Publications of the Astronomical Society of Australia (PASA), Vol. 35, e031, 82 pages (2018).
© Astronomical Society of Australia 2018; published by Cambridge University Press.
doi:10.1017/pasa.2018.25

PHANTOM: A Smoothed Particle Hydrodynamics and Magnetohydrodynamics Code for Astrophysics

Daniel J. Price[1,18], James Wurster[2,1], Terrence S. Tricco[3,1], Chris Nixon[4], Stéven Toupin[5], Alex Pettitt[6], Conrad Chan[1], Daniel Mentiplay[1], Guillaume Laibe[7], Simon Glover[8], Clare Dobbs[2], Rebecca Nealon[4,1], David Liptai[1], Hauke Worpel[9,1], Clément Bonnerot[10], Giovanni Dipierro[4], Giulia Ballabio[4], Enrico Ragusa[11], Christoph Federrath[12], Roberto Iaconi[13], Thomas Reichardt[13], Duncan Forgan[14], Mark Hutchison[15,16,1], Thomas Constantino[2], Ben Ayliffe[17,1], Kieran Hirsh[1] and Giuseppe Lodato[11]

[1]Monash Centre for Astrophysics (MoCA) and School of Physics and Astronomy, Monash University, Vic. 3800, Australia
[2]School of Physics, University of Exeter, Exeter EX4 4QL, UK
[3]Canadian Institute for Theoretical Astrophysics (CITA), University of Toronto, Toronto, ON M5S 3H8, Canada
[4]Theoretical Astrophysics Group, Department of Physics & Astronomy, University of Leicester, Leicester LE1 7RH, UK
[5]Institut d'Astronomie et d'Astrophysique (IAA), Université Libre de Bruxelles (ULB), CP226, Boulevard du Triomphe B1050 Brussels, Belgium
[6]Department of Cosmosciences, Hokkaido University, Sapporo 060-0810, Japan
[7]Centre de Recherche Astrophysique de Lyon, Univ Lyon, ENS de Lyon, CNRS, Saint-Genis-Laval F-69230, France
[8]Institut für Theoretische Astrophysik, Zentrum für Astronomie der Universität Heidelberg, D-69120 Heidelberg, Germany
[9]AIP Potsdam, An der Sternwarte 16, 14482 Potsdam, Germany
[10]Leiden Observatory, Leiden University, PO Box 9513, NL-2300 RA Leiden, the Netherlands
[11]Dipartimento di Fisica, Università Degli Studi di Milano, Via Celoria 16, Milano 20133, Italy
[12]Research School of Astronomy and Astrophysics, Australian National University, Canberra ACT 2611, Australia
[13]Department of Physics and Astronomy, Macquarie University, 2109 Sydney, Australia
[14]St Andrews Centre for Exoplanet Science and School of Physics and Astronomy, University of St. Andrews, North Haugh, St. Andrews, Fife KY16 9SS, UK
[15]Physikalisches Institut, Universität Bern, Gesellschaftstrasse 6, 3012 Bern, Switzerland
[16]Institute for Computational Science, University of Zurich, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland
[17]Met Office, FitzRoy Road, Exeter EX1 3PB, UK
[18]Email: daniel.price@monash.edu

(RECEIVED February 14, 2017; ACCEPTED June 15, 2018)

Abstract

We present PHANTOM, a fast, parallel, modular, and low-memory smoothed particle hydrodynamics and magnetohydrodynamics code developed over the last decade for astrophysical applications in three dimensions. The code has been developed with a focus on stellar, galactic, planetary, and high energy astrophysics, and has already been used widely for studies of accretion discs and turbulence, from the birth of planets to how black holes accrete. Here we describe and test the core algorithms as well as modules for magnetohydrodynamics, self-gravity, sink particles, dust–gas mixtures, H_2 chemistry, physical viscosity, external forces including numerous galactic potentials, Lense–Thirring precession, Poynting–Robertson drag, and stochastic turbulent driving. PHANTOM is hereby made publicly available.

Keywords: accretion, accretion disks – hydrodynamics – ISM: general – magnetohydrodynamics (MHD) – methods: numerical

1 INTRODUCTION

Numerical simulations are the 'third pillar' of astrophysics, standing alongside observations and analytic theory. Since it is difficult to perform laboratory experiments in the relevant physical regimes and over the correct range of length and timescales involved in most astrophysical problems, we turn instead to 'numerical experiments' in the computer for understanding and insight. As algorithms and simulation codes become ever more sophisticated, the public availability of simulation codes has become crucial to ensure that these experiments can be both verified and reproduced.

PHANTOM is a smoothed particle hydrodynamics (SPH) code developed over the last decade. It has been used widely for studies of turbulence (e.g. Kitsionas et al. 2009; Price & Federrath 2010; Price, Federrath, & Brunt 2011), accretion (e.g. Lodato & Price 2010; Nixon, King, & Price 2012a; Rosotti, Lodato, & Price 2012), star formation including non-ideal magnetohydrodynamics (MHD) (e.g. Wurster et al. 2016, Wurster, Price, & Bate 2017), star cluster formation

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

(Liptai et al. 2017), and for studies of the Galaxy (Pettitt et al. 2014; Dobbs et al. 2016), as well as for simulating dust–gas mixtures (e.g. Dipierro et al. 2015; Ragusa et al. 2017; Tricco, Price, & Laibe 2017). Although the initial applications and some details of the basic algorithm were described in Price & Federrath (2010), Lodato & Price (2010), and Price (2012a), the code itself has never been described in detail and, until now, has remained closed source.

One of the initial design goals of PHANTOM was to have a low memory footprint. A secondary motivation was the need for a public SPH code that is not primarily focused on cosmology, as in the highly successful GADGET code (Springel, Yoshida, & White 2001; Springel 2005). The needs of different communities produce rather different outcomes in the code. For cosmology, the main focus is on simulating the gravitational collapse of dark matter in large volumes of the universe, with gas having only a secondary effect. This is reflected in the ability of the public GADGET-2 code to scale to exceedingly large numbers of dark matter particles, yet neglecting elements of the core SPH algorithm that are essential for stellar and planetary problems, such as the Morris & Monaghan (1997) artificial viscosity switch [c.f. the debate between Bauer & Springel (2012) and Price (2012b)], the ability to use a spatially variable gravitational force softening (Bate & Burkert 1997; Price & Monaghan 2007) or any kind of artificial conductivity, necessary for the correct treatment of contact discontinuities (Chow & Monaghan 1997; Price & Monaghan 2005; Rosswog & Price 2007; Price 2008). Almost all of these have since been implemented in development versions of GADGET-3 (e.g. Iannuzzi & Dolag 2011; Beck et al. 2016; see recent comparison project by Sembolini et al. 2016) but remain unavailable or unused in the public version. Likewise, the implementation of dust, non-ideal MHD, and other physics relevant to star and planet formation is unlikely to be high priority in a code designed for studying cosmology or galaxy formation.

Similarly, the SPHNG code (Benz et al. 1990; Bate 1995) has been a workhorse for our group for simulations of star formation (e.g. Price & Bate 2007, 2009; Price, Tricco, & Bate 2012; Lewis, Bate, & Price 2015) and accretion discs (e.g. Lodato & Rice 2004; Cossins, Lodato, & Clarke 2009), contains a rich array of physics necessary for star and planet formation including all of the above algorithms, but the legacy nature of the code makes it difficult to modify or debug and there are no plans to make it public.

GASOLINE (Wadsley, Stadel, & Quinn 2004) is another code that has been widely and successfully used for galaxy formation simulations, with its successor, GASOLINE 2 (Wadsley et al. 2017), recently publicly released. Hubber et al. (2011) have developed SEREN with similar goals to PHANTOM, focused on star cluster simulations. SEREN thus presents more advanced N-body algorithms compared to what is in PHANTOM but does not yet include magnetic fields, dust, or H_2 chemistry.

A third motivation was the need to distinguish between the 'high performance' code used for 3D simulations from simpler codes used to develop and test algorithms, such as our already-public NDSPMHD code (Price 2012a). PHANTOM is designed to 'take what works and make it fast', rather than containing options for every possible variation on the SPH algorithm. Obsolete options are actively deleted.

The initial release of PHANTOM has been developed with a focus on stellar, planetary, and Galactic astrophysics, as well as accretion discs. In this first paper, coinciding with the first stable public release, we describe and validate the core algorithms as well as some example applications. Various novel aspects and optimisation strategies are also presented. This paper is an attempt to document in detail what is currently available in the code, which include modules for MHD, dust–gas mixtures, self-gravity, and a range of other physics. The paper is also designed to serve as guide to the correct use of the various algorithms. Stable releases of PHANTOM are posted on the web[1], while the development version and wiki documentation are available on the BITBUCKET platform[2].

The paper is organised as follows: We describe the numerical methods in Section 2 with corresponding numerical tests in Section 5. We cover SPH basics (Section 2.1), our implementation of hydrodynamics (Sections 2.2 and 5.1), the timestepping algorithm (Section 2.3), external forces (Sections 2.4 and 5.2), turbulent forcing (Sections 2.5 and 6.1), accretion disc viscosity (Sections 2.6 and 5.3), Navier–Stokes viscosity (Sections 2.7 and 5.4), sink particles (Sections 2.8 and 5.5), stellar physics (Section 2.9), MHD (Sections 2.10 and 5.6), non-ideal MHD (Sections 2.11 and 5.7), self-gravity (Sections 2.12 and 5.8), dust–gas mixtures (Sections 2.13 and 5.9), ISM chemistry and cooling (Sections 2.14 and 5.10), and particle injection (Section 2.15). We present the algorithms for generating initial conditions in Section 3. Our approach to software engineering is described in Section 4. We give five examples of recent applications highlighting different aspects of PHANTOM in Section 6. We summarise in Section 7.

2 NUMERICAL METHOD

PHANTOM is based on the SPH technique, invented by Lucy (1977) and Gingold & Monaghan (1977) and the subject of numerous reviews (Benz 1990; Monaghan 1992, 2005, 2012; Rosswog 2009; Springel 2010; Price 2012a).

In the following, we adopt the convention that a, b, and c refer to particle indices; i, j, and k refer to vector or tensor indices; and n and m refer to indexing of nodes in the treecode.

2.1. Fundamentals

2.1.1. *Lagrangian hydrodynamics*

SPH solves the equations of hydrodynamics in Lagrangian form. The fluid is discretised onto a set of 'particles' of mass m that are moved with the local fluid velocity $\boldsymbol{v}$. Hence, the

[1] https://phantomsph.bitbucket.io/
[2] https://bitbucket.org/danielprice/phantom

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

two basic equations common to all physics in PHANTOM are

$$\frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} = \boldsymbol{v}, \tag{1}$$

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\rho(\nabla\cdot\boldsymbol{v}), \tag{2}$$

where $\boldsymbol{r}$ is the particle position and ρ is the density. These equations use the Lagrangian time derivative, $\mathrm{d}/\mathrm{d}t \equiv \partial/\partial t + \boldsymbol{v}\cdot\nabla$, and are the Lagrangian update of the particle position and the continuity equation (expressing the conservation of mass), respectively.

2.1.2. *Conservation of mass in SPH*

The density is computed in PHANTOM using the usual SPH density sum

$$\rho_a = \sum_b m_b W(|\boldsymbol{r}_a - \boldsymbol{r}_b|, h_a), \tag{3}$$

where a and b are particle labels, m is the mass of the particle, W is the smoothing kernel, h is the smoothing length, and the sum is over neighbouring particles (i.e. those within $R_{\rm kern}h$, where $R_{\rm kern}$ is the dimensionless cut-off radius of the smoothing kernel). Taking the Lagrangian time derivative of (3), one obtains the discrete form of (2) in SPH

$$\frac{\mathrm{d}\rho_a}{\mathrm{d}t} = \frac{1}{\Omega_a}\sum_b m_b(\boldsymbol{v}_a - \boldsymbol{v}_b)\cdot\nabla_a W_{ab}(h_a), \tag{4}$$

where $W_{ab}(h_a) \equiv W(|\boldsymbol{r}_a - \boldsymbol{r}_b|, h_a)$ and Ω_a is a term related to the gradient of the smoothing length (Springel & Hernquist 2002; Monaghan 2002) given by

$$\Omega_a \equiv 1 - \frac{\partial h_a}{\partial \rho_a}\sum_b m_b \frac{\partial W_{ab}(h_a)}{\partial h_a}. \tag{5}$$

Equation (4) is not used directly to compute the density in PHANTOM, since evaluating (3) provides a time-independent solution to (2) (see e.g. Monaghan 1992; Price 2012a for details). The time-dependent version (4) is equivalent to (3) up to a boundary term (see Price 2008) but is only used in PHANTOM to predict the smoothing length at the next timestep in order to reduce the number of iterations required to evaluate the density (see below).

Since (3)–(5) all depend on the kernel evaluated on neighbours within $R_{\rm kern}$ times h_a, all three of these summations may be computed simultaneously using a single loop over the same set of neighbours. Details of the neighbour finding procedure are given in Section 2.1.7.

2.1.3. *Setting the smoothing length*

The smoothing length itself is specified as a function of the particle number density, n, via

$$h_a = h_{\rm fact} n_a^{-1/3} = h_{\rm fact}\left(\frac{m_a}{\rho_a}\right)^{1/3}, \tag{6}$$

where $h_{\rm fact}$ is the proportionality factor specifying the smoothing length in terms of the mean local particle spacing and the second equality holds only for equal mass particles, which are enforced in PHANTOM. The restriction to equal mass particles means that the resolution strictly follows mass, which may be restrictive for problems involving large density contrasts (e.g. Hutchison et al. 2016). However, our view is that the potential pitfalls of unequal mass particles (see e.g. Monaghan & Price 2006) are currently too great to allow for a robust implementation in a public code.

Table 1. Compact support radii, variance, standard deviation, recommended ranges of $h_{\rm fact}$, and recommended default $h_{\rm fact}$ settings ($h^d_{\rm fact}$) for the kernel functions available in PHANTOM.

