.g. Agertz et al. 2007; RHA10). By contrast, the SPHS results show the growth of KH rolls on the correct time-scale and resolved mixing into the fully non-linear regime. Furthermore, the single mass simulation and multimass simulation (middle and bottom panels) are in excellent agreement. They differ in the details of the non-linear evolution caused by the growth of smaller noise-seeded rolls. But the growth time for the primary KH roll is in excellent agreement with analytic expectations, while the non-linear evolution is qualitatively similar. The multimass simulation is slightly more diffusive due to the additional mass dissipation between particles at the boundary. However, this simulation (because of the lower particle number) ran almost five times faster.

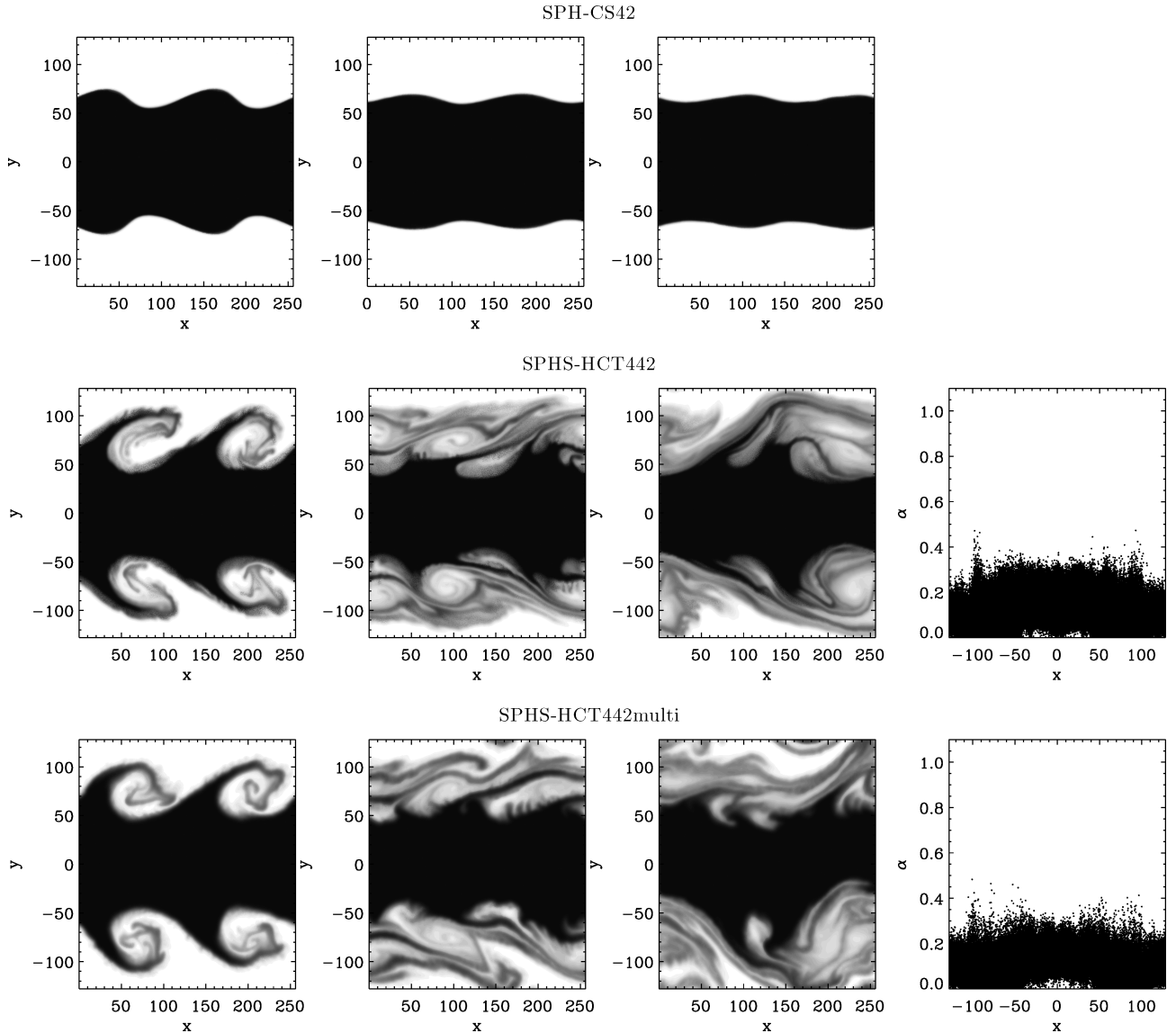
As discussed in our previous paper (RHA10), the improved performance for the KH test in SPHS is a result of both the improved force accuracy (due to the increased neighbour number and higher order, stable kernel), and the improved dissipation. We demonstrate this in Appendix F, where we show the effect of switching off the entropy and mass dissipation terms (equations 33 and 36) for this test.

### 7.6 The ‘blob’ test

The ‘blob’ test is a spherical cloud of gas of radius  $R_{\text{cl}}$  in a wind tunnel with periodic boundary conditions. The ambient medium is 10 times hotter and 10 times less dense than the cloud so that the system is in pressure equilibrium. The wind velocity ( $v_{\text{wind}} = c_s \mathcal{M}$ ) has an associated Mach number  $\mathcal{M} = 2.7$ . This leads to the formation of a bow shock after which the post-shock subsonic flow interacts with the cloud and turns supersonic as it flows past it. The test was first presented (with a full analytic analysis) in Agertz et al. (2007). Here, we set up the test as in RHA10 with  $N = 126\,744$  in the blob, arranged on a lattice. As in RHA10, we seed an initial inwards perturbation on the blob surface.

The results at times  $\tau_{\text{KH}} = 1$  (top), 2 (middle) and 3 (bottom) in classic SPH (left) and SPHS (right) are given in Fig. 9. In classic SPH, similarly to what has been reported in previous works, the blob does not break up and survives for the full length of the simulation.





**Figure 8.** KH1:8 test results. From left- to right-hand panels show logarithmic density contours at  $\tau_{\text{KH}} = 1, 2$  and  $3$ , and the dissipation parameter  $\alpha$  at  $\tau_{\text{KH}} = 1$ . The top panels show results for a single particle mass simulation ( $N = 2359296$ ); the bottom panels for a multimass simulation with a uniform density particle distribution ( $N = 524288$ ).

Furthermore, the suppression of instabilities at the fluid interface is sufficient to remove the inward perturbation that was seeded in the initial conditions. By contrast, this perturbation is clearly visible in the SPHS simulation and grows causing the blob to split down the middle in good agreement with both Eulerian codes and our OSPH method (RHA10). Finally, notice that the symmetry of the blob is well preserved even at  $\tau_{\text{KH}} = 3$  – well into the non-linear regime.