Kernel	$R_{\rm kern}$	σ^2/h^2	σ/h	$h_{\rm fact}$	$h^{\rm d}_{\rm fact}$	$N_{\rm neigh}$
M_4	2.0	9/10	0.95	1.0–1.2	1.2	57.9
M_5	2.5	23/20	1.07	1.0–1.2	1.2	113
M_6	3.0	7/5	1.18	1.0–1.1	1.0	113
C^2	2.0	4/5	0.89	$\geqslant$1.35	1.4	92
C^4	2.0	8/13	0.78	$\geqslant$1.55	1.6	137
C^6	2.0	1/2	0.71	$\geqslant$1.7	2.2	356

As described in Price (2012a), the proportionality constant $h_{\rm fact}$ can be related to the mean neighbour number according to

$$\overline{N}_{\rm neigh} = \frac{4}{3}\pi(R_{\rm kern}h_{\rm fact})^3, \tag{7}$$

however, this is only equal to the *actual* neighbour number for particles in a uniform density distribution (more specifically, for a density distribution with no second derivative), meaning that the actual neighbour number varies. The default setting for $h_{\rm fact}$ is 1.2, corresponding to an average of 57.9 neighbours for a kernel truncated at $2h$ (i.e. for $R_{\rm kern} = 2$) in three dimensions. Table 1 lists the settings recommended for different choices of kernel. The derivative required in (5) is given by

$$\frac{\partial h_a}{\partial \rho_a} = -\frac{3h_a}{\rho_a}. \tag{8}$$

2.1.4. *Iterations for h and ρ*

The mutual dependence of ρ and h means that a rootfinding procedure is necessary to solve both (3) and (6) simultaneously. The procedure implemented in PHANTOM follows Price & Monaghan (2004b, 2007), solving, for each particle, the equation

$$f(h_a) = \rho_{\rm sum}(h_a) - \rho(h_a) = 0, \tag{9}$$

where $\rho_{\rm sum}$ is the density computed from (3) and

$$\rho(h_a) = m_a(h_{\rm fact}/h_a)^3, \tag{10}$$

from (6). Equation (9) is solved with Newton–Raphson iterations:

$$h_{a,\rm new} = h_a - \frac{f(h_a)}{f'(h_a)}, \tag{11}$$

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

where the derivative is given by

$$f'(h_a) = \sum_b m_b \frac{\partial W_{ab}(h_a)}{\partial h_a} - \frac{\partial \rho_a}{\partial h_a} = -\frac{3\rho_a}{h_a}\Omega_a. \tag{12}$$

The iterations proceed until $|h_{a,\,\mathrm{new}} - h_a|/h_{a,\,0} < \varepsilon_h$, where $h_{a,\,0}$ is the smoothing length of particle a at the start of the iteration procedure and ε_h is the tolerance. The convergence with Newton–Raphson is fast, with a quadratic reduction in the error at each iteration, meaning that no more than 2–3 iterations are required even with a rapidly changing density field. We avoid further iterations by predicting the smoothing length from the previous timestep according to

$$h_a^0 = h_a + \Delta t \frac{\mathrm{d}h_a}{\mathrm{d}t} = h_a + \Delta t \frac{\partial h_a}{\partial \rho_a}\frac{\mathrm{d}\rho_a}{\mathrm{d}t}, \tag{13}$$

where $\mathrm{d}\rho_a/\mathrm{d}t$ is evaluated from (4).

Since h and ρ are mutually dependent, we store only the smoothing length, from which the density can be obtained at any time via a function call evaluating $\rho(h)$. The default value of ε_h is 10^{-4} so that h and ρ can be used interchangeably. Setting a small tolerance does not significantly change the computational cost, as the iterations quickly fall below a tolerance of 'one neighbour' according to (7), so any iterations beyond this refer to loops over the same set of neighbours which can be efficiently cached. However, it is important that the tolerance may be enforced to arbitrary precision rather than being an integer as implemented in the public version of GADGET, since (9) expresses a mathematical relationship between h and ρ that is assumed throughout the derivation of the SPH algorithm (see discussion in Price 2012a). The precision to which this is enforced places a lower limit on the total energy conservation. Fortunately, floating point neighbour numbers are now default in most GADGET-3 variants also.

2.1.5. *Kernel functions*

We write the kernel function in the form

$$W_{ab}(r,h) \equiv \frac{C_{\mathrm{norm}}}{h^3} f(q), \tag{14}$$

where C_{norm} is a normalisation constant, the factor of h^3 gives the dimensions of inverse volume, and $f(q)$ is a dimensionless function of $q \equiv |\boldsymbol{r}_a - \boldsymbol{r}_b|/h$. Various relations for kernels in this form are given in Morris (1996a) and in Appendix B of Price (2010). Those used in PHANTOM are the kernel gradient

$$\nabla_a W_{ab} = \hat{\boldsymbol{r}}_{ab} F_{ab}, \text{ where } F_{ab} \equiv \frac{C_{\mathrm{norm}}}{h^4} f'(q), \tag{15}$$

and the derivative of the kernel with respect to h,

$$\frac{\partial W_{ab}(r,h)}{\partial h} = -\frac{C_{\mathrm{norm}}}{h^4}\left[3f(q) + qf'(q)\right]. \tag{16}$$

Notice that the $\partial W/\partial h$ term in particular can be evaluated simply from the functions needed to compute the density and kernel gradient and hence does not need to be derived separately if a different kernel is used.

2.1.6. *Choice of smoothing kernel*

The default kernel function in SPH for the last 30 yr (since Monaghan & Lattanzio 1985) has been the M_4 cubic spline from the Schoenberg (1946) B-spline family, given by

$$f(q) = \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3, & 0 \le q < 1; \\ \frac{1}{4}(2-q)^3, & 1 \le q < 2; \\ 0. & q \ge 2, \end{cases} \tag{17}$$

where the normalisation constant $C_{\mathrm{norm}} = 1/\pi$ in 3D and the compact support of the function implies that $R_{\mathrm{kern}} = 2$. While the cubic spline kernel is satisfactory for many applications, it is not always the best choice. Most SPH kernels are based on approximating the Gaussian, but with compact support to avoid the $\mathcal{O}(N^2)$ computational cost. Convergence in SPH is guaranteed to be second order ($\propto h^2$) to the degree that the finite summations over neighbouring particles approximate integrals (e.g. Monaghan 1992, 2005; Price 2012a). Hence, the choice of kernel and the effect that a given kernel has on the particle distribution are important considerations.

In general, more accurate results will be obtained with a kernel with a larger compact support radius, since it will better approximate the Gaussian which has excellent convergence and stability properties (Morris 1996a; Price 2012a; Dehnen & Aly 2012). However, care is required. One should not simply increase h_{fact} for the cubic spline kernel because even though this implies more neighbours [via (7)], it increases the resolution length. For the B-splines, it also leads to the onset of the 'pairing instability' where the particle distribution becomes unstable to transverse modes, leading to particles forming close pairs (Thomas & Couchman 1992; Morris 1996a, 1996b; Børve, Omang, & Trulsen 2004; Price 2012a; Dehnen & Aly 2012). This is the motivation of our default choice of $h_{\mathrm{fact}} = 1.2$ for the cubic spline kernel, since it is just short of the maximum neighbour number that can be used while remaining stable to the pairing instability.

A better approach to reducing kernel bias is to keep the same resolution length[3] but to use a kernel that has a larger compact support radius. The traditional approach (e.g. Morris 1996a, 1996b; Børve et al. 2004; Price 2012a) has been to use the higher kernels in the B-spline series, i.e. the M_5 quartic

[3] This leads to the question of what is the appropriate definition of the 'smoothing length' to use when comparing kernels with different compact support radii. Recently, it has been shown convincingly by Dehnen & Aly (2012) and Violeau & Leroy (2014) that the resolution length in SPH is proportional to the standard deviation of W. Hence, the Gaussian has the same resolution length as the M_6 quintic with compact support radius of $3h$ with $h_{\mathrm{fact}} = 1.2$. Setting the number of neighbours, though related, is not a good way of specifying the resolution length.

which extends to $2.5h$

$$f(q) = \begin{cases} \left(\frac{5}{2} - q\right)^4 - 5\left(\frac{3}{2} - q\right)^4 & 0 \leq q < \frac{1}{2}, \\ +10\left(\frac{1}{2} - q\right)^4, & \\ \left(\frac{5}{2} - q\right)^4 - 5\left(\frac{3}{2} - q\right)^4, & \frac{1}{2} \leq q < \frac{3}{2}, \\ \left(\frac{5}{2} - q\right)^4, & \frac{3}{2} \leq q < \frac{5}{2}, \\ 0, & q \geq \frac{5}{2}, \end{cases} \tag{18}$$

where $C_{\rm norm} = 1/(20\pi)$, and the M_6 quintic extending to $3h$,

$$f(q) = \begin{cases} (3-q)^5 - 6(2-q)^5 + 15(1-q)^5, & 0 \leq q < 1, \\ (3-q)^5 - 6(2-q)^5, & 1 \leq q < 2, \\ (3-q)^5, & 2 \leq q < 3, \\ 0, & q \geq 3, \end{cases} \tag{19}$$

where $C_{\rm norm} = 1/(120\pi)$ in 3D. The quintic in particular gives results virtually indistinguishable from the Gaussian for most problems.

Recently, there has been tremendous interest in the use of the Wendland (1995) kernels, particularly since Dehnen & Aly (2012) showed that they are stable to the pairing instability at all neighbour numbers despite having a Gaussian-like shape and compact support. These functions are constructed as the unique polynomial functions with compact support but with a positive Fourier transform, which turns out to be a necessary condition for stability against the pairing instability (Dehnen & Aly 2012). The 3D Wendland kernels scaled to a radius of $2h$ are given by C^2:

$$f(q) = \begin{cases} \left(1 - \frac{q}{2}\right)^4 (2q + 1), & q < 2, \\ 0, & q \geq 2, \end{cases} \tag{20}$$

where $C_{\rm norm} = 21/(16\pi)$, the C^4 kernel

$$f(q) = \begin{cases} \left(1 - \frac{q}{2}\right)^6 \left(\frac{35q^2}{12} + 3q + 1\right), & q < 2, \\ 0, & q \geq 2, \end{cases} \tag{21}$$

where $C_{\rm norm} = 495/(256\pi)$, and the C^6 kernel

$$f(q) = \begin{cases} \left(1 - \frac{q}{2}\right)^8 \left(4q^3 + \frac{25q^2}{4} + 4q + 1\right), & q < 2, \\ 0, & q \geq 2, \end{cases} \tag{22}$$

where $C_{\rm norm} = 1365/(512\pi)$. Figure 1 graphs $f(q)$ and its first and second derivative for each of the kernels available in Phantom.

Several authors have argued for use of the Wendland kernels by default. For example, Rosswog (2015) found best results on simple test problems using the C^6 Wendland kernel. However, 'best' in that case implied using an average of 356 neighbours in 3D (i.e. $h_{\rm fact} = 2.2$ with $R_{\rm kern} = 2.0$) which is a factor of 6 more expensive than the standard approach. Similarly, Hu et al. (2014) recommend the C^4 kernel with 200 neighbours which is 3.5 times more expensive. The large number of neighbours are needed because the Wendland kernels are always worse than the B-splines for a given number of neighbours due to the positive Fourier transform, meaning that the kernel bias (related to the Fourier transform) is always positive where the B-spline errors oscillate around zero (Dehnen & Aly 2012). Hence, whether or not this additional cost is worthwhile depends on the application. A more comprehensive analysis would be valuable here, as the 'best' choice of kernel remains an open question (see also the kernels proposed by Cabezón, García-Senz, & Relaño 2008; García-Senz et al. 2014). An even broader question regards the kernel used for dissipation terms, for gravitational force softening and for drag in two-fluid applications (discussed further in Section 2.13). For example, Laibe & Price (2012a) found that double-hump-shaped kernels led to more than an order of magnitude improvement in accuracy when used for drag terms.

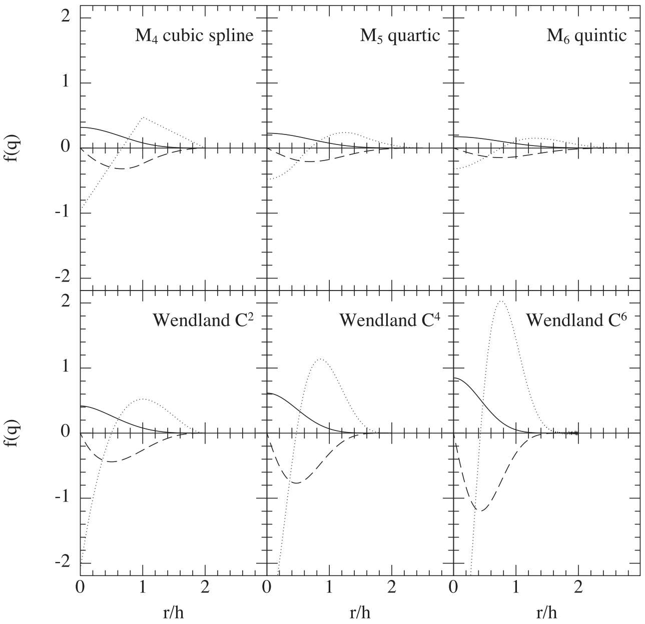

Figure 1. Smoothing kernels available in Phantom (solid lines) together with their first (dashed lines) and second (dotted lines) derivatives. Wendland kernels in Phantom (bottom row) are given compact support radii of 2, whereas the B-spline kernels (top row) adopt the traditional practice where the support radius increases by 0.5. Thus, use of alternative kernels requires adjustment of $h_{\rm fact}$, the ratio of smoothing length to particle spacing (see Table 1).

A simple and practical approach to checking that kernel bias does not affect the solution that we have used and advocate when using Phantom is to first attempt a simulation with the cubic spline, but then to check the results with a low resolution calculation using the quintic kernel. If the results are identical, then it indicates that the kernel bias is not important, but if not, then use of smoother but costlier kernels such as M_6 or C^6 may be warranted. Wendland kernels are mainly

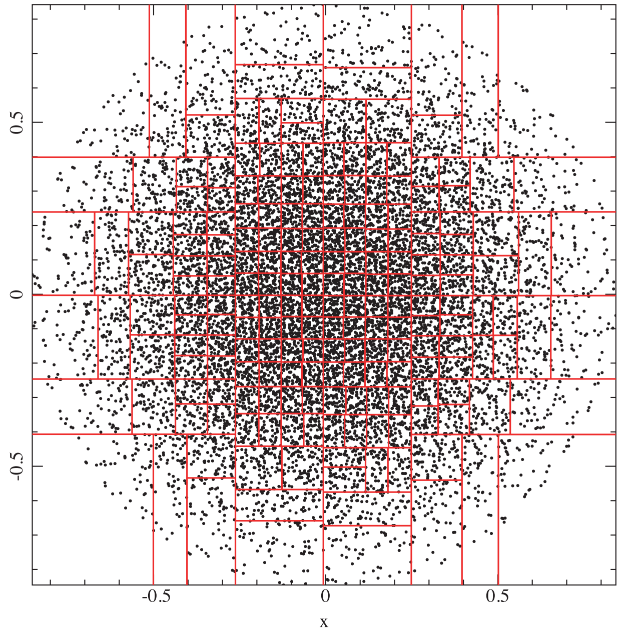

Figure 2. Example of the *k*d-tree build. For illustrative purposes only, we have constructed a 2D version of the tree on the projected particle distribution in the *x*–*y* plane of the particle distribution from a polytrope test with 13 115 particles. Each level of the tree recursively splits the particle distribution in half, bisecting the longest axis at the centre of mass until the number of particles in a given cell is $<N_{\rm min}$. For clarity, we have used $N_{\rm min} = 100$ in the above example, while $N_{\rm min} = 10$ by default.

useful for preventing the pairing instability and are necessary if one desires to employ a large number of neighbours.

2.1.7. *Neighbour finding*

Finding neighbours is the main computational expense to any SPH code. Earlier versions of PHANTOM contained three different options for neighbour-finding: a Cartesian grid, a cylindrical grid, and a *k*d-tree. This was because we wrote the code originally with non-self-gravitating problems in mind, for which the overhead associated with a treecode is unnecessary. Since the implementation of self-gravity in PHANTOM the *k*d-tree has become the default, and is now sufficiently well optimised that the fixed-grid modules are more efficient only for simulations that do not employ either self-gravity or individual particle timesteps, which are rare in astrophysics.

A key optimisation strategy employed in PHANTOM is to perform the neighbour search for groups of particles. The results of this search (i.e. positions of all trial neighbours) are then cached and used to check for neighbours for individual particles in the group. Our *k*d-tree algorithm closely follows Gafton & Rosswog (2011), splitting the particles recursively based on the centre of mass and bisecting the longest axis at each level (Figure 2). The tree build is refined until a cell contains less than $N_{\rm min}$ particles, which is then referred to as a 'leaf node'. By default, $N_{\rm min} = 10$. The neighbour search is then performed once for each leaf node. Further details are given in Appendix A.3.1.

2.2. Hydrodynamics

2.2.1. *Compressible hydrodynamics*

The equations of compressible hydrodynamics are solved in the form

$$\frac{{\rm d}\boldsymbol{v}}{{\rm d}t} = -\frac{\nabla P}{\rho} + \Pi_{\rm shock} + \boldsymbol{a}_{\rm ext}(\boldsymbol{r}, t) + \boldsymbol{a}_{\rm sink-gas} + \boldsymbol{a}_{\rm selfgrav}, \tag{23}$$

$$\frac{{\rm d}u}{{\rm d}t} = -\frac{P}{\rho}(\nabla\cdot\boldsymbol{v}) + \Lambda_{\rm shock} - \frac{\Lambda_{\rm cool}}{\rho}, \tag{24}$$

where P is the pressure, u is the specific internal energy, $\boldsymbol{a}_{\rm ext}$, $\boldsymbol{a}_{\rm sink-gas}$ and $\boldsymbol{a}_{\rm selfgrav}$ refer to (optional) accelerations from 'external' or 'body' forces (Section 2.4), sink particles (Section 2.8), and self-gravity (Section 2.12), respectively. $\Pi_{\rm shock}$ and $\Lambda_{\rm shock}$ are dissipation terms required to give the correct entropy increase at a shock front, and $\Lambda_{\rm cool}$ is a cooling term.

2.2.2. *Equation of state*

The equation set is closed by an equation of state (EOS) relating the pressure to the density and/or internal energy. For an ideal gas, this is given by

$$P = (\gamma - 1)\rho u, \tag{25}$$

where γ is the adiabatic index and the sound speed $c_{\rm s}$ is given by

$$c_{\rm s} = \sqrt{\frac{\gamma P}{\rho}}. \tag{26}$$

The internal energy, u, can be related to the gas temperature, T, using

$$P = \frac{\rho k_{\rm B} T}{\mu m_{\rm H}}, \tag{27}$$

giving

$$T = \frac{\mu m_{\rm H}}{k_{\rm B}}(\gamma - 1)u, \tag{28}$$

where $k_{\rm B}$ is Boltzmann's constant, μ is the mean molecular weight, and $m_{\rm H}$ is the mass of a hydrogen atom. Thus, to infer the temperature, one needs to specify a composition, but only the internal energy affects the gas dynamics. Equation (25) with $\gamma = 5/3$ is the default EOS in PHANTOM.

In the case where shocks are assumed to radiate away all of the heat generated at the shock front (i.e. $\Lambda_{\rm shock} = 0$) and there is no cooling ($\Lambda_{\rm cool} = 0$), (24) becomes simply, using (2)

$$\frac{{\rm d}u}{{\rm d}t} = \frac{P}{\rho^2}\frac{{\rm d}\rho}{{\rm d}t}, \tag{29}$$

which, using (25) can be integrated to give

$$P = K\rho^{\gamma}, \tag{30}$$

where K is the polytropic constant. Even more simply, in the case where the temperature is assumed constant, or prescribed

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

as a function of position, the EOS is simply

$$P = c_s^2 \rho. \tag{31}$$

In both of these cases, (30) and (31), the internal energy does not need to be stored. In this case, the temperature is effectively set by the value of K (and the density if $\gamma \neq 1$). Specifically,

$$T = \frac{\mu m_{\rm H}}{k_{\rm B}} K \rho^{\gamma - 1}. \tag{32}$$

2.2.3. Code units

For pure hydrodynamics, physical units are irrelevant to the numerical results since (1)–(2) and (23)–(24) are scale free to all but the Mach number. Hence, setting physical units is only useful when comparing simulations with Nature, when physical heating or cooling rates are applied via (24), or when one wishes to interpret the results in terms of temperature using (28) or (32).

In the case where gravitational forces are applied, either using an external force (Section 2.4) or using self-gravity (Section 2.12), we adopt the standard procedure of transforming units such that $G = 1$ in code units, i.e.

$$u_{\rm time} = \sqrt{\frac{u_{\rm dist}^3}{G u_{\rm mass}}}, \tag{33}$$

where $u_{\rm time}$, $u_{\rm dist}$, and $u_{\rm mass}$ are the units of time, length, and mass, respectively. Additional constraints apply when using relativistic terms (Section 2.4.5) or magnetic fields (Section 2.10.3).

2.2.4. Equation of motion in SPH

We follow the variable smoothing length formulation described by Price (2012a), Price & Federrath (2010), and Lodato & Price (2010). We discretise (23) using

$$\frac{{\rm d}\boldsymbol{v}_a}{{\rm d}t} = -\sum_b m_b \left[\frac{P_a + q_{ab}^a}{\rho_a^2 \Omega_a} \nabla_a W_{ab}(h_a) + \frac{P_b + q_{ab}^b}{\rho_b^2 \Omega_b} \nabla_a W_{ab}(h_b) \right] + \boldsymbol{a}_{\rm ext}(\boldsymbol{r}_a, t) + \boldsymbol{a}_{\rm sink-gas}^a + \boldsymbol{a}_{\rm selfgrav}^a, \tag{34}$$

where the q_{ab}^a and q_{ab}^b terms represent the artificial viscosity (discussed in Section 2.2.7).

2.2.5. Internal energy equation

The internal energy equation (24) is discretised using the time derivative of the density sum (c.f. 29), which from (4) gives

$$\frac{{\rm d}u_a}{{\rm d}t} = \frac{P_a}{\rho_a^2 \Omega_a} \sum_b m_b \boldsymbol{v}_{ab} \cdot \nabla_a W_{ab}(h_a) + \Lambda_{\rm shock} - \frac{\Lambda_{\rm cool}}{\rho}, \tag{35}$$

where $\boldsymbol{v}_{ab} \equiv \boldsymbol{v}_a - \boldsymbol{v}_b$. In the variational formulation of SPH (e.g. Price 2012a), this expression is used as a constraint to derive (34), which guarantees both the conservation of energy and entropy (the latter in the absence of dissipation terms). The shock capturing terms in the internal energy equation are discussed below.

By default, we assume an adiabatic gas, meaning that $P{\rm d}V$ work and shock heating terms contribute to the thermal energy of the gas, no energy is radiated to the environment, and total energy is conserved. To approximate a radiative gas, one may set one or both of these terms to zero. Neglecting the shock heating term, $\Lambda_{\rm shock}$, gives an approximation equivalent to a polytropic EOS (30), as described in Section 2.2.2. Setting both shock and work contributions to zero implies that ${\rm d}u/{\rm d}t = 0$, meaning that each particle will simply retain its initial temperature.

2.2.6. Conservation of energy in SPH

Does evolving the internal energy equation imply that total energy is not conserved? Wrong! Total energy in SPH, for the case of hydrodynamics, is given by

$$E = \sum_a m_a \left(\frac{1}{2} v_a^2 + u_a \right). \tag{36}$$

Taking the (Lagrangian) time derivative, we find that conservation of energy corresponds to

$$\frac{{\rm d}E}{{\rm d}t} = \sum_a m_a \left(\boldsymbol{v}_a \cdot \frac{{\rm d}\boldsymbol{v}_a}{{\rm d}t} + \frac{{\rm d}u_a}{{\rm d}t} \right) = 0. \tag{37}$$

Inserting our expressions (34) and (35), and neglecting for the moment dissipative terms and external forces, we find

$$\frac{{\rm d}E}{{\rm d}t} = -\sum_a \sum_b m_a m_b \left[\frac{P_a \boldsymbol{v}_b}{\rho_a^2 \Omega_a} \cdot \nabla_a W_{ab}(h_a) + \frac{P_b \boldsymbol{v}_a}{\rho_b^2 \Omega_b} \cdot \nabla_a W_{ab}(h_b) \right] = 0. \tag{38}$$

The double summation on the right-hand side equals zero because the kernel gradient, and hence the overall sum, is antisymmetric. That is, $\nabla_a W_{ab} = -\nabla_b W_{ba}$. This means one can relabel the summation indices arbitrarily in one half of the sum, and add it to one half of the original sum to give zero. One may straightforwardly verify that this remains true when one includes the dissipative terms (see below).

This means that even though we employ the internal energy equation, total energy remains conserved to machine precision in the spatial discretisation. That is, energy is conserved irrespective of the number of particles, the number of neighbours or the choice of smoothing kernel. The only non-conservation of energy arises from the ordinary differential equation solver one employs to solve the left-hand side of the equations. We thus employ a symplectic time integration scheme in order to preserve the conservation properties as accurately as possible (Section 2.3.1).

2.2.7. Shock capturing: momentum equation

The shock capturing dissipation terms are implemented following Monaghan (1997), derived by analogy with Riemann solvers from the special relativistic dissipation terms proposed by Chow & Monaghan (1997). These were extended by Price & Monaghan (2004b, 2005) to MHD and recently to dust–gas mixtures by Laibe & Price (2014b). In a recent paper, Puri & Ramachandran (2014) found this approach, along with the Morris & Monaghan (1997) switch (which they referred to as the 'Monaghan–Price–Morris' formulation) to be

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

the most accurate and robust method for shock capturing in SPH when compared to several other approaches, including Godunov SPH (e.g. Inutsuka 2002; Cha & Whitworth 2003).

The formulation in PHANTOM differs from that given in Price (2012a) only by the way that the density and signal speed in the q terms are averaged, as described in Price & Federrath (2010) and Lodato & Price (2010). That is, we use

$$\Pi^a_{\rm shock} \equiv -\sum_b m_b \left[\frac{q^a_{ab}}{\rho_a^2 \Omega_a}\nabla_a W_{ab}(h_a) + \frac{q^b_{ab}}{\rho_b^2\Omega_b}\nabla_a W_{ab}(h_b)\right], \tag{39}$$

where

$$q^a_{ab} = \begin{cases} -\frac{1}{2}\rho_a v_{{\rm sig},a}\boldsymbol{v}_{ab}\cdot\hat{\boldsymbol{r}}_{ab}, & \boldsymbol{v}_{ab}\cdot\hat{\boldsymbol{r}}_{ab} < 0 \\ 0 & \text{otherwise,}\end{cases} \tag{40}$$

where $\boldsymbol{v}_{ab} \equiv \boldsymbol{v}_a - \boldsymbol{v}_b$, $\hat{\boldsymbol{r}}_{ab} \equiv (\boldsymbol{r}_a - \boldsymbol{r}_b)/|\boldsymbol{r}_a - \boldsymbol{r}_b|$ is the unit vector along the line of sight between the particles, and $v_{\rm sig}$ is the maximum signal speed, which depends on the physics implemented. For hydrodynamics, this is given by

$$v_{{\rm sig},a} = \alpha^{\rm AV}_a c_{{\rm s},a} + \beta^{\rm AV}|\boldsymbol{v}_{ab}\cdot\hat{\boldsymbol{r}}_{ab}|, \tag{41}$$

where in general $\alpha^{\rm AV}_a \in [0, 1]$ is controlled by a switch (see Section 2.2.9), while $\beta^{\rm AV} = 2$ by default.

Importantly, α does *not* multiply the $\beta^{\rm AV}$ term. The $\beta^{\rm AV}$ term provides a second-order Von Neumann & Richtmyer-like term that prevents particle interpenetration (e.g. Lattanzio et al. 1986; Monaghan 1989), and thus $\beta^{\rm AV} \geqslant 2$ is needed wherever particle penetration may occur. This is important in accretion disc simulations where use of a low α may be acceptable in the absence of strong shocks, but a low β will lead to particle penetration of the disc midplane, which is the cause of a number of convergence issues (Meru & Bate 2011, 2012). Price & Federrath (2010) found that $\beta^{\rm AV} = 4$ was necessary at high Mach number ($M \gtrsim 5$) to maintain a sharp shock structure, which despite nominally increasing the viscosity was found to give less dissipation overall because particle penetration no longer occurred at shock fronts.

2.2.8. Shock capturing: internal energy equation

The key insight from Chow & Monaghan (1997) was that shock capturing not only involves a viscosity term but involves dissipating the jump in each component of the energy, implying a conductivity term in hydrodynamics and resistive dissipation in MHD (see Section 2.10.5). The resulting contribution to the internal energy equation is given by (e.g. Price 2012a)

$$\begin{aligned}\Lambda_{\rm shock} \equiv & -\frac{1}{\Omega_a\rho_a}\sum_b m_b v_{{\rm sig},a}\frac{1}{2}(\boldsymbol{v}_{ab}\cdot\hat{\boldsymbol{r}}_{ab})^2 F_{ab}(h_a) \\ & + \sum_b m_b \alpha_u v^u_{\rm sig}(u_a - u_b)\frac{1}{2}\left[\frac{F_{ab}(h_a)}{\Omega_a\rho_a} + \frac{F_{ab}(h_b)}{\Omega_b\rho_b}\right] \\ & + \Lambda_{\rm artres},\end{aligned} \tag{42}$$

where the first term provides the viscous shock heating, the second term provides an artificial thermal conductivity, F_{ab} is defined as in (15), and $\Lambda_{\rm artres}$ is the heating due to artificial resistivity [Equation (182)]. The signal speed we use for conductivity term differs from the one used for viscosity, as discussed by Price (2008, 2012a). In PHANTOM, we use

$$v^u_{\rm sig} = \sqrt{\frac{|P_a - P_b|}{\overline{\rho}_{ab}}} \tag{43}$$

for simulations that do not involve self-gravity or external body forces (Price 2008), and

$$v^u_{\rm sig} = |\boldsymbol{v}_{ab}\cdot\hat{\boldsymbol{r}}_{ab}| \tag{44}$$

for simulations that do (Wadsley, Veeravalli, & Couchman 2008). The importance of the conductivity term for treating contact discontinuities was highlighted by Price (2008), explaining the poor results found by Agertz et al. (2007) in SPH simulations of Kelvin–Helmholtz instabilities run across contact discontinuities (discussed further in Section 5.1.4). With (44), we have found there is no need for further switches to reduce conductivity (e.g. Price 2004; Price & Monaghan 2005; Valdarnini 2016), since the effective thermal conductivity κ is second order in the smoothing length ($\propto h^2$). PHANTOM therefore uses $\alpha_u = 1$ by default in (42) and we have not yet found a situation where this leads to excess smoothing.