### 7.7 Conservation

Fig. 10 summarizes the conservation performance of SPHS for all of our tests. From left to right, we show the conservation of energy, momentum, angular momentum and (where relevant) mass. The results are normalized to a simulation time of 1, where ‘1’ is the maximum time presented in this paper (i.e. for the KH1:8 test this is  $\tau_{\text{KH}} = 3$ ). Momentum and angular momentum conservation results are only shown where these quantities are not zero in the initial conditions (to avoid a divide by zero in the percentage errors).

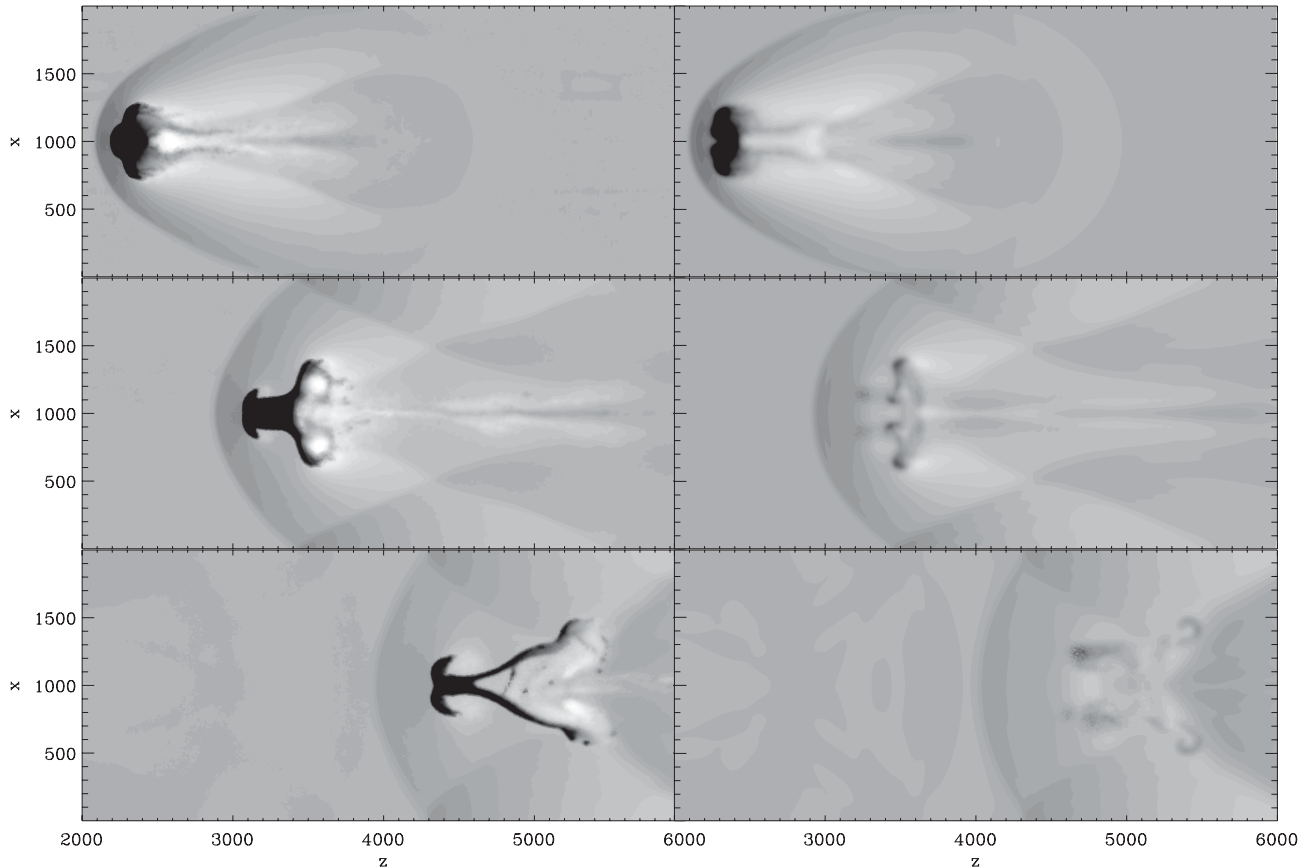
The worst performance is for the Sedov–Taylor test that conserves energy at the  $\sim 5$  per cent level. However all other tests conserve energy, mass, momentum and angular momentum to better than 1 per cent over the full simulation time.

### 7.8 Code performance

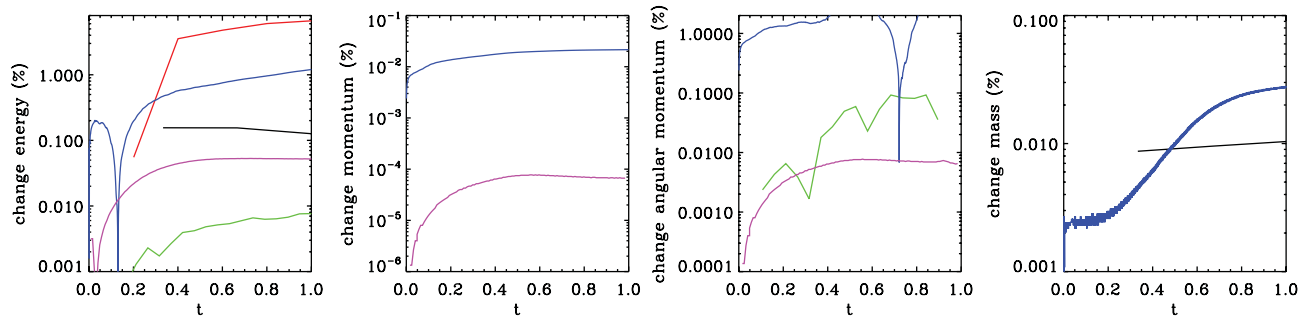
Fig. 11 compares the ratio of the speed of our default SPHS scheme (SPHS-HCT442; black squares) and SPH-CS442 (red squares) to classic SPH (SPH-CS42). (We use the simulation naming convention as described in Section 7.1.) For all tests, we used 16 processors. The Sod tests were compared at the  $N_{\text{ID}} = 600$  resolution, the vortex tests at  $N_{\text{ID}} = 128$  and the Sedov tests at  $N_{\text{ID}} = 128$ . In all cases, we eliminate the start-up time costs (the time taken to complete step zero). There is some significant variation in speed across all of the tests with the cost of SPHS ranging from two to four times that of SPH-CS42, but typically SPHS is three to four times slower at like particle number.