It may be readily shown that the total energy remains conserved in the presence of dissipation by combining (42) with the corresponding dissipative terms in (34). The contribution to the entropy from both viscosity and conductivity is also positive definite (see the appendix in Price & Monaghan 2004b for the mathematical proof in the case of conductivity).

2.2.9. Shock detection

The standard approach to reducing dissipation in SPH away from shocks for the last 15 yr has been the switch proposed by Morris & Monaghan (1997), where the dimensionless viscosity parameter α is evolved for each particle a according to

$$\frac{{\rm d}\alpha_a}{{\rm d}t} = \max(-\nabla\cdot\boldsymbol{v}_a, 0) - \frac{(\alpha_a - \alpha_{\rm min})}{\tau_a}, \tag{45}$$

where $\tau \equiv h/(\sigma_{\rm decay}v_{\rm sig})$ and $\sigma_{\rm decay} = 0.1$ by default. We set $v_{\rm sig}$ in the decay time equal to the sound speed to avoid the need to store ${\rm d}\alpha/{\rm d}t$, since $\nabla\cdot\boldsymbol{v}$ is already stored in order to compute (4). This is the switch used for numerous turbulence applications with PHANTOM (e.g. Price & Federrath 2010; Price et al. 2011; Tricco et al. 2016b) where it is important to minimise numerical dissipation in order to maximise the Reynolds number (e.g. Valdarnini 2011; Price 2012b).

More recently, Cullen & Dehnen (2010) proposed a more advanced switch using the time derivative of the velocity divergence. A modified version based on the gradient of the velocity divergence was also proposed by Read & Hayfield (2012). We implement a variation on the Cullen & Dehnen (2010) switch, using a shock indicator of the form

$$A_a = \xi_a \max\left[-\frac{\rm d}{{\rm d}t}(\nabla\cdot\boldsymbol{v}_a), 0\right], \tag{46}$$

where

$$\xi = \frac{|\nabla \cdot \boldsymbol{v}|^2}{|\nabla \cdot \boldsymbol{v}|^2 + |\nabla \times \boldsymbol{v}|^2} \tag{47}$$

is a modification of the Balsara (1995) viscosity limiter for shear flows. We use this to set α according to

$$\alpha_{\mathrm{loc},a} = \min\left(\frac{10h_a^2 A_a}{c_{\mathrm{s},a}^2}, \alpha_{\max}\right), \tag{48}$$

where c_{s} is the sound speed and $\alpha_{\max} = 1$. We use c_{s} in the expression for α_{loc} also for MHD (Section 2.10) since we found using the magnetosonic speed led to a poor treatment of MHD shocks. If $\alpha_{\mathrm{loc},\,a} > \alpha_a$, we set $\alpha_a = \alpha_{\mathrm{loc},\,a}$, otherwise we evolve α_a according to

$$\frac{\mathrm{d}\alpha_a}{\mathrm{d}t} = -\frac{(\alpha_a - \alpha_{\mathrm{loc},a})}{\tau_a}, \tag{49}$$

where τ is defined as in the Morris & Monaghan (1997) version, above. We evolve α in the predictor part of the integrator only, i.e. with a first-order time integration, to avoid complications in the corrector step. However, we perform the predictor step implicitly using a backward Euler method, i.e.

$$\alpha_a^{n+1} = \frac{\alpha_a^n + \alpha_{\mathrm{loc},a}\Delta t/\tau_a}{1 + \Delta t/\tau_a}, \tag{50}$$

which ensures that the decay is stable regardless of the timestep (we do this for the Morris & Monaghan method also).

We use the method outlined in Appendix B3 of Cullen & Dehnen (2010) to compute $\mathrm{d}(\nabla \cdot \boldsymbol{v}_a)/\mathrm{d}t$. That is, we first compute the gradient tensors of the velocity, $\boldsymbol{v}$, and acceleration, $\boldsymbol{a}$ (used from the previous timestep), during the density loop using an SPH derivative operator that is exact to linear order, that is, with the matrix correction outlined in Price (2004, 2012a), namely

$$R_a^{ij}\frac{\partial v_a^k}{\partial x_a^j} = \sum_b m_b \left(v_b^k - v_a^k\right)\frac{\partial W_{ab}(h_a)}{\partial x^i}, \tag{51}$$

where

$$R_a^{ij} = \sum_b m_b \left(x_b^i - x_a^i\right)\frac{\partial W_{ab}(h_a)}{\partial x^j} \approx \delta^{ij}, \tag{52}$$

and repeated tensor indices imply summation. Finally, we construct the time derivative of the velocity divergence according to

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial v_a^i}{\partial x_a^i}\right) = \frac{\partial a_a^i}{\partial x_a^i} - \frac{\partial v_a^i}{\partial x_a^j}\frac{\partial v_a^j}{\partial x_a^i}, \tag{53}$$

where, as previously, repeated i and j indices imply summation. In Cartesian coordinates, the last term in (53) can be written out explicitly using

$$\frac{\partial v_a^i}{\partial x_a^j}\frac{\partial v_a^j}{\partial x_a^i} = \left(\frac{\partial v^x}{\partial x}\right)^2 + \left(\frac{\partial v^y}{\partial y}\right)^2 + \left(\frac{\partial v^z}{\partial z}\right)^2 + 2\left[\frac{\partial v^x}{\partial y}\frac{\partial v^y}{\partial x} + \frac{\partial v^x}{\partial z}\frac{\partial v^z}{\partial x} + \frac{\partial v^z}{\partial y}\frac{\partial v^y}{\partial z}\right]. \tag{54}$$

2.2.10. *Cooling*

The cooling term Λ_{cool} can be set either from detailed chemical calculations (Section 2.14.1) or, for discs, by the simple 'β-cooling' prescription of Gammie (2001), namely

$$\Lambda_{\mathrm{cool}} = \frac{\rho u}{t_{\mathrm{cool}}}, \tag{55}$$

where

$$t_{\mathrm{cool}} \equiv \frac{\Omega(R)}{\beta_{\mathrm{cool}}}, \tag{56}$$

with β_{cool} an input parameter to the code specifying the cooling timescale in terms of the local orbital time. We compute Ω in (56) using $\Omega \equiv 1/(x^2 + y^2 + z^2)^{3/2}$, i.e. assuming Keplerian rotation around a central object with mass equal to unity, with $G = 1$ in code units.

2.2.11. *Conservation of linear and angular momentum*

The total linear momentum is given by

$$\boldsymbol{P} = \sum_a m_a \boldsymbol{v}_a, \tag{57}$$

such that conservation of momentum corresponds to

$$\frac{\mathrm{d}\boldsymbol{P}}{\mathrm{d}t} = \sum_a m_a \frac{\mathrm{d}\boldsymbol{v}_a}{\mathrm{d}t} = 0. \tag{58}$$

Inserting our discrete equation (34), we find

$$\frac{\mathrm{d}\boldsymbol{P}}{\mathrm{d}t} = \sum_a \sum_b m_a m_b \left[\frac{P_a + q_{ab}^a}{\rho_a^2 \Omega_a}\nabla_a W_{ab}(h_a) + \frac{P_b + q_{ab}^b}{\rho_b^2 \Omega_b}\nabla_a W_{ab}(h_b)\right] = 0, \tag{59}$$

where, as for the total energy (Section 2.2.6), the double summation is zero because of the antisymmetry of the kernel gradient. The same argument applies to the conservation of angular momentum,

$$\sum_a m_a \boldsymbol{r}_a \times \boldsymbol{v}_a \tag{60}$$

(see e.g. Price 2012a for a detailed proof). As with total energy, this means linear and angular momentum are exactly conserved by our SPH scheme to the accuracy with which they are conserved by the timestepping scheme.

In PHANTOM, linear and angular momentum are both conserved to round-off error (typically $\sim 10^{-16}$ in double precision) with global timestepping, but exact conservation is violated when using individual particle timesteps or when using the kd-tree to compute gravitational forces. The magnitude of these quantities, as well as the total energy and the individual components of energy (kinetic, internal, potential, and magnetic), should thus be monitored by the user at runtime. Typically with individual timesteps, one should expect energy conservation to $\Delta E/E \sim 10^{-3}$ and linear and angular momentum conservation to $\sim 10^{-6}$ with default code settings. The code execution is aborted if conservation errors exceed 10%.

2.3. Time integration

2.3.1. *Timestepping algorithm*

We integrate the equations of motion using a generalisation of the Leapfrog integrator which is reversible in the case of both velocity dependent forces and derivatives which depend on the velocity field. The basic integrator is the Leapfrog method in 'Kick–Drift–Kick' or 'Velocity Verlet' form (Verlet 1967), where the positions and velocities of particles are updated from time t^n to t^{n+1} according to

$$\boldsymbol{v}^{n+\frac{1}{2}} = \boldsymbol{v}^n + \frac{1}{2}\Delta t \boldsymbol{a}^n, \tag{61}$$

$$\boldsymbol{r}^{n+1} = \boldsymbol{r}^n + \Delta t \boldsymbol{v}^{n+\frac{1}{2}}, \tag{62}$$

$$\boldsymbol{a}^{n+1} = \boldsymbol{a}(\boldsymbol{r}^{n+1}), \tag{63}$$

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^{n+\frac{1}{2}} + \frac{1}{2}\Delta t \boldsymbol{a}^{n+1}, \tag{64}$$

where $\Delta t \equiv t^{n+1} - t^n$. This is identical to the formulation of Leapfrog used in other astrophysical SPH codes (e.g. Springel 2005; Wadsley et al. 2004). The Verlet scheme, being both reversible and symplectic (e.g. Hairer, Lubich, & Wanner 2003), preserves the Hamiltonian nature of the SPH algorithm (e.g. Gingold & Monaghan 1982b; Monaghan & Price 2001). In particular, both linear and angular momentum are exactly conserved, there is no long-term energy drift, and phase space volume is conserved (e.g. for orbital dynamics). In SPH, this is complicated by velocity-dependent terms in the acceleration from the shock-capturing dissipation terms. In this case, the corrector step, (64), becomes implicit. The approach we take is to notice that these terms are not usually dominant over the position-dependent terms. Hence, we use a first-order prediction of the velocity, as follows:

$$\boldsymbol{v}^{n+\frac{1}{2}} = \boldsymbol{v}^n + \frac{1}{2}\Delta t \boldsymbol{a}^n, \tag{65}$$

$$\boldsymbol{r}^{n+1} = \boldsymbol{r}^n + \Delta t \boldsymbol{v}^{n+\frac{1}{2}}, \tag{66}$$

$$\boldsymbol{v}^{*} = \boldsymbol{v}^{n+\frac{1}{2}} + \frac{1}{2}\Delta t \boldsymbol{a}^{n}, \tag{67}$$

$$\boldsymbol{a}^{n+1} = \boldsymbol{a}(\boldsymbol{r}^{n+1}, \boldsymbol{v}^*), \tag{68}$$

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^* + \frac{1}{2}\Delta t \left[\boldsymbol{a}^{n+1} - \boldsymbol{a}^n\right]. \tag{69}$$

At the end of the step, we then check if the error in the first-order prediction is less than some tolerance ε according to

$$e = \frac{|\boldsymbol{v}^{n+1} - \boldsymbol{v}^*|}{|\boldsymbol{v}^{\rm mag}|} < \epsilon_{\rm v}, \tag{70}$$

where $\boldsymbol{v}^{\rm mag}$ is the mean velocity on all SPH particles (we set the error to zero if $|\boldsymbol{v}^{\rm mag}| = 0$) and by default $\varepsilon_{\rm v} = 10^{-2}$. If this criterion is violated, then we recompute the accelerations by replacing $\boldsymbol{v}^*$ with $\boldsymbol{v}^{n+1}$ and iterating (68) and (69) until the criterion in (70) is satisfied. In practice, this happens rarely, but occurs for example in the first few steps of the Sedov problem where the initial conditions are discontinuous (Section 5.1.3). As each iteration is as expensive as halving the timestep, we also constrain the subsequent timestep such that iterations should not occur, i.e.

$$\Delta t = \min\left(\Delta t, \frac{\Delta t}{\sqrt{e_{\rm max}/\epsilon}}\right), \tag{71}$$

where $e_{\rm max} = \max(e)$ over all particles. A caveat to the above is that velocity iterations are not currently implemented when using individual particle timesteps.

Additional variables such as the internal energy, u, and the magnetic field, $\boldsymbol{B}$, are timestepped with a predictor and trapezoidal corrector step in the same manner as the velocity, following (65), (67) and (69).

Velocity-dependent external forces are treated separately, as described in Section 2.4.

2.3.2. *Timestep constraints*

The timestep itself is determined at the end of each step, and is constrained to be less than the maximum stable timestep. For a given particle, a, this is given by (e.g. Lattanzio et al. 1986; Monaghan 1997)

$$\Delta t_{{\rm C},a} \equiv C_{\rm cour}\frac{h_a}{v^{\rm dt}_{{\rm sig},a}}, \tag{72}$$

where $C_{\rm cour} = 0.3$ by default (Lattanzio et al. 1986) and $v^{\rm dt}_{\rm sig}$ is taken as the maximum of (41) over the particle's neighbours assuming $\alpha^{\rm AV} = \max(\alpha^{\rm AV}, 1)$. The criterion above differs from the usual Courant–Friedrichs–Lewy condition used in Eulerian codes (Courant, Friedrichs, & Lewy 1928) because it depends only on the difference in velocity between neighbouring particles, not the absolute value.

An additional constraint is applied from the accelerations (the 'force condition'), where

$$\Delta t_{{\rm f},a} \equiv C_{\rm force}\sqrt{\frac{h_a}{|\boldsymbol{a}_a|}}, \tag{73}$$

where $C_{\rm force} = 0.25$ by default. A separate timestep constraint is applied for external forces

$$\Delta t_{{\rm ext},a} \equiv C_{\rm force}\sqrt{\frac{h}{|\boldsymbol{a}_{{\rm ext},a}|}}, \tag{74}$$

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

and for accelerations to SPH particles to/from sink particles (Section 2.8)

$$\Delta t_{\mathrm{sink-gas},a} \equiv C_{\mathrm{force}} \sqrt{\frac{h_a}{|\boldsymbol{a}_{\mathrm{sink-gas},a}|}}. \tag{75}$$

For external forces with potentials defined such that $\Phi \to 0$ as $r \to \infty$, an additional constraint is applied using (Dehnen & Read 2011)

$$\Delta t_{\Phi,a} \equiv C_{\mathrm{force}} \eta_{\Phi} \sqrt{\frac{|\Phi_a|}{|\nabla \Phi|_a^2}}, \tag{76}$$

where $\eta_{\Phi} = 0.05$ (see Section 2.8.5).

The timestep for particle a is then taken to be the minimum of all of the above constraints, i.e.

$$\Delta t_a = \min\left(\Delta t_{\mathrm{C}}, \Delta t_{\mathrm{f}}, \Delta t_{\mathrm{ext}}, \Delta t_{\mathrm{sink-gas}}, \Delta t_{\Phi}\right)_a, \tag{77}$$

with possible other constraints arising from additional physics as described in their respective sections. With global timestepping, the resulting timestep is the minimum over all particles

$$\Delta t = \min_a (\Delta t_a). \tag{78}$$

2.3.3. *Substepping of external forces*

In the case where the timestep is dominated by any of the external force timesteps, i.e. (74)–(76), we implement an operator splitting approach implemented according to the reversible reference system propagator algorithm (RESPA) derived by Tuckerman, Berne, & Martyna (1992) for molecular dynamics. RESPA splits the acceleration into 'long-range' and 'short-range' contributions, which in PHANTOM are defined to be the SPH and external/point-mass accelerations, respectively.

Our implementation follows Tuckerman et al. (1992) (see their Appendix B), where the velocity is first predicted to the half step using the 'long-range' forces, followed by an inner loop where the positions are updated with the current velocity and the velocities are updated with the 'short-range' accelerations. Thus, the timestepping proceeds according to

$$\boldsymbol{v} = \boldsymbol{v} + \frac{\Delta t_{\mathrm{sph}}}{2} \boldsymbol{a}_{\mathrm{sph}}^n, \tag{79}$$

$$\text{over substeps} \left\{ \begin{array}{ll} \boldsymbol{v} = \boldsymbol{v} + \dfrac{\Delta t_{\mathrm{ext}}}{2} \boldsymbol{a}_{\mathrm{ext}}^m, & (80) \\ \boldsymbol{r} = \boldsymbol{r} + \Delta t_{\mathrm{ext}} \boldsymbol{v}, & (81) \\ \text{get } \boldsymbol{a}_{\mathrm{ext}}(\boldsymbol{r}), & (82) \\ \boldsymbol{v} = \boldsymbol{v} + \dfrac{\Delta t_{\mathrm{ext}}}{2} \boldsymbol{a}_{\mathrm{ext}}^{m+1}, & (83) \end{array} \right.$$

$$\text{get } \boldsymbol{a}_{\mathrm{sph}}(\boldsymbol{r}), \tag{84}$$

$$\boldsymbol{v} = \boldsymbol{v} + \frac{\Delta t_{\mathrm{sph}}}{2} \boldsymbol{a}_{\mathrm{sph}}^n, \tag{85}$$

where $\boldsymbol{a}_{\mathrm{sph}}$ indicates the SPH acceleration evaluated from (34) and $\boldsymbol{a}_{\mathrm{ext}}$ indicates the external forces. The SPH and external accelerations are stored separately to enable this. Δt_{ext} is the minimum of all timesteps relating to sink–gas and external forces [equations (74)–(76)], while Δt_{sph} is the timestep relating to the SPH forces [equations (72), (73), and (288)]. Δt_{ext} is allowed to vary on each substep, so we take as many steps as required such that $\sum_j^{m-1} \Delta t_{\mathrm{ext},j} + \Delta t_{\mathrm{ext},f} = \Delta t_{\mathrm{sph}}$, where $\Delta t_{\mathrm{ext},f} < \Delta t_{\mathrm{ext},j}$ is chosen so that the sum will identically equal Δt_{sph}. The number of substeps is $m \approx \mathrm{int}(\Delta t_{\mathrm{ext,\,min}}/\Delta t_{\mathrm{sph,\,min}}) + 1$, where the minimum is taken over all particles.

2.3.4. *Individual particle timesteps*

For simulations of stiff problems with a large range in timestep over the domain, it is more efficient to allow each particle to evolve on its own timestep independently (Bate 1995; Springel 2005; Saitoh & Makino 2010). This violates all of the conservation properties of the Leapfrog integrator [see Makino et al. (2006) for an attempt to solve this], but can speed up the calculation by an order of magnitude or more. We implement this in the usual blockstepped manner by assigning timesteps in factor-of-two decrements from some maximum timestep $\Delta t_{\max}$, which for convenience is set equal to the time between output files.

We implement a timestep limiter where the timestep for an active particle is constrained to be within a factor of 2 of its neighbours, similar to condition employed by Saitoh & Makino (2009). Additionally, inactive particles will be woken up as required to ensure that their timestep is within a factor of 2 of its neighbours.

The practical side of individual timestepping is described in Appendix A.6.

2.4. External forces

2.4.1. *Point-mass potential*

The simplest external force describes a point mass, M, at the origin, which yields gravitational potential and acceleration:

$$\Phi_a = -\frac{GM}{r_a}, \qquad \boldsymbol{a}_{\mathrm{ext},a} = -\nabla \Phi_a = -\frac{GM}{|\boldsymbol{r}_a|^3} \boldsymbol{r}_a, \tag{86}$$

where $r_a \equiv |\boldsymbol{r}_a| \equiv \sqrt{\boldsymbol{r}_a \cdot \boldsymbol{r}_a}$. When this potential is used, we allow for particles within a certain radius, R_{acc}, from the origin to be accreted. This allows for a simple treatment of accretion discs where the mass of the disc is assumed to be negligible compared to the mass of the central object. The accreted mass in this case is recorded but not added to the central mass. For more massive discs, or when the accreted mass is significant with respect to the central mass, it is better to model the central star using a sink particle (Section 2.8) where there are mutual gravitational forces between the star and the disc, and any mass accreted is added to the point mass (Section 2.8.2).

2.4.2. *Binary potential*

We provide the option to model motion in binary systems where the mass of the disc is negligible. In this case, the binary motion is prescribed using

$$\boldsymbol{r}_1 = [(1-M)\cos(t), (1-M)\sin(t), 0], \tag{87}$$

$$\boldsymbol{r}_2 = [-M\cos(t), -M\sin(t), 0], \tag{88}$$

where M is the mass ratio in units of the total mass (which is therefore unity). For this potential, G and Ω are set to unity in computational units, where Ω is the angular velocity of the binary. Thus, only M needs to be specified to fix both m_1 and m_2. Hence, the binary remains fixed on a circular orbit at $r = 1$. The binary potential is therefore

$$\Phi_a = -\frac{M}{|\boldsymbol{r}_a - \boldsymbol{r}_1|} - \frac{(1-M)}{|\boldsymbol{r}_a - \boldsymbol{r}_2|}, \tag{89}$$

such that the external acceleration is given by

$$\boldsymbol{a}_{\text{ext},a} = -\nabla\Phi_a = -M\frac{(\boldsymbol{r}_a - \boldsymbol{r}_1)}{|\boldsymbol{r}_a - \boldsymbol{r}_1|^3} - (1-M)\frac{(\boldsymbol{r}_a - \boldsymbol{r}_2)}{|\boldsymbol{r}_a - \boldsymbol{r}_2|^3}. \tag{90}$$

Again, there is an option to accrete particles that fall within a certain radius from either star ($R_{\text{acc, 1}}$ or $R_{\text{acc, 2}}$, respectively). For most binary accretion disc simulations (e.g. planet migration), it is better to use 'live' sink particles to represent the binary so that there is feedback between the binary and the disc (we have used a live binary in all of our simulations to date, e.g. Nixon, King, & Price 2013; Facchini, Lodato, & Price 2013; Martin et al. 2014a, 2014b; Doğan et al. 2015; Ragusa, Lodato, & Price 2016; Ragusa et al. 2017), but the binary potential remains useful under limited circumstances—in particular, when one wishes to turn off the feedback between the disc and the binary.

Given that the binary potential is time-dependent, for efficiency, we compute the position of the binary only once at the start of each timestep, and use these stored positions to compute the accelerations of the SPH particles via (90).

2.4.3. Binary potential with gravitational wave decay

An alternative binary potential including the effects of gravitational wave decay was used by Cerioli, Lodato, & Price (2016) to study the squeezing of discs during the merger of supermassive black holes. Here the motion of the binary is prescribed according to

$$\boldsymbol{r}_1 = \left[-\frac{m_2}{m_1+m_2}a\cos(\theta), -\frac{m_2}{m_1+m_2}a\sin(\theta), 0\right],$$

$$\boldsymbol{r}_2 = \left[\frac{m_1}{m_1+m_2}a\cos(\theta), \frac{m_1}{m_1+m_2}a\sin(\theta), 0\right], \tag{91}$$

where the semi-major axis, a, decays according to

$$a(t) = a_0\left(1 - \frac{t}{\tau}\right)^{\frac{1}{4}}. \tag{92}$$

The initial separation is a_0, with τ defined as the time to merger, given by the usual expression (e.g. Lodato et al. 2009)

$$\tau \equiv \frac{5}{256}\frac{a_0^4}{\mu_{12}(m_1+m_2)^2}, \tag{93}$$

where

$$\mu_{12} \equiv \frac{m_1 m_2}{m_1+m_2}. \tag{94}$$

The angle θ is defined using

$$\Omega \equiv \frac{\mathrm{d}\theta}{\mathrm{d}t} = \sqrt{\frac{G(m_1+m_2)}{a^3}}. \tag{95}$$

Inserting the expression for a and integrating gives (Cerioli et al. 2016)

$$\theta(t) = -\frac{8\tau}{5}\sqrt{\frac{G(m_1+m_2)}{a_0^3}}\left(1 - \frac{t}{\tau}\right). \tag{96}$$

The positions of the binary, $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, can be inserted into (89) to obtain the binary potential, with the acceleration as given in (90). The above can be used as a simple example of a time-dependent external potential.

2.4.4. Galactic potentials

We implement a range of external forces representing various galactic potentials, as used in Pettitt et al. (2014). These include arm, bar, halo, disc, and spheroidal components. We refer the reader to the paper above for the actual forms of the potentials.

For the non-axisymmetric potentials, a few important parameters that determine the morphology can be changed at run time rather than compile time. These include the pattern speed, arm number, arm pitch angle, and bar axis lengths (where applicable). In the case of non-axisymmetric components, the user should be aware that some will add mass to the system, whereas others simply perturb the galactic disc. These potentials can be used for any galactic system, but the various default scale lengths and masses are chosen to match the Milky Way's rotation curve (Sofue 2012).

The most basic potential in PHANTOM is a simple logarithmic potential from Binney & Tremaine (1987), which allows for the reproduction of a purely flat rotation curve with steep decrease at the galactic centre, and approximates the halo, bulge, and disc contributions. Also included is the standard flattened disc potential of Miyamoto–Nagai (Miyamoto & Nagai 1975) and an exponential profile disc, specifically the form from Khoperskov et al. (2013). Several spheroidal components are available, including the potentials of Plummer (1911), Hernquist (1990), and Hubble (1930). These can be used generally for bulges and halos if given suitable mass and scale lengths. We also include a few halo-specific profiles; the NFW (Navarro, Frenk, & White 1996), Begeman, Broeils, & Sanders (1991), Caldwell & Ostriker (1981), and the Allen & Santillan (1991) potentials.

The arm potentials include some of the more complicated profiles. The first is the potential of Cox & Gómez (2002), which is a relatively straightforward superposition of three sinusoidal-based spiral components to damp the potential 'troughs' in the inter-arm minima. The other spiral potential is from Pichardo et al. (2003), and is more complicated. Here, the arms are constructed from a superposition of oblate spheroids whose loci are placed along a standard logarithmic spiral. As the force from this potential is computationally expensive it is prudent to pre-compute a grid of potential/force and read it at run time. The python code to generate the appropriate grid files is distributed with the code.

Finally, the bar components: We include the bar potentials of Dehnen (2000a), Wada & Koda (2001), the 'S' shaped bar of Vogt & Letelier (2011), both biaxial and triaxial versions provided in Long & Murali (1992), and the boxy-bulge bar of Wang et al. (2012). This final bar contains both a small inner non-axisymmetric bulge and longer bar component, with the forces calculated by use of Hernquist–Ostriker expansion coefficients of the bar density field. PHANTOM contains the coefficients for several different forms of this bar potential.

2.4.5. *Lense–Thirring precession*

Lense–Thirring precession (Lense & Thirring 1918) from a spinning black hole is implemented in a post-Newtonian approximation following Nelson & Papaloizou (2000), which has been used in Nixon et al. (2012b), Nealon, Price, & Nixon (2015), and Nealon et al. (2016). In this case, the external acceleration consists of a point-mass potential (Section 2.4.1) and the Lense–Thirring term:

$$\boldsymbol{a}_{\mathrm{ext},a} = -\nabla\Phi_a + \boldsymbol{v}_a \times \boldsymbol{\Omega}_{p,a}, \tag{97}$$

where Φ_a is given by (86) and $\boldsymbol{v}_a \times \boldsymbol{\Omega}_{p,a}$ is the gravitomagnetic acceleration. A dipole approximation is used, yielding

$$\boldsymbol{\Omega}_{p,a} \equiv \frac{2\boldsymbol{S}}{|\boldsymbol{r}_a|^3} - \frac{6(\boldsymbol{S}\cdot\boldsymbol{r}_a)\boldsymbol{r}_a}{|\boldsymbol{r}_a|^5}, \tag{98}$$

with $\boldsymbol{S} = a_{\mathrm{spin}}(GM)^2\boldsymbol{k}/c^3$, where $\boldsymbol{k}$ is a unit vector in the direction of the black hole spin. When using the Lense–Thirring force, geometric units are assumed such that $G = M = c = 1$, as described in Section 2.2.3, but with the additional constraints on the unit system from M and c.