SPH-CS42

SPHS-HCT442



**Figure 9.** Blob test results at  $\tau_{KH} = 1$  (top), 2 (middle) and 3 (bottom) for classic SPH (left) and SPHS (right). All plots show logarithmic density contours.



**Figure 10.** Conservation in SPHS. From left to right, the panels show conservation of energy, momentum, angular momentum and (where relevant) mass for the simulation test suite presented in this paper. The coloured lines show results for the multimass Sod test (black), the Sedov–Taylor test (red; for  $N = 128^3$  particles), the Gresho vortex test (green; for  $N = 64 \times 64 \times 8$ ), the multimass KH1:8 test (blue) and the Blob test (purple). The results are normalized to a simulation time of 1, where ‘1’ is the maximum time presented in this paper (i.e. for the KH1:8 test this is  $\tau_{KH} = 3$ ). Momentum and angular momentum conservation results are only shown where these quantities are not zero in the initial conditions (to avoid a divide by zero in the percentage errors).

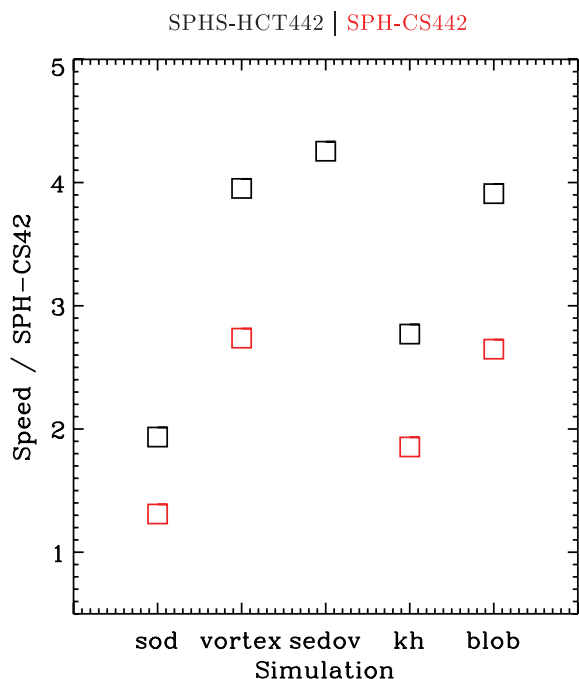
Note that the above speed tests are conservative. We could equally well conduct the tests at like *numerical error*, rather than like particle number. For the Gresho vortex test, for example, it is unlikely that SPH-CS42 can ever achieve equivalent accuracy to SPHS for any reasonable particle number (see Fig. 6). To obtain  $\sim 1$  per cent accuracy on this test, classic SPH would require an enormous particle number and be significantly slower than SPHS.

Finally, we have not made any attempt to optimize our current implementation of SPHS. Faster neighbour search algorithms, or neighbour caching could conceivably gain back much of the speed losses as compared to classic SPH. In addition, for real astrophysics applications, the additional work done on the neighbours may be

compensated by improved time-stepping (due to the reduced noise), and better load balancing in highly parallel simulations. Such considerations are beyond the scope of this present work.

## 8 CONCLUSIONS

We have presented an implementation of SPHS that has two novel features. The first is an improved treatment of dissipation. We use the spatial derivative of the velocity divergence as a higher order dissipation switch. Our switch – which is second-order accurate – detects flow convergence *before* it occurs. If particle trajectories are going to cross, we switch on the usual SPH artificial viscosity, as



**Figure 11.** The ratio of the speed of our default SPHS scheme (SPHS-HCT442; black squares) and SPH-CS442 (red squares) to ‘classic’ SPH (SPH-CS442). (We use the simulation naming convention as described in Section 7.1.) For all tests, we used 16 processors. The Sod tests were compared at the  $N_{1D} = 600$  resolution, the vortex tests at  $N_{1D} = 128$  and the Sedov tests at  $N_{1D} = 128$ . In all cases, we eliminate the start-up time costs (the time taken to complete step zero).

well as conservative dissipation in all advected fluid quantities (e.g. the entropy). The viscosity and dissipation terms (that are numerical errors) are designed to ensure that all fluid quantities remain single valued as particles approach one another, to respect conservation laws, and to vanish on a given physical scale as the resolution is increased. The second novel feature is the use of significantly larger neighbour number (442) to improve the force accuracy. As in our previous work, we use a novel kernel function that is (i) higher order such that the spatial resolution is not significantly degraded by our larger neighbour number, and (ii) that has a constant first derivative in the centre to prevent particle clumping (this latter ensures a smooth particle distribution on the kernel scale, which is necessary to obtain the improvement in the force accuracy).

We have demonstrated that SPHS alleviates a number of known problems with ‘classic’ SPH,<sup>10</sup> successfully resolving mixing and recovering numerical convergence with increasing resolution. An additional key advantage is that – treating the particle mass similarly to the entropy – we are able to use multimass particles, giving significantly improved control over the refinement strategy.

We have presented a wide range of code tests: the Sod shock tube, Sedov–Taylor blast wave, Gresho vortex, KH instability, the ‘blob test’ and some convergence tests. Our method performed well on all tests, giving good agreement with analytic expectations. For some tests, like the Gresho vortex, most of the improvement over ‘classic’ SPH is due to the improved force accuracy. For other tests like the (high density contrast) KH instability, the improved dissipation is equally important. We deliberately picked challenging tests that involve sharp features in one or more of the fluid quantities. These

are inherently difficult to resolve for our method that is manifestly smooth, yet we show that SPHS copes well even in such situations.

In our current implementation (that is likely suboptimal) SPHS is typically three to four times slower than ‘classic’ SPH (using 42 nearest neighbours) for like particle number. However, this additional cost should be offset against the improvement in the quality of the hydrodynamic solution in SPHS, the significantly reduced noise and the improved rate of convergence. For the Gresho vortex test, for example, SPHS achieves approximately per cent level accuracy as compared to  $\sim 10$  per cent in SPH for the same particle number.

The main remaining flaw in the SPHS algorithm is its low order. This means that formal convergence requires the neighbour number to be increased along with the particle number (using increasingly higher order stable kernels to maintain spatial resolution). However, our default kernel choice – the HOCT4 kernel with 442 neighbours – is already sufficient to obtain per cent level accuracy on the hydrodynamic tests we present here, and shows no sign of reaching a plateau in convergence even at the highest resolutions we consider. It is possible that convergence in practice does not require ever-increasing neighbour numbers, but rather the neighbour number and kernel order simply set the *rate* of convergence. Either way, it is unlikely that the neighbour number/kernel order will need to be increased beyond our default HOCT4 kernel with 442 neighbours for most astrophysical applications of interest.