Since in this case the external force depends on velocity, it cannot be implemented directly into Leapfrog. The method we employ to achieve this is simpler than those proposed elsewhere [c.f. attempts by Quinn et al. (2010) and Rein & Tremaine (2011) to adapt the Leapfrog integrator to Hill's equations]. Our approach is to split the acceleration into position and velocity-dependent parts, i.e.

$$\boldsymbol{a}_{\mathrm{ext}} = \boldsymbol{a}_{\mathrm{ext,x}}(\boldsymbol{r}) + \boldsymbol{a}_{\mathrm{ext,v}}(\boldsymbol{r}, \boldsymbol{v}). \tag{99}$$

The position-dependent part [i.e. $-\nabla\Phi(\boldsymbol{r})$] is integrated as normal. The velocity-dependent Lense–Thirring term is added to the predictor step, (66)–(67), as usual, but the corrector step, (69), is written in the form

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^{n+\frac{1}{2}} + \frac{1}{2}\Delta t\left[\boldsymbol{a}^{n+1}_{\mathrm{sph}} + \boldsymbol{a}^{n+1}_{\mathrm{ext,x}} + \boldsymbol{a}_{\mathrm{ext,v}}(\boldsymbol{r}^{n+1}, \boldsymbol{v}^{n+1})\right], \tag{100}$$

where $\boldsymbol{v}^{n+\frac{1}{2}} \equiv \boldsymbol{v}^n + \frac{1}{2}\Delta t\boldsymbol{a}^n$ as in (65). This equation is implicit but the trick is to notice that it can be solved analytically for simple forces[4]. In the case of Lense–Thirring precession, we have

$$\boldsymbol{v}^{n+1} = \tilde{\boldsymbol{v}} + \frac{1}{2}\Delta t\left[\boldsymbol{v}^{n+1} \times \boldsymbol{\Omega}_p(\boldsymbol{r}^{n+1})\right], \tag{101}$$

where $\tilde{\boldsymbol{v}} \equiv \boldsymbol{v}^{n+\frac{1}{2}} + \frac{1}{2}\Delta t(\boldsymbol{a}^{n+1}_{\mathrm{sph}} + \boldsymbol{a}^{n+1}_{\mathrm{ext,x}})$. We therefore have a matrix equation in the form

$$\boldsymbol{R}\boldsymbol{v}^{n+1} = \tilde{\boldsymbol{v}}, \tag{102}$$

where $\boldsymbol{R}$ is the 3 × 3 matrix given by

$$\boldsymbol{R} \equiv \begin{bmatrix} 1 & -\frac{\Delta t}{2}\Omega^z_p & \frac{\Delta t}{2}\Omega^y_p \\ \frac{\Delta t}{2}\Omega^z_p & 1 & -\frac{\Delta t}{2}\Omega^x_p \\ -\frac{\Delta t}{2}\Omega^y_p & \frac{\Delta t}{2}\Omega^x_p & 1 \end{bmatrix}. \tag{103}$$

Rearranging (102), $\boldsymbol{v}^{n+1}$ is obtained by using

$$\boldsymbol{v}^{n+1} = \boldsymbol{R}^{-1}\tilde{\boldsymbol{v}}, \tag{104}$$

where $\boldsymbol{R}^{-1}$ is the inverse of $\boldsymbol{R}$, which we invert using the analytic solution.

2.4.6. *Generalised Newtonian potential*

The generalised Newtonian potential described by Tejeda & Rosswog (2013) is implemented, where the acceleration terms are given by

$$\boldsymbol{a}_{\mathrm{ext},a} = -\frac{GM\boldsymbol{r}_a}{|\boldsymbol{r}_a|^3}f^2 + \frac{2R_{\mathrm{g}}\boldsymbol{v}_a(\boldsymbol{v}_a\cdot\boldsymbol{r}_a)}{|\boldsymbol{r}_a|^3 f} - \frac{3R_{\mathrm{g}}\boldsymbol{r}_a(\boldsymbol{v}_a\times\boldsymbol{r}_a)^2}{|\boldsymbol{r}_a|^5}, \tag{105}$$

with $R_{\mathrm{g}} \equiv GM/c^3$ and $f \equiv \left(1 - 2R_{\mathrm{g}}/|\boldsymbol{r}_a|\right)$. See Bonnerot et al. (2016) for a recent application. This potential reproduces several features of the Schwarzschild (1916) spacetime, in particular, reproducing the orbital and epicyclic frequencies to better than 7% (Tejeda & Rosswog 2013). As the acceleration involves velocity-dependent terms, it requires a semi-implicit solution like Lense–Thirring precession. Since the matrix equation is rather involved for this case, the corrector step is iterated using fixed point iterations until the velocity of each particle is converged to a tolerance of 1%.

2.4.7. *Poynting–Robertson drag*

The radiation drag from a central point-like, gravitating, radiating, and non-rotating object may be applied as an external force. The implementation is intended to be as general as possible. The acceleration of a particle subject to these external forces is

$$\begin{aligned}\boldsymbol{a}_{\mathrm{ext},a} = {} & \frac{(k_0\beta_{\mathrm{PR}} - 1)GM}{|\boldsymbol{r}_a|^3}\boldsymbol{r}_a \\ & -\beta_{\mathrm{PR}}\left(k_1\frac{GM}{|\boldsymbol{r}_a|^3}\frac{v_r}{c}\boldsymbol{r}_a - k_2\frac{GM}{|\boldsymbol{r}_a|^2}\frac{\boldsymbol{v}_a}{c}\right),\end{aligned} \tag{106}$$

where v_r is the component of the velocity in the radial direction. The parameter β_{PR} is the ratio of radiation to gravitational forces, supplied by a separate user-written module.

[4] The procedure for Hill's equations would be identical to our method for Lense–Thirring precession. The method we use is both simpler and more direct than any of the schemes proposed by Quinn et al. (2010) and Rein & Tremaine (2011), and is time reversible unlike the methods proposed in those papers.

Relativistic effects are neglected because these are thought to be less important than radiation forces for low ($\beta_{\rm PR} < 0.01$) luminosities, even in accreting neutron star systems where a strong gravitational field is present (e.g., Miller & Lamb 1993).

The three terms on the right side of (106) correspond, respectively, to gravity (reduced by outward radiation pressure), redshift-related modification to radiation pressure caused by radial motion, and Poynting–Robertson drag against the direction of motion. These three terms can be scaled independently by changing the three parameters k_0, k_1, and k_2, whose default values are unity. Rotation of the central object can be crudely emulated by changing k_2.

As for Lense–Thirring precession, the $\boldsymbol{a}^{n+1}$ term of the Leapfrog integration scheme can be expanded into velocity-dependent and non-velocity-dependent component. We obtain, after some algebra,

$$\boldsymbol{v}^{n+1} = -\frac{\boldsymbol{T} - Qk_1(\boldsymbol{v}^{n+1}\cdot\hat{\boldsymbol{r}})\hat{\boldsymbol{r}}}{1 + Qk_2}, \tag{107}$$

where

$$\boldsymbol{T} = \boldsymbol{v}^n + \frac{1}{2}\Delta t\boldsymbol{a}^n - \frac{(1 - k_0\beta_{\rm PR})GM\Delta t}{2r^3}\boldsymbol{r} \tag{108}$$

and

$$Q = \frac{GM\beta_{\rm PR}\Delta t}{2cr^2}. \tag{109}$$

Equation (107) yields a set of simultaneous equations for the three vector components that can be solved analytically. A detailed derivation is given in Worpel (2015).

2.4.8. Coriolis and centrifugal forces

Under certain circumstances, it is useful to perform calculations in a co-rotating reference frame (e.g. for damping binary stars into equilibrium with each other). The resulting acceleration terms are given by

$$\boldsymbol{a}_{{\rm ext},a} = -\boldsymbol{\Omega}\times(\boldsymbol{\Omega}\times\boldsymbol{r}_a) - 2(\boldsymbol{\Omega}\times\boldsymbol{v}_a), \tag{110}$$

which are the centrifugal and Coriolis terms, respectively, with $\boldsymbol{\Omega}$ the angular rotation vector. The timestepping algorithm is as described above for Lense–Thirring precession, with the velocity-dependent term handled by solving the 3 × 3 matrix in the Leapfrog corrector step.

2.5. Driven turbulence

Phantom implements turbulence driving in periodic domains via an Ornstein–Uhlenbeck stochastic driving of the acceleration field, as first suggested by Eswaran & Pope (1988). This is an SPH adaptation of the module used in the grid-based simulations by Schmidt, Hillebrandt, & Niemeyer (2006) and Federrath, Klessen, & Schmidt (2008) and many subsequent works. This module was first used in Phantom by Price & Federrath (2010) to compare the statistics of isothermal, supersonic turbulence between SPH, and grid methods. Subsequent applications have been to the density variance–Mach number relation (Price et al. 2011), subsonic turbulence (Price 2012b), supersonic MHD turbulence (Tricco et al. 2016b), and supersonic turbulence in a dust–gas mixture (Tricco et al. 2017). Adaptations of this module have also been incorporated into other SPH codes (Bauer & Springel 2012; Valdarnini 2016).

The amplitude and phase of each Fourier mode is initialised by creating a set of six random numbers, z_n, drawn from a random Gaussian distribution with unit variance. These are generated by the usual Box–Muller transformation (e.g. Press et al. 1992) by selecting two uniform random deviates u_1, u_2 $\in [0, 1]$ and constructing the amplitude according to

$$z = \sqrt{2\log(1/u_1)}\cos(2\pi u_2). \tag{111}$$

The six Gaussian random numbers are set up according to

$$\boldsymbol{x}_n = \sigma z_n, \tag{112}$$

where the standard deviation, σ, is set to the square root of the energy per mode divided by the correlation time, $\sigma = \sqrt{E_{\rm m}/t_{\rm decay}}$, where both $E_{\rm m}$ and $t_{\rm decay}$ are user-specified parameters.

The ‘red noise’ sequence (Uhlenbeck & Ornstein 1930) is generated for each mode at each timestep according to (Bartosch 2001)

$$\boldsymbol{x}_{n+1} = f\boldsymbol{x}_n + \sigma\sqrt{(1 - f^2)}z_n, \tag{113}$$

where $f = \exp(-\Delta t/t_{\rm decay})$ is the damping factor. The resulting sequence has zero mean with root-mean-square equal to the variance. The power spectrum in the time domain can vary from white noise [$P(f)$ constant] to ‘brown noise’ [$P(f) = 1/f^2$].

The amplitudes and phases of each mode are constructed by splitting $\boldsymbol{x}_n$ into two vectors, $\boldsymbol{\Phi}_a$ and $\boldsymbol{\Phi}_b$ of length 3, employed to construct divergence- and curl-free fields according to

$$\boldsymbol{A}_m = w[\boldsymbol{\Phi}_a - (\boldsymbol{\Phi}_a\cdot\hat{\boldsymbol{k}})\hat{\boldsymbol{k}}] + (1 - w)[(\boldsymbol{\Phi}_b\cdot\hat{\boldsymbol{k}})\hat{\boldsymbol{k}}], \tag{114}$$

$$\boldsymbol{B}_m = w[\boldsymbol{\Phi}_b - (\boldsymbol{\Phi}_b\cdot\hat{\boldsymbol{k}})\hat{\boldsymbol{k}}] + (1 - w)[(\boldsymbol{\Phi}_a\cdot\hat{\boldsymbol{k}})\hat{\boldsymbol{k}}], \tag{115}$$

where $\boldsymbol{k} = [k_x, k_y, k_z]$ is the mode frequency. The parameter $w \in [0, 1]$ is the ‘solenoidal weight’, specifying whether the driving should be purely solenoidal ($w = 1$) or purely compressive ($w = 0$) (see Federrath et al. 2008, 2010a).

The spectral form of the driving is defined in Fourier space, with options for three possible spectral forms

$$C_m = \begin{cases} 1 & \text{uniform} \\ 4(a_{\rm min} - 1)\dfrac{(k - k_c)^2}{(k_{\rm max} - k_{\rm min})^2} + 1 & \text{parabolic} \\ k/k_{\rm min}^{-5/3} & \text{Kolmogorov,} \end{cases} \tag{116}$$

where $k = \sqrt{k_x^2 + k_y^2 + k_z^2}$ is the wavenumber, with non-zero amplitudes defined only for wavenumbers where $k_{\rm min} \leqslant k \leqslant k_{\rm max}$, and $a_{\rm min}$ is the amplitude of the modes at $k_{\rm min}$ and $k_{\rm max}$ in the parabolic case (we use $a_{\rm min} = 0$ in the code). The frequency grid is defined using frequencies from $k_x = n_x 2\pi/L_x$

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

in the x direction, where $n_x \in [0, 20]$ is an integer and L_x is the box size in the x-direction, while $k_y = n_y 2\pi/L_y$ and $k_z = n_z 2\pi/L_z$ with $n_y \in [0, 8]$ and $n_z \in [0, 8]$. We then set up four modes for each combination of n_x, n_y, and n_z, corresponding to $[k_x, k_y, k_z]$, $[k_x, -k_y, k_z]$, $[k_x, k_y, -k_z]$, and $[k_x, -k_y, -k_z]$. That is, we effectively sum from $[-(N-1)/2, (N-1)/2]$ in the k_y and k_z directions in the discrete Fourier transform, where $N = \max(n_x)$ is the number of frequencies. The default values for $k_{\rm min}$ and $k_{\rm max}$ are 2π and 6π, respectively, corresponding to large-scale driving of only the first few Fourier modes, so with default settings there are 112 non-zero Fourier modes. The maximum number of modes, defining the array sizes needed to store the stirring information, is currently set to 1,000.

We apply the forcing to the particles by computing the discrete Fourier transform over the stirring modes directly, i.e.

$$\boldsymbol{a}_{{\rm forcing},a} = f_{\rm sol} \sum_{m=1}^{n_{\rm modes}} C_m \left[\boldsymbol{A}_m \cos(\boldsymbol{k}\cdot\boldsymbol{r}_a) - \boldsymbol{B}_m \sin(\boldsymbol{k}\cdot\boldsymbol{r}_a)\right], \qquad (117)$$

where the factor $f_{\rm sol}$ is defined from the solenoidal weight, w, according to

$$f_{\rm sol} = \sqrt{\frac{3}{n_{\rm dim}}}\sqrt{\frac{3}{1 - 2w + n_{\rm dim} w^2}}, \qquad (118)$$

such that the rms acceleration is the same irrespective of the value of w. We default to purely solenoidal forcing ($w = 1$), with the factor $f_{\rm sol}$ thus equal to $\sqrt{3/2}$ by default. For individual timesteps, we update the driving force only when a particle is active.

To aid reproducibility, it is often desirable to pre-generate the entire driving sequence prior to performing a calculation, which can then be written to a file and read back at runtime. This was the procedure used in Price & Federrath (2010), Tricco et al. (2016b), and Tricco et al. (2017).

2.6. Accretion disc viscosity

Accretion disc viscosity is implemented in PHANTOM via two different approaches, as described by Lodato & Price (2010).

2.6.1. *Disc viscosity using the shock viscosity term*

The default approach is to adapt the shock viscosity term to represent a Shakura & Sunyaev (1973) α-viscosity, as originally proposed by Artymowicz & Lubow (1994) and Murray (1996). The key is to note that (39) and (40) represent a Navier–Stokes viscosity term with a fixed ratio between the bulk and shear viscosity terms (e.g. Murray 1996; Jubelgas, Springel, & Dolag 2004; Lodato & Price 2010; Price 2012b; Meru & Bate 2012). In particular, it can be shown (e.g. Español & Revenga 2003) that

$$\sum_b \frac{m_b}{\overline{\rho}_{ab}} (\boldsymbol{v}_{ab}\cdot\hat{\boldsymbol{r}}_{ab}) \frac{\nabla_a W_{ab}}{|r_{ab}|} \approx \frac{1}{5}\nabla(\nabla\cdot\boldsymbol{v}) + \frac{1}{10}\nabla^2\boldsymbol{v}, \qquad (119)$$

where $\overline{\rho}_{ab}$ is some appropriate average of the density. This enables the artificial viscosity term, (39), to be translated into the equivalent Navier–Stokes terms. In order for the artificial viscosity to represent a disc viscosity, we make the following modifications (Lodato & Price 2010):

1. The viscosity term is applied for both approaching and receding particles.
2. The speed $v_{\rm sig}$ is set equal to $c_{\rm s}$.
3. A constant $\alpha^{\rm AV}$ is adopted, turning off shock detection switches (Section 2.2.9).
4. The viscosity term is multiplied by a factor $h/|r_{ab}|$.

The net result is that (40) becomes

$$q^a_{ab} = \begin{cases} -\dfrac{\rho_a h_a}{2|r_{ab}|}\left(\alpha^{\rm AV} c_{{\rm s},a} + \beta_{\rm AV}|\boldsymbol{v}_{ab}\cdot\hat{\boldsymbol{r}}_{ab}|\right)\boldsymbol{v}_{ab}\cdot\hat{\boldsymbol{r}}_{ab}, & \boldsymbol{v}_{ab}\cdot\hat{\boldsymbol{r}}_{ab} < 0 \\ -\frac{\rho_a h_a}{2|r_{ab}|}\alpha^{\rm AV} c_{{\rm s},a}\boldsymbol{v}_{ab}\cdot\hat{\boldsymbol{r}}_{ab}. & \text{otherwise.} \end{cases} \qquad (120)$$

With the above modifications, the shear and bulk coefficients can be translated using (119) to give (e.g. Monaghan 2005; Lodato & Price 2010; Meru & Bate 2012)

$$\nu_{\rm AV} \approx \frac{1}{10}\alpha^{\rm AV} c_{\rm s} h, \qquad (121)$$

$$\zeta_{\rm AV} = \frac{5}{3}\nu_{\rm AV} \approx \frac{1}{6}\alpha^{\rm AV} c_{\rm s} h. \qquad (122)$$

The Shakura–Sunyaev prescription is

$$\nu = \alpha_{\rm SS} c_{\rm s} H, \qquad (123)$$

where H is the scale height. This implies that $\alpha_{\rm SS}$ may be determined from $\alpha_{\rm AV}$ using

$$\alpha_{\rm SS} \approx \frac{\alpha^{\rm AV}}{10}\frac{\langle h\rangle}{H}, \qquad (124)$$

where $\langle h\rangle$ is the mean smoothing length on particles in a cylindrical ring at a given radius.

In practice, this means that one must uniformly resolve the scale height in order to obtain a constant $\alpha_{\rm SS}$ in the disc. We have achieved this in simulations to date by choosing the initial surface density profile and the power-law index of the temperature profile (when using a locally isothermal EOS) to ensure that this is the case (Lodato & Pringle 2007). Confirmation that the scaling provided by (124) is correct is shown in Figure 4 of Lodato & Price (2010) and is checked automatically in the PHANTOM test suite.

In the original implementation (Lodato & Price 2010), we also set the $\beta^{\rm AV}$ to zero, but this is dangerous if the disc dynamics are complex as there is nothing to prevent particle penetration (see Section 2.2.7). Hence, in the current version of the code, $\beta^{\rm AV} = 2$ by default even if disc viscosity is set, but is only applied to approaching particles (c.f. 120). Applying any component of viscosity to only approaching particles can affect the sign of the precession induced in warped discs (Lodato & Pringle 2007), but in general retaining the $\beta^{\rm AV}$

term is safer with no noticeable effect on the overall dissipation due to the second-order dependence of this term on resolution.

Using $\alpha^{\rm AV}$ to set the disc viscosity has two useful consequences. First, it forces one to consider whether or not the scale height, H, is resolved. Second, knowing the value of $\alpha^{\rm AV}$ is helpful, as $\alpha^{\rm AV} \approx 0.1$ represents the lower bound below which a physical viscosity is not resolved in SPH (that is, viscosities smaller than this produce disc spreading independent of the value of $\alpha^{\rm AV}$, see Bate 1995; Meru & Bate 2012), while $\alpha^{\rm AV} > 1$ constrains the timestep (Section 2.3.2).

2.6.2. Disc viscosity using the Navier–Stokes viscosity terms

An alternative approach is to compute viscous terms directly from the Navier–Stokes equation. Details of how the Navier–Stokes terms are represented are given below (Section 2.7), but for disc viscosity a method for determining the kinematic viscosity is needed, which in turn requires specifying the scale height as a function of radius. We use

$$H_a \equiv \frac{c_{\rm s}^a}{\Omega(R_a)}, \tag{125}$$

where we assume Keplerian rotation $\Omega = \sqrt{GM/R^3}$ and $c_{\rm s}$ is obtained for a given particle from the EOS (which for consistency must be either isothermal or locally isothermal). It is important to note that this restricts the application of this approach only to discs where R can be meaningfully defined, excluding, for example, discs around binary stars.

The shear viscosity is then set using

$$\nu_a = \alpha_{\rm SS} c_{\rm s}^a H_a, \tag{126}$$

where $\alpha_{\rm SS}$ is a pre-defined/input parameter. The timestep is constrained using $C_{\rm visc} h^2/\nu$ as described in Section 2.7. The advantage to this approach is that the shear viscosity is set directly and does not depend on the smoothing length. However, as found by Lodato & Price (2010), it remains necessary to apply some bulk viscosity to capture shocks and prevent particle penetration of the disc midplane, so one should apply the shock viscosity as usual. Using a shock-detection switch (Section 2.2.9) means that this is usually not problematic. This formulation of viscosity was used in Facchini et al. (2013).

2.7. Navier–Stokes viscosity

Physical viscosity is implemented as described in Lodato & Price (2010). Here, (23) and (24) are replaced by the compressible Navier–Stokes equations, i.e.

$$\frac{{\rm d}v^i}{{\rm d}t} = -\frac{1}{\rho}\frac{\partial S^{ij}_{\rm NS}}{\partial x^j} + \Pi_{\rm shock} + \boldsymbol{a}_{\rm ext}(\boldsymbol{r}, t) + \boldsymbol{a}_{\rm sink-gas} + \boldsymbol{a}_{\rm selfgrav}, \tag{127}$$

$$\frac{{\rm d}u}{{\rm d}t} = -\frac{P}{\rho}(\nabla\cdot\boldsymbol{v}) + \Lambda_{\rm visc} + \Lambda_{\rm shock} - \frac{\Lambda_{\rm cool}}{\rho}, \tag{128}$$

with the stress tensor given by

$$S^{ij}_{\rm NS} = \left[P - \left(\zeta - \frac{2}{3}\eta\right)\frac{\partial v^k}{\partial x^k}\right]\delta^{ij} - \eta\left(\frac{\partial v^i}{\partial x^j} + \frac{\partial v^j}{\partial x^i}\right), \tag{129}$$

where δ^{ij} is the Kronecker delta, and ζ and η are the bulk and shear viscosity coefficients, related to the volume and kinematic shear viscosity coefficients by $\zeta_v \equiv \zeta/\rho$ and $\nu \equiv \eta/\rho$.

2.7.1. Physical viscosity using two first derivatives

As there is no clear consensus on the best way to implement physical viscosity in SPH, PHANTOM currently contains two implementations. The simplest is to use two first derivatives, which is essentially that proposed by Flebbe et al. (1994), Watkins et al. (1996), and Sijacki & Springel (2006). In this case, (127) is discretised in the standard manner using

$$\frac{{\rm d}v^i_a}{{\rm d}t} = -\sum_b m_b\left[\frac{S^{ij}_{{\rm NS},a}}{\Omega_a\rho_a^2}\frac{\partial W_{ab}(h_a)}{\partial x^j_a} + \frac{S^{ij}_{{\rm NS},b}}{\Omega_b\rho_b^2}\frac{\partial W_{ab}(h_b)}{\partial x^j_a}\right] + \Pi^i_{\rm shock} + a^i_{\rm ext}(\boldsymbol{r}, t) + a^i_{\rm sink-gas} + a^i_{\rm selfgrav}, \tag{130}$$

where the velocity gradients are computed during the density loop using

$$\frac{\partial v^i_a}{\partial x^j_a} = -\frac{1}{\Omega_a\rho_a}\sum_b m_b v^i_{ab}\nabla^j_a W_{ab}(h_a). \tag{131}$$

Importantly, the differenced SPH operator is used in (131), whereas (130) uses the symmetric gradient operator. The use of conjugate operators[5] is a common requirement in SPH in order to guarantee energy conservation and a positive definite entropy increase from dissipative terms (e.g. Price 2010; Tricco & Price 2012). Total energy conservation means that

$$\frac{{\rm d}E}{{\rm d}t} = \sum_a m_a\left(\frac{{\rm d}u_a}{{\rm d}t} + v^i_a\frac{{\rm d}v^i_a}{{\rm d}t}\right) = 0. \tag{132}$$

This implies a contribution to the thermal energy equation given by

$$\frac{{\rm d}u_a}{{\rm d}t} = \frac{S^{ij}_{{\rm NS},a}}{\Omega_a\rho_a^2}\sum_b m_b v^i_{ab}\nabla^j_a W_{ab}(h_a), \tag{133}$$

which can be seen to reduce to (24) in the inviscid case ($S^{ij}_{\rm NS} = P\delta^{ij}$), but in general is an SPH expression for

$$\frac{{\rm d}u_a}{{\rm d}t} = -\frac{S^{ij}_{{\rm NS},a}}{\rho_a}\frac{\partial v^i_a}{\partial x^j_a}. \tag{134}$$

Using $S^{ij}_{\rm NS} = S^{ji}_{\rm NS}$, we have

$$\frac{{\rm d}u_a}{{\rm d}t} = -\frac{1}{2}\frac{S^{ij}_{{\rm NS},a}}{\rho_a}\left(\frac{\partial v^i_a}{\partial x^j_a} + \frac{\partial v^j_a}{\partial x^i_a}\right), \tag{135}$$

which, using (129), gives

$$\Lambda_{\rm visc} = \left(\zeta_{v,a} - \frac{2}{3}\nu_a\right)(\nabla\cdot\boldsymbol{v})^2_a + \frac{\nu_a}{2}\left(\frac{\partial v^i_a}{\partial x^j_a} + \frac{\partial v^j_a}{\partial x^i_a}\right)^2. \tag{136}$$

[5] The SPH difference operator is discretely skew adjoint to the symmetric operator (Cummins & Rudman 1999).

By the square in the last term, we mean the tensor summation $\Sigma_j \Sigma_i T_{ij} T_{ij}$, where $T_{ij} \equiv \partial v_a^i / \partial x_a^j + \partial v_a^j / \partial x_a^i$. The heating term is therefore positive definite provided that the velocity gradients and divergence are computed using the difference operator (131), both in (136) and when computing the stress tensor (129).

The main disadvantage of the first derivatives approach is that it requires storing the strain tensor for each particle, i.e. six additional quantities when taking account of symmetries.

2.7.2. *Physical viscosity with direct second derivatives*

The second approach is to use SPH second derivative operators directly. Here we use modified versions of the identities given by Español & Revenga (2003) (see also Monaghan 2005; Price 2012a), namely

$$\nabla\left[A(\nabla\cdot\boldsymbol{v})\right] \approx - \sum_b m_b \left[\frac{A_a}{\rho_a} G_{ab}(h_a) + \frac{A_b}{\rho_b} G_{ab}(h_b)\right] (\boldsymbol{v}_{ab}\cdot\hat{\boldsymbol{r}}_{ab})\hat{\boldsymbol{r}}_{ab}, \tag{137}$$

$$\nabla\cdot(C\nabla\boldsymbol{v}) \approx - \sum_b m_b \left[\frac{C_a}{\rho_a} G_{ab}(h_a) + \frac{C_b}{\rho_b} G_{ab}(h_b)\right] \boldsymbol{v}_{ab}, \tag{138}$$

where $G_{ab} \equiv -2F_{ab}/|r_{ab}|$, i.e. the scalar part of the kernel gradient divided by the particle separation, which can be thought of as equivalent to defining a new 'second derivative kernel' (Brookshaw 1985, 1994; Price 2012a; Price & Laibe 2015a).

From the compressible Navier–Stokes equations, (127) with (129), the coefficients in these two terms are

$$A \equiv \frac{1}{2}\left(\zeta + \frac{\eta}{3}\right), \tag{139}$$

$$C \equiv \frac{1}{2}\eta, \tag{140}$$

so that we can simply use

$$\left(\frac{\mathrm{d}v_a^i}{\mathrm{d}t}\right)_{\rm visc} = \sum_b m_b \left[\frac{\tau_a}{\rho_a} G_{ab}(h_a) + \frac{\tau_b}{\rho_b} G_{ab}(h_b)\right] (\boldsymbol{v}_{ab}\cdot\hat{\boldsymbol{r}}_{ab})\hat{r}_{ab}^i + \sum_b m_b \left[\frac{\kappa_a}{\rho_a} G_{ab}(h_a) + \frac{\kappa_b}{\rho_b} G_{ab}(h_b)\right] v_{ab}^i \tag{141}$$

where

$$\tau = \frac{5}{2}A, \tag{142}$$

$$\kappa = \left(C - \frac{A}{2}\right). \tag{143}$$

The corresponding heating terms in the thermal energy equation are given by

$$\Lambda_{\rm visc} = \frac{\tau_a}{\rho_a} \sum_b m_b (\boldsymbol{v}_{ab}\cdot\hat{\boldsymbol{r}}_{ab})^2 G_{ab}(h_a) + \frac{\kappa_a}{\rho_a} \sum_b m_b (\boldsymbol{v}_{ab})^2 G_{ab}(h_a). \tag{144}$$

This is the default formulation of Navier–Stokes viscosity in the code since it does not require additional storage. In practice, we have found little difference between the two formulations of physical viscosity, but this would benefit from a detailed study. In general one might expect the two first derivatives formulation to offer a less noisy estimate at the cost of additional storage. However, direct second derivatives are the method used in 'Smoothed Dissipative Particle Dynamics' (Español & Revenga 2003).

2.7.3. *Timestep constraint*

Both approaches to physical viscosity use explicit timestepping, and therefore imply a constraint on the timestep given by

$$\Delta t_{\rm visc}^a \equiv C_{\rm visc} \frac{h_a^2}{\nu_a}, \tag{145}$$

where $C_{\rm visc} = 0.25$ by default (Brookshaw 1994). When physical viscosity is turned on, this constraint is included with other timestep constraints according to (77).