SPHS will be useful for any astrophysics application involving multiphase fluid flow (e.g. resolving the ISM in galaxy discs), or where the use of multimass particles would be advantageous. The improved treatment of ram pressure stripping will likely be important for cosmological simulations where galaxies and clusters form through a sequence of hierarchical mergers (e.g. White & Rees 1978). This may explain the striking difference between Eulerian and SPH codes for the entropy profile of gas in simulated adiabatic galaxy clusters (the Santa Barbara test; Frenk et al. 1999). SPHS is also likely to be important for simulations involving rapid gas cooling, where multiphase flow is unavoidable. In particular, SPH simulations of cooling galaxy haloes are found to produce cold gas clouds in the halo (Kaufmann et al. 2009), while Eulerian codes do not (Joung, Bryan & Putman 2012). We will use SPHS to explore astrophysics problems like these in a series of forthcoming papers.

## ACKNOWLEDGMENTS

JIR would like to acknowledge support from SNF grant PPO0P2\_128540/1. This research used the ALICE High Performance Computing (HPC) facility at the University of Leicester and the Brutus HPC facility at the ETH Zürich. We would like to thank Peter Creasey for kindly supplying his Sedov–Taylor test initial conditions and solution, and for pointing out that OSPH does not perform well for the Sedov–Taylor test. We would also like to thank Stuart Muldrew and Frazer Pearce for sharing their version of the Saitoh & Makino (2009) time-stepping algorithm; Walter Dehnen, Lee Cullen, Chris Nixon, Alex Hobbs and Romain Teyssier for useful discussions; the referee Spyridon Kitsionas for useful comments that have improved the clarity of this work. Finally, we thank Volker Springel for making his GADGET-2 code available to the community, and for useful discussions.

## REFERENCES

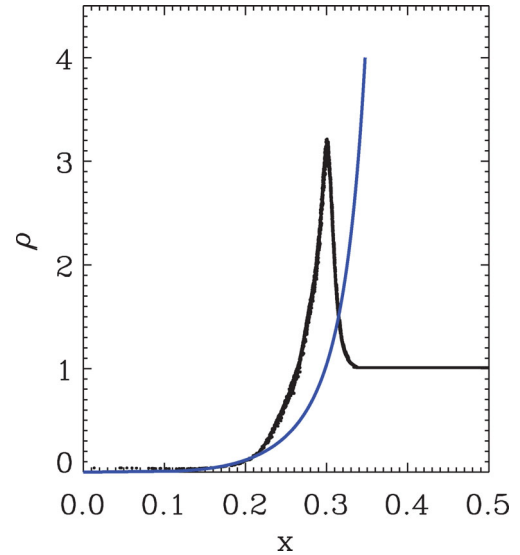
- Abel T., 2011, MNRAS, 413, 129  
 Agertz O. et al., 2007, MNRAS, 380, 963

<sup>10</sup> We define ‘classic’ SPH as that implemented in the public release version of the GADGET-2 code (Springel 2005), and similar.

- Balsara D. S., 1989, PhD thesis, Univ. Illinois at Urbana-Champaign  
 Cha S., Inutsuka S., Nayakshin S., 2010, MNRAS, 403, 1165  
 Chandrasekhar S., 1961, Hydrodynamic and Hydromagnetic Stability. Clarendon Press, Oxford  
 Couchman H. M. P., Thomas P. A., Pearce F. R., 1995, ApJ, 452, 797  
 Cullen L., Dehnen W., 2010, MNRAS, 408, 669  
 Dehnen W., 2000, ApJ, 536, L39  
 Dehnen W., Read J. I., 2011, European Phys. J. Plus, 126, 55  
 Dills G., 1999, Int. J. Numer. Methods Eng., 44, 1115  
 Fan J., Gijbels I., 1996, Local Polynomial Modelling and Its Applications. Chapman & Hall, London  
 Frenk C. S. et al., 1999, ApJ, 525, 554  
 Gaburov E., Nitadori K., 2011, MNRAS, 414, 129  
 Gingold R. A., Monaghan J. J., 1977, MNRAS, 181, 375  
 Gresho P. M., Chan S. T., 1990, Int. J. Numer. Methods Fluids, 11, 621  
 Heß S., Springel V., 2010, MNRAS, 406, 2289  
 Inutsuka S. I., 2002, J. Comput. Phys., 179, 238  
 Joung M. R., Bryan G. L., Putman M. E., 2012, ApJ, 745, 148  
 Kaufmann T., Bullock J. S., Maller A. H., Fang T., Wadsley J., 2009, MNRAS, 396, 191  
 Kawata D., Okamoto T., Cen R., Gibson B. K., 2009, arXiv e-prints (0902.4002)  
 Kitsionas S., Whitworth A. P., 2002, MNRAS, 330, 129  
 Landau L. D., Lifshitz E. M., 1966, Hydrodynamik: Lehrbuch der Theoretischen Physik. Akademie-Verlag, Berlin  
 Lucy L. B., 1977, AJ, 82, 1013  
 Maron J. L., Howes G. G., 2003, ApJ, 595, 564  
 Meglicki Z., Wickramasinghe D., Bicknell G. V., 1993, MNRAS, 264, 691  
 Monaghan J. J., 1992, ARA&A, 30, 543  
 Monaghan J. J., 1997, J. Comput. Phys., 136, 298  
 Monaghan J. J., Varnas S. R., 1988, MNRAS, 231, 515  
 Morris J. P., 1995, arXiv (astro-ph/9503124)  
 Morris J. P., 1996a, Publ. Astron. Soc. Australia, 13, 97  
 Morris J. P., 1996b, PhD thesis, Department of Mathematics, Monash University  
 Morris J. P., Monaghan J. J., 1997, J. Comput. Phys., 136, 41  
 Murante G., Borgani S., Brunino R., Cha S.-H., 2011, MNRAS, 417, 136  
 Oger G., Doring M., Alessandrini B., Ferrant P., 2007, J. Comput. Phys., 225, 1472  
 Ott F., Schnetter E., 2003, arXiv (physics/0303112)  
 Price D., 2005, arXiv (astro-ph/0507472)  
 Price D. J., 2008, J. Comput. Phys., 227, 10040  
 Price D. J., 2012, J. Comput. Phys., 231, 759  
 Read J. I., Hayfield T., Agertz O., 2010, MNRAS, 405, 1513 (RHA10)  
 Ritchie B. W., Thomas P. A., 2001, MNRAS, 323, 743  
 Rosswog S., 2009, New Astron. Rev., 53, 78  
 Rosswog S., 2010, J. Comput. Phys., 229, 8591  
 Rosswog S., Price D., 2007, MNRAS, 379, 915  
 Saitoh T. R., Makino J., 2009, ApJ, 697, L99  
 Springel V., 2005, MNRAS, 364, 1105  
 Springel V., 2010a, MNRAS, 401, 791  
 Springel V., 2010b, ARA&A, 48, 391  
 Springel V., Hernquist L., 2002, MNRAS, 333, 649  
 Valcke S., de Rijcke S., Rödigier E., Dejonghe H., 2010, MNRAS, 408, 71  
 Vietri M., Ferrara A., Miniati F., 1997, ApJ, 483, 262  
 Wadsley J. W., Stadel J., Quinn T., 2004, New Astron., 9, 137  
 Wadsley J. W., Veeravalli G., Couchman H. M. P., 2008, MNRAS, 387, 427  
 White S. D. M., Rees M. J., 1978, MNRAS, 183, 341