2.7.4. *Physical viscosity and the tensile instability*

Caution is required in the use of physical viscosity at high Mach number, since negative stress can lead to the tensile instability (Morris 1996b; Monaghan 2000; Gray, Monaghan, & Swift 2001). For subsonic applications, this is usually not a problem since the strain tensor and velocity divergence are small compared to the pressure. In the current code, we simply emit a warning if physical viscosity leads to negative stresses during the calculation, but this would benefit from a detailed study.

2.7.5. *Physical viscosity and angular momentum conservation*

Neither method for physical viscosity exactly conserves angular momentum because the force is no longer directed along the line of sight joining the particles. However, the error is usually small (see discussion in Bonet & Lok 1999, Section 5 of Price & Monaghan 2004b or Hu & Adams 2006). Recently, Müller, Fedosov, & Gompper (2015) have proposed an algorithm for physical viscosity in SPH that explicitly conserves angular momentum by tracking particle spin, which may be worth investigating.

2.8. Sink particles

Sink particles were introduced into SPH by Bate, Bonnell, & Price (1995) in order to follow star formation simulations beyond the point of fragmentation. In PHANTOM, these are treated separately to the SPH particles, and interact with other particles, including other sink particles, only via gravity. The differences with other point-mass particles implemented in the code (e.g. dust, stars, and dark matter) are that (i) the gravitational interaction is computed using a direct N^2 summation which is *not* softened by default (i.e., the N-body algorithm is collisional); (ii) they are allowed to accrete gas; and (iii) they store the accreted angular momentum and other extended properties, such as the accreted mass. Sink particles are evolved in time using the RESPA algorithm (Section 2.3.3), which is second-order accurate, symplectic, and allows sink particles to evolve on shorter timesteps compared to SPH particles.

2.8.1. *Sink particle accelerations*

The equations of motion for a given sink particle, i, are

$$\frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t} = -\sum_{j=1}^{N_{\rm sink}} GM_j\phi'_{ij}(\epsilon)\hat{\boldsymbol{r}}_{ij} - \sum_{b=1}^{N_{\rm part}} Gm_b\phi'_{ib}(\epsilon_{ib})\hat{\boldsymbol{r}}_{ib}, \tag{146}$$

where ϕ'_{ab} is the softening kernel (Section 2.12.2), $N_{\rm part}$ is the total number of gas particles, and $N_{\rm sink}$ is the total number of sink particles. The sink–gas softening length, ε_{ib}, is defined as the maximum of the (fixed) softening length defined for the sink particles, ε, and the softening length of the gas particle, ε_b. That is, $\varepsilon_{ib} \equiv \max(\varepsilon, \varepsilon_b)$. SPH particles receive a corresponding acceleration

$$\boldsymbol{a}^a_{\rm sink-gas} = -\sum_{j=1}^{N_{\rm sink}} GM_j\phi'_{aj}(\epsilon_{aj})\hat{\boldsymbol{r}}_{aj}. \tag{147}$$

Softening of sink–gas interactions is not applied if the softening length for sink particles is set to zero, in which case the sink–gas accelerations reduce simply to

$$\boldsymbol{a}^a_{\rm sink-gas} = -\sum_{j=1}^{N_{\rm sink}} \frac{GM_j}{|\boldsymbol{r}_a - \boldsymbol{r}_j|^3}\boldsymbol{r}_{aj}. \tag{148}$$

This is the default behaviour when sink particles are used in the code. Softening of sink–gas interactions is useful to model a point-mass particle that does not accrete gas (e.g. by setting the accretion radius to zero). For example, we used a softened sink particle to simulate the core of the red giant in Iaconi et al. (2017). The sink–sink interaction is unsoftened by default ($\varepsilon = 0$), giving the usual

$$\boldsymbol{a}^i_{\rm sink-sink} = -\sum_{j=1}^{N_{\rm sink}} \frac{GM_j}{|\boldsymbol{r}_i - \boldsymbol{r}_j|^3}\boldsymbol{r}_{ij}. \tag{149}$$

Caution is required when integrating tight binary or multiple systems when $\varepsilon = 0$ to ensure that the timestep conditions (Section 2.3.2) are strict enough.

2.8.2. *Accretion onto sink particles*

Accretion of gas particles onto a sink particle occurs when a gas particle passes a series of accretion checks within the accretion radius $r_{\rm acc}$ of a sink particle (set in the initial conditions or when the sink particle is created, see Section 2.8.4). First, a gas particle is indiscriminately accreted without undergoing any additional checks if it falls within $f_{\rm acc}r_{\rm acc}$, where $0 \leqslant f_{\rm acc} \leqslant 1$ (default $f_{\rm acc} = 0.8$). In the boundary region $f_{\rm acc}r_{\rm acc} < r < r_{\rm acc}$, a gas particle undergoes accretion if

1. $|\boldsymbol{L}_{ai}| < |\boldsymbol{L}_{\rm acc}|$, that is, its specific angular momentum is less than that of a Keplerian orbit at $r_{\rm acc}$,
2. $e = \frac{v_{ai}^2}{2} - \frac{GM_i}{r_{ai}} < 0$, i.e., it is gravitationally bound to the sink particle, and
3. e for this gas–sink pair is smaller than e with any other sink particle, that is, out of all sink particles, the gas particle is most bound to this one.

In the above conditions, $\boldsymbol{L}_{ai}$ is the relative specific angular momentum of the gas–sink pair, $a - i$, defined by

$$|\boldsymbol{L}_{ai}^2| \equiv |\boldsymbol{r}_{ai} \times \boldsymbol{v}_{ai}|^2 = r_{ai}^2 v_{ai}^2 - (\boldsymbol{r}_{ai} \cdot \boldsymbol{v}_{ai})^2\,, \tag{150}$$

while $|\boldsymbol{L}_{\rm acc}| = r_{\rm acc}^2\Omega_{\rm acc}$ is the angular momentum at $r_{\rm acc}$, where $\Omega_{\rm acc} = \sqrt{GM_i/r_{ai}^3}$ is the Keplerian angular speed at $r_{\rm acc}$, v_{ai}, and r_{ai} are the relative velocity and position, respectively, and M_i is the mass of the sink particle.

When a particle, a, passes the accretion checks, then the mass, position, velocity, acceleration, and spin angular momentum of the sink particle are updated according to

$$\boldsymbol{r}_i = \frac{(\boldsymbol{r}_a m_a + \boldsymbol{r}_i M_i)}{M_i + m_a}, \tag{151}$$

$$\boldsymbol{v}_i = \frac{(\boldsymbol{v}_a m_a + \boldsymbol{v}_i M_i)}{M_i + m_a}, \tag{152}$$

$$\boldsymbol{a}_i = \frac{(\boldsymbol{a}_a m_a + \boldsymbol{a}_i M_i)}{M_i + m_a}, \tag{153}$$

$$\boldsymbol{S}_i = \boldsymbol{S}_i + \frac{m_a M_i}{M_i + m_a}\left[(\boldsymbol{r}_a - \boldsymbol{r}_i) \times (\boldsymbol{v}_a - \boldsymbol{v}_i)\right], \tag{154}$$

$$M_i = M_i + m_a. \tag{155}$$

This ensures that mass, linear momentum, and angular momentum (but not energy) are conserved by the accretion process. The accreted mass as well as the total mass for each sink particle is stored to avoid problems with round-off error in the case where the particle masses are much smaller than the sink mass. Accreted particles are tagged by setting their smoothing lengths negative. Those particles with $h \leqslant 0$ are subsequently excluded when the kd-tree is built.

2.8.3. *Sink particle boundary conditions*

No special sink particle boundary conditions are implemented in PHANTOM at present. More advanced boundary conditions to regularise the density, pressure, and angular momentum near a sink have been proposed by Bate et al. (1995) and used in Bate & Bonnell (1997), and proposed again more recently by Hubber, Walch, & Whitworth (2013b). While these conditions help to regularise the flow near the sink particle, they can also cause problems—particularly the angular momentum boundary condition if the disc near the sink particle has complicated structure such as spiral density waves (Bate 2014, private communication). Often it is more cost effective to simply reduce the accretion radius of the sink. This may change in future code versions.

2.8.4. Dynamic sink particle creation

As described in Bate et al. (1995), it is also possible to create sink particles on-the-fly provided certain physical conditions are met and self-gravity is turned on (Section 2.12). The primary conditions required for sink particle formation are that the density of a given particle exceeds some threshold physical density somewhere in the domain, and that this density peak occurs more than a critical distance $r_{\rm crit}$ from an existing sink. Once these conditions are met on a particular particle, a, the creation of a new sink particle occurs by passing the following conditions (Bate et al. 1995):

1. The particle is a gas particle.
2. $\nabla\cdot\boldsymbol{v}_a \leq 0$, that is, gas surrounding the particle is at rest or collapsing.
3. $h_a < r_{\rm acc}/2$, i.e., the smoothing length of the particle is less than half of the accretion radius.
4. All neighbours within $r_{\rm acc}$ are currently active.
5. The ratio of thermal to gravitational energy of particles within $r_{\rm acc}$, $\alpha_{\rm J}$, satisfies $\alpha_J \leqslant 1/2$.
6. $\alpha_{\rm J} + \beta_{\rm rot} \leqslant 1$, where $\beta_{\rm rot} = |e_{\rm rot}|/|e_{\rm grav}|$ is the ratio of rotational energy to the magnitude of the gravitational energy for particles within $r_{\rm acc}$.
7. $e_{\rm tot} < 0$, that is, the total energy of particles within $r_{\rm acc}$ is negative (i.e. the clump is gravitationally bound).
8. The particle is at a local potential minimum, i.e. Φ is less than Φ computed on all other particles within $r_{\rm acc}$ (Federrath et al. 2010b).

A new sink particle is created at the position of particle a if these checks are passed, and immediately the particles within $r_{\rm acc}$ are accreted by calling the routine described in Section 2.8.2. The checks above are the same as those in Bate et al. (1995), with the additional check from Federrath et al. (2010b) to ensure that sink particles are only created in a local minimum of the gravitational potential.

The various energies used to evaluate the criteria above are computed according to

$$e_{\rm kin} = \frac{1}{2}\sum_{b=1}^{N<r_{\rm acc}} m_b(\boldsymbol{v}_b - \boldsymbol{v}_a)^2, \tag{156}$$

$$e_{\rm therm} = \sum_{b=1}^{N<r_{\rm acc}} m_b u_b, \tag{157}$$

$$e_{\rm grav} = -\frac{1}{2}\sum_{b=1}^{N<r_{\rm acc}}\sum_{c=b}^{N<r_{\rm acc}} G m_b m_c, \times\left[\phi(|\boldsymbol{r}_b - \boldsymbol{r}_c|, h_b) + \phi(|\boldsymbol{r}_b - \boldsymbol{r}_c|, h_c)\right], \tag{158}$$

$$e_{\rm tot} = e_{\rm kin} + e_{\rm therm} + e_{\rm grav}, \tag{159}$$

$$e_{\rm rot} \equiv \sqrt{e^2_{{\rm rot},x} + e^2_{{\rm rot},y} + e^2_{{\rm rot},z}}, \tag{160}$$

$$e_{{\rm rot},x} \equiv \frac{1}{2}\sum_{b=1}^{N<r_{\rm acc}} m_b \frac{L^2_{ab,x}}{\sqrt{(y_a - y_b)^2 + (z_a - z_b)^2}}, \tag{161}$$

$$e_{{\rm rot},y} \equiv \frac{1}{2}\sum_{b=1}^{N<r_{\rm acc}} m_b \frac{L^2_{ab,y}}{\sqrt{(x_a - x_b)^2 + (z_a - z_b)^2}}, \tag{162}$$

$$e_{{\rm rot},z} \equiv \frac{1}{2}\sum_{b=1}^{N<r_{\rm acc}} m_b \frac{L^2_{ab,z}}{\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}}, \tag{163}$$

where $\boldsymbol{L}_{ab} \equiv (\boldsymbol{r}_a - \boldsymbol{r}_b)\times(\boldsymbol{v}_a - \boldsymbol{v}_b)$ is the specific angular momentum between a pair of particles, and ϕ is the gravitational softening kernel (defined in Section 2.12), which has units of inverse length. Adding the contribution from *all* pairs, $b - c$, within the clump is required to obtain the total potential of the clump.

2.8.5. Sink particle timesteps

Sink particles are integrated together with a global, but adaptive, timestep, following the inner loop of the RESPA algorithm given in (80)–(83) corresponding to a second-order Leapfrog integration. The timestep is controlled by the minimum of the sink–gas timestep, (75), and a sink–sink timestep (Dehnen & Read 2011)

$$\Delta t_{\rm sink-sink} \equiv \min_i\left(C_{\rm force}\eta_\Phi\sqrt{\frac{|\Phi_i^{\rm sink-sink}|}{|\nabla\Phi_i^{\rm sink-sink}|^2}}\right), \tag{164}$$

where the potential and gradient include other sink particles, plus any external potentials applied to sink particles except the sink–gas potential. We set $\eta_\Phi = 0.05$ by default, resulting in ~300–500 steps per orbit for a binary orbit with the default $C_{\rm force} = 0.25$ (see Section 5.5.1).

More accurate integrators such as the fourth-order Hermite scheme (Makino & Aarseth 1992) or the fourth-order symplectic schemes proposed by Omelyan, Mryglod, & Folk (2002) or Chin & Chen (2005) are not yet implemented in

PHANTOM, but it would be a worthwhile effort to incorporate one of these in a future code version. See Hubber et al. (2013a) for a recent implementation of a fourth-order Hermite scheme for sink particles in SPH.

2.9. Stellar physics

A tabulated EOS can be used to take account of the departure from an ideal gas, for example, due to changes in ionisation or molecular dissociation and recombination. This tabulated EOS in PHANTOM is adapted from the $\log P_{\rm gas} - T$ EOS tables provided with the open source package Modules for Experiments in Stellar Astrophysics MESA (Paxton et al. 2011). Details of the data, originally compiled from blends of equations of state from Saumon, Chabrier, & van Horn (1995) (SCVH), Timmes & Swesty (2000), Rogers & Nayfonov (2002, also the 2005 update), Potekhin & Chabrier (2010) and for an ideal gas, are outlined by Paxton et al. (2011).

In our implementation (adapted from original routines for the MUSIC code; Goffrey et al. 2017), we compute the pressure and other required EOS variables for a particular composition by interpolation between sets of tables for different hydrogen abundance $X = 0.0, 0.2, 0.4, 0.6, 0.8$ and metallicity $Z = 0.0, 0.02, 0.04$. Pressure is calculated with bicubic interpolation, and $\Gamma_1 \equiv \partial \ln P/\partial \ln \rho|_s$ with bilinear interpolation, in $\log u$ and $\log V \equiv \log \rho - 0.7 \log u + 20$. The tables are currently valid in the ranges $10.5 \leqslant \log u \leqslant 17.5$ and $0.0 \leqslant \log V \leqslant 14.0$. Values requested outside the tables are currently computed by linear extrapolation. This triggers a warning