## APPENDIX A: THE TROUBLE WITH ‘RT’ PRESSURES

In RHA10, we used the same equations of motion as SPHS, but with the pressure estimator in equation (20). This ensured manifestly smooth pressures throughout the flow, allowing us to successfully model mixing between different fluid phases. However, while equation (20) gives excellent performance for multiphase



**Figure A1.** Sedov-Taylor blast wave test results using the ‘RT’ pressure estimator (equation 20) without entropy dissipation. The plot shows the density profile of the gas at time  $t = 0.05$ , similarly to Fig. 7. Notice that the shock front is displaced with respect to the analytic curve (blue).

flow applications, it performs poorly in strong shocks where the entropy gradients on the kernel scale are large. We show this in Fig. A1, where we plot results for the Sedov–Taylor blast wave problem (with  $N = 128^3$ ; and see Section 7.4), using the ‘RT’ pressure estimator (equation 20) without entropy dissipation. As can be seen, the resulting shock front, while very smooth, is not in good agreement with the analytic curve shown in blue.

The above highlights the key problem with ‘RT’ densities and pressures. Particles in the kernel with very different entropies are down-weighted in the sum. This means that to obtain good kernel sampling, we must scale the neighbour number with the entropy contrast on the kernel scale. For the Sedov–Taylor blast wave, where the initial entropy contrast is  $\sim 7000$ , this is prohibitively expensive. Not doing this, however, leads to a significant numerical error as can be seen in Fig. A1. For this reason, in this paper, we have abandoned the density and pressure estimators given in equations (19) and (20). Instead, we ensure smooth pressures through our higher order dissipation switch described in Section 4.

## APPENDIX B: A FULLY CONSERVATIVE VERSION OF SPHS

A fully conservative version of SPHS can be constructed by replacing equation (7) with that in Springel & Hernquist (2002):

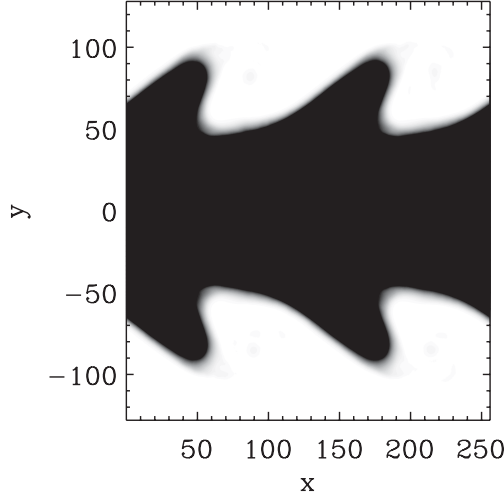
$$\frac{d\mathbf{v}_i}{dt} = - \sum_j^N m_j \left[ f_i \frac{P_i}{\rho_i^2} \nabla_i W_{ij}(h_i) + f_j \frac{P_j}{\rho_j^2} \nabla_j W_{ij}(h_j) \right], \quad (\text{B1})$$

where the function  $f_i$  is a correction factor that ensures energy conservation for varying smoothing lengths:

$$f_i = \left( 1 + \frac{h_i}{3\rho_i} \frac{\partial \rho_i}{\partial h_i} \right)^{-1}. \quad (\text{B2})$$

As discussed in RHA10, the above momentum equation gives improved (in fact manifest) energy conservation, but larger truncation error. For applications where energy conservation is of paramount importance (for example, where a system is evolved for many dynamical times), the above equation should be used. However, in this





**Figure B1.** KH1:8 multimass test results at  $\tau_{\text{KH}} = 1$  for a fully conservative version of SPHS (that uses equation B1 instead of equation 7). Notice that the results are significantly more diffusive than our default scheme shown in Fig. 8.

case, care must also be taken over the time-stepping (e.g. Dehnen & Read 2011). For the tests presented in this paper, the energy losses due to variable time-steps dominate and the above momentum equation gains only  $\sim 0.5$  per cent in energy conservation for the KH1:8 multimass test (see Section 7.7), and  $\sim 2$  per cent for the Sedov test (a factor of  $\sim 2$  improvement in both cases). However, the larger truncation error introduces significantly more diffusion. This is shown in Fig. B1, where we present results for the multimass KH1:8 test (see Section 7.5) at time  $\tau_{\text{KH}} = 1$ , using equation (B1). For this reason, our default choice for SPHS is the momentum equation (7).

### APPENDIX C: THE SENSITIVITY OF SPHS TO THE DISSIPATION PARAMETERS

In this appendix, we assess how sensitive SPHS is to the choice of dissipation parameters. As our default, we have assumed a single maximum dissipation parameter for viscosity, mass dissipation and entropy dissipation:  $\alpha_{\text{max}} = \alpha_v = \alpha_m = \alpha_A = 1$ . This default choice is natural from the definition of the dissipation/viscosity equations (29), (31), (33) and (36). These assert that the dissipation should proceed proportional to the jump in the given fluid quantity (mass, entropy, etc.) and on a time-scale set by the signal velocity. Thus,

we expect a normalization parameter in each case of order unity. Nonetheless,  $\alpha$  is a free parameter and we should check that our results are not sensitive to it. To test this, in Fig. C1, we consider the effect of varying  $\alpha_v$  and  $\alpha_A$  for the Sod shock tube test (Section 7.2) at two different resolutions.

From Fig. C1, we see that our results are not sensitive to the entropy dissipation parameter  $\alpha_A$ . Over a wide range  $0.1 < \alpha_A < 5$ , the results change only slightly. More importantly, the differences decrease with increasing resolution (compare the red and black lines in the left two panels of Fig. C1). The results are more sensitive, however, to the choice of viscosity parameter  $\alpha_v$ . For low viscosity ( $\alpha_v = 0.1$ ), we have spurious oscillations in the solution. Reassuringly, however, these decrease with increasing resolution (compare the green and black curves in the right two panels of Fig. C1). The results are poor, however, if  $\alpha_v$  is too large. For  $\alpha_v = 5$ , there is a strong undershoot in the density at the shock that does not improve with increasing resolution.

We conclude that the results in SPHS converge with increasing resolution independently of  $\alpha_v$  or  $\alpha_A$ , as long as  $\alpha_v$  is not too large.

### APPENDIX D: FITTING AN $N$ TH ORDER POLYNOMIAL TO A FLUID QUANTITY

We describe here an algorithm for fitting an order  $N$  polynomial to an irregular point distribution (see e.g. Fan & Gijbels 1996; Maron & Howes 2003). We give the relevant equations for a second order fit in three dimensions, but the method straightforwardly generalizes to arbitrary order and dimension. Assuming that a fluid quantity  $q_i$  defined at particle position  $i$  is smooth (and therefore differentiable), we can perform a second-order polynomial expansion at a point  $j$  about  $i$ :

$$q_{ij} = a_{0,i} + a_{1,i}x_{ij} + a_{2,i}y_{ij} + a_{3,i}z_{ij} + a_{4,i}x_{ij}^2 + a_{5,i}y_{ij}^2 + a_{6,i}z_{ij}^2 + a_{7,i}x_{ij}y_{ij} + a_{8,i}x_{ij}z_{ij} + a_{9,i}y_{ij}z_{ij} + O(h^3), \quad (\text{D1})$$

where  $\mathbf{x}_{ij} = \mathbf{r}_{ij}/h_i = [x_{ij}, y_{ij}, z_{ij}]$ .

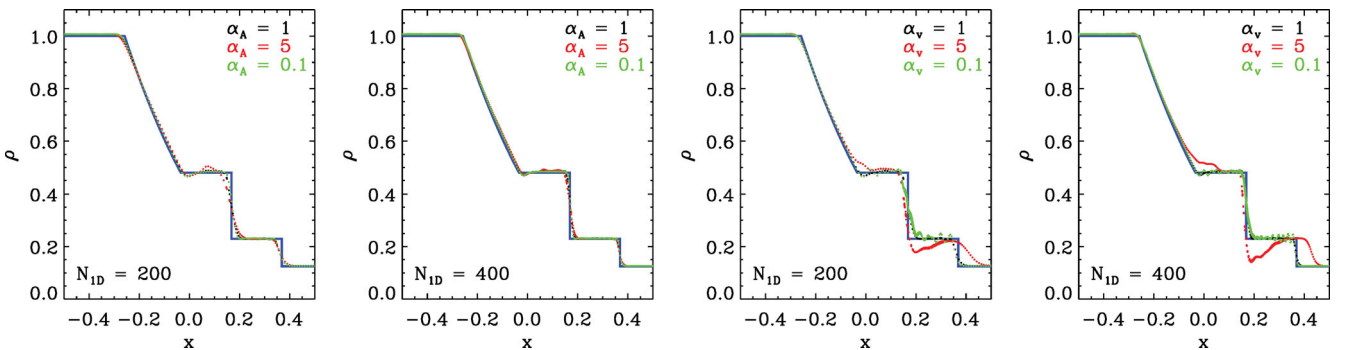
The coefficients of this expansion can then be determined by inverting the following  $10 \times 10$  matrix equation:

$$\mathbf{M}\mathbf{a} = \mathbf{q}, \quad (\text{D2})$$

where

$$\mathbf{a}^T = [a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9], \quad (\text{D3})$$

$$\mathbf{q}^T = \sum_j^N m_j q_j \overline{W}_{ij} \begin{bmatrix} 1, x_{ij}, y_{ij}, z_{ij}, x_{ij}^2, y_{ij}^2, z_{ij}^2, \\ x_{ij}y_{ij}, x_{ij}z_{ij}, y_{ij}z_{ij} \end{bmatrix}, \quad (\text{D4})$$



**Figure C1.** Sensitivity to the dissipation parameters for the Sod shock tube test (Section 7.2). From left to right, the panels show the density profile at  $t = 0.2$  (similarly to Fig. 3), for varying entropy function dissipation parameter  $\alpha_A$  and viscous dissipation parameter  $\alpha_v$ , at low resolution  $N_{\text{ID}} = 200$  and higher resolution  $N_{\text{ID}} = 400$ , as marked. The blue line marks the analytic solution. Notice that so long as  $\alpha_v$  is not too large, the results converge with increasing resolution independently of the choice of dissipation parameters.

$$\mathbf{M} = \sum_j^N m_j \overline{W}_{ij} \begin{pmatrix} 1 & x_{ij} & y_{ij} \cdots \\ x_{ij} & x_{ij}^2 & x_{ij} y_{ij} \cdots \\ y_{ij} & y_{ij} x_{ij} & y_{ij}^2 \cdots \\ z_{ij} & z_{ij} x_{ij} & z_{ij} y_{ij} \cdots \\ x_{ij}^2 & x_{ij}^3 & x_{ij}^2 y_{ij} \cdots \\ y_{ij}^2 & y_{ij}^2 x_{ij} & y_{ij}^3 \cdots \\ z_{ij}^2 & z_{ij}^2 x_{ij} & z_{ij}^2 y_{ij} \cdots \\ x_{ij} y_{ij} & x_{ij}^2 y_{ij} & x_{ij} y_{ij}^2 \cdots \\ x_{ij} z_{ij} & z_{ij} x_{ij}^2 & x_{ij} z_{ij} y_{ij} \cdots \\ y_{ij} z_{ij} & y_{ij} z_{ij} x_{ij} & z_{ij} y_{ij}^2 \cdots \\ \cdots z_{ij} & x_{ij}^2 & y_{ij}^2 & z_{ij}^2 \cdots \\ \cdots x_{ij} z_{ij} & x_{ij}^3 & x_{ij} y_{ij}^2 & x_{ij} z_{ij}^2 \cdots \\ \cdots y_{ij} z_{ij} & y_{ij} x_{ij}^2 & y_{ij}^3 & y_{ij} z_{ij}^2 \cdots \\ \cdots z_{ij}^2 & z_{ij} x_{ij}^2 & z_{ij} y_{ij}^2 & z_{ij}^3 \cdots \\ \cdots x_{ij}^2 z_{ij} & x_{ij}^4 & x_{ij}^2 y_{ij}^2 & x_{ij}^2 z_{ij}^2 \cdots \\ \cdots y_{ij}^2 z_{ij} & y_{ij}^2 x_{ij}^2 & y_{ij}^4 & y_{ij}^2 z_{ij}^2 \cdots \\ \cdots z_{ij}^3 & z_{ij}^2 x_{ij}^2 & z_{ij}^2 y_{ij}^2 & z_{ij}^4 \cdots \\ \cdots x_{ij} y_{ij} z_{ij} & y_{ij} x_{ij}^3 & x_{ij} y_{ij}^3 & x_{ij} y_{ij} z_{ij}^2 \cdots \\ \cdots x_{ij} z_{ij}^2 & z_{ij} x_{ij}^3 & x_{ij} z_{ij} y_{ij}^2 & x_{ij} z_{ij}^3 \cdots \\ \cdots y_{ij} z_{ij}^2 & y_{ij} z_{ij} x_{ij}^2 & z_{ij} y_{ij}^3 & y_{ij} z_{ij}^3 \cdots \\ \cdots x_{ij} y_{ij} & x_{ij} z_{ij} & y_{ij} z_{ij} \\ \cdots x_{ij}^2 y_{ij} & x_{ij}^2 z_{ij} & x_{ij} y_{ij} z_{ij} \\ \cdots x_{ij} y_{ij}^2 & y_{ij} x_{ij} z_{ij} & y_{ij}^2 z_{ij} \\ \cdots z_{ij} x_{ij} y_{ij} & x_{ij} z_{ij}^2 & y_{ij} z_{ij}^2 \\ \cdots x_{ij}^3 y_{ij} & x_{ij}^3 z_{ij} & x_{ij}^2 y_{ij} z_{ij} \\ \cdots x_{ij} y_{ij}^3 & y_{ij}^2 x_{ij} z_{ij} & y_{ij}^3 z_{ij} \\ \cdots z_{ij}^2 x_{ij} y_{ij} & x_{ij} z_{ij}^3 & y_{ij} z_{ij}^3 \\ \cdots x_{ij}^2 y_{ij}^2 & y_{ij} x_{ij}^2 z_{ij} & x_{ij} y_{ij}^2 z_{ij} \\ \cdots z_{ij} x_{ij}^2 y_{ij} & x_{ij}^2 z_{ij}^2 & x_{ij} y_{ij} z_{ij}^2 \\ \cdots z_{ij} x_{ij} y_{ij}^2 & y_{ij} x_{ij} z_{ij}^2 & y_{ij}^2 z_{ij}^2 \end{pmatrix} \quad (\text{D5})$$

and  $\overline{W}_{ij} = (1/2)[W_{ij}(h_i) + W_{ij}(h_j)]$  is the symmetrized smoothing kernel (the superscript  $\text{T}$  means transpose).

Having determined all of the coefficients of  $\mathbf{a}$  (by solving  $\mathbf{a} = \mathbf{M}^{-1}\mathbf{q}$ ), the gradients of  $q$  evaluated at  $i$  then simply follow as

$$\frac{\partial q_i}{\partial x} = a_1, \quad \frac{\partial q_i}{\partial y} = a_2, \quad \frac{\partial q_i}{\partial z} = a_3, \quad (\text{D6})$$

and similarly for the second derivatives.

## APPENDIX E: A GENERAL DERIVATION OF CONSERVATION TERMS FOR MULTIMASS SPHS

In Section 4.3.3, we introduced a multimass dissipation term for SPHS. This requires some correction terms to restore energy and momentum conservation. In this appendix, we derive the general class of such correction terms.

Our dissipation terms must obey mass, momentum and energy conservation:

$$\dot{M} = 0 = \sum_j \dot{m}_j, \quad (\text{E1})$$

$$\dot{M}\mathbf{V} = 0 = \sum_j m_j \dot{\mathbf{v}}_j = \sum_j m_j \dot{\mathbf{v}}_j + \dot{m}_j \mathbf{v}_j, \quad (\text{E2})$$

$$\dot{E} = 0 = \sum_j \dot{m}_j \left( \frac{1}{2} \mathbf{v}_j \cdot \mathbf{v}_j + u_j \right) + m_j (\mathbf{v}_j \cdot \dot{\mathbf{v}}_j + \dot{u}_j). \quad (\text{E3})$$

First, let us verify that equation (36) satisfies equation (E1). Substituting for  $\dot{m}_j = \dot{m}_{\text{diss},j}$ , we have

$$\dot{M} = \sum_{j,k} Q_{jk} (m_j - m_k) K_{jk} = 0, \quad (\text{E4})$$

where  $Q_{ij} = Q_{ji} = \overline{m}_{ij} / \overline{\rho}_{ij} \overline{\alpha}_{ij} v_{\text{sig},ij}^p L_{ij}$  and the above is zero because it is antisymmetric in  $j, k$ . Note that this explains why we must use a symmetrized mass in equation (36):  $Q_{ij}$  must be symmetric in order to ensure mass conservation.

Now, let us substitute  $\dot{m}_j = \dot{m}_{\text{diss},j}$  into equation (E2):

$$0 = \sum_j m_j \dot{\mathbf{v}}_{\text{diss},j} + \sum_{j,k} Q_{jk} (m_j - m_k) K_{jk} \mathbf{v}_j, \quad (\text{E5})$$

where we have split the acceleration into a dissipative correction term, and all other normal SPHS terms:  $\dot{\mathbf{v}}_i = \dot{\mathbf{v}}_{\text{diss},i} + \dot{\mathbf{v}}_{\text{rest},i}$ , and then used the fact that  $\sum_j m_j \dot{\mathbf{v}}_{\text{rest},j} = 0$  by construction for SPHS.

We may now select any form we like for  $\dot{\mathbf{v}}_{\text{diss},i}$  so long as it satisfies equation (E5). In Section 4.3.3, we chose a form that conserves momentum on a per particle basis, but we may also choose a form that fluxes the momenta, for example:

$$\dot{\mathbf{v}}_i = \sum_j Q_{ij} \frac{(m_i - m_j)}{m_i} K_{ij} \mathbf{v}_j. \quad (\text{E6})$$

It is straightforward to show that the above correction term also conserves momentum since it makes equation (E5) antisymmetric in  $j, k$ .

We may then derive a similar constraint equation for our energy correction term. As an example, let us substitute  $\dot{m}_j = \dot{m}_{\text{diss},j}$  and equation (E6) into equation (E3). This gives

$$0 = \sum_{j,k} (Q_{jk} (m_j - m_k) K_{jk}) \left[ \frac{1}{2} \mathbf{v}_j \cdot \mathbf{v}_j + u_j \right] + m_j \left[ \mathbf{v}_j \cdot \left( Q_{jk} \frac{(m_j - m_k)}{m_j} K_{jk} \mathbf{v}_k \right) + \dot{u}_{\text{diss},j} \right], \quad (\text{E7})$$

where similarly to the above, we have dropped all contributions involving the standard SPHS terms since these are already conservative and therefore vanish.

We may then derive a correction term for  $\dot{u}_{\text{diss},j}$ :

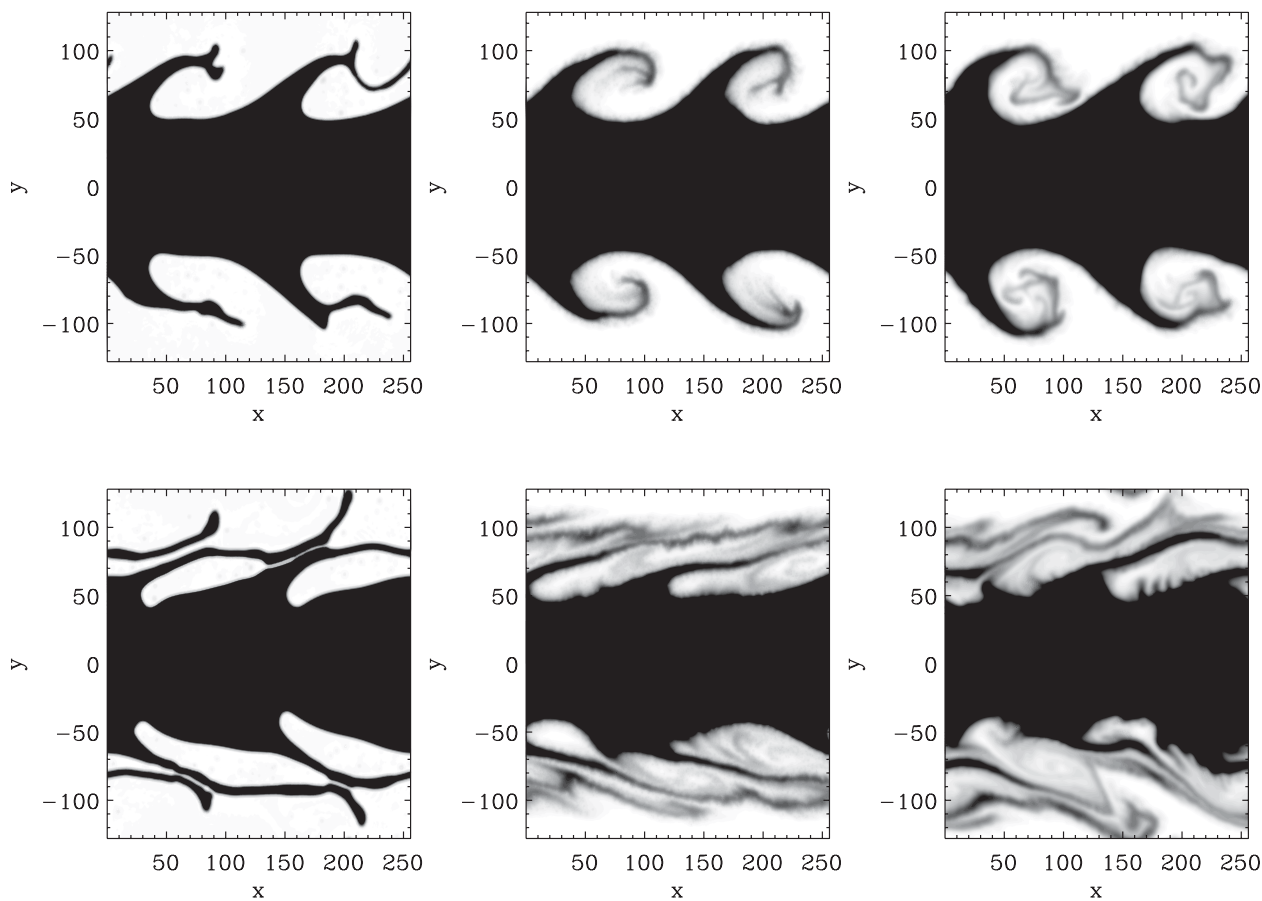
$$\dot{u}_{\text{diss},i} = \sum_j Q_{ij} \frac{(m_i - m_j)}{m_i} K_{ij} \left[ \frac{1}{2} \mathbf{v}_j \cdot \mathbf{v}_j + u_j \right]. \quad (\text{E8})$$

It is straightforward to verify that substituting equation (E8) into equation (E7) makes the equation antisymmetric in  $j, k$  and thus restores energy conservation.

It is clear from the above examples that we may use the above constraints to derive a whole class of correction terms. Some of these may give better performance than equations (37) and (38) that we use as our default in this paper. Such a study is, however, beyond the scope of this present work.

## APPENDIX F: THE IMPORTANCE OF DISSIPATION TERMS FOR MULTIPHASE AND MULTIMASS FLOW IN SPHS

In this appendix, we show the effect of switching off our dissipation terms in entropy and mass for the KH1:8 multimass test (Section 7.5). The results are shown in Fig. F1. As expected, the entropy dissipation is extremely important: without it there is no mixing between the different fluid phases (see left-hand panels). Notice,



**Figure F1.** KH1:8 multimass test results: the effect of removing the dissipation terms. From left to right the panels show logarithmic density contours at  $\tau_{\text{KH}} = 1$  (top) and  $\tau_{\text{KH}} = 2$  (bottom) for SPHS run without entropy or mass dissipation (left), without mass dissipation (middle) and the full SPHS scheme (right). (The right-hand panels reproduce the results from Fig. 8, bottom row.)

however, that even without dissipation, the KH rolls do grow on the correct time-scale unlike in the classic SPH simulation (Fig. 8, top row). This demonstrates (similarly to our findings in RHA10) that the improved force accuracy in SPHS is responsible for the correct growth rate of the rolls, while the improved dissipation is responsible for actual mixing between the different fluid phases.

The effect of the mass dissipation is more subtle. Without mass dissipation, mixing is also inhibited, but the effects are less strong than for the case without entropy dissipation because, unlike

the entropies, the masses are smoothed inside the density sum<sup>11</sup> (equation 6).

<sup>11</sup> In fact, the results for this test without mass dissipation are rather similar to the KH1:8 single mass test we presented using OSPH in RHA10. This similarity arises because in OSPH the entropy – like the particle masses – is smoothed inside the pressure estimator (equation 20).

This paper has been typeset from a  $\text{\TeX}/\text{\LaTeX}$  file prepared by the author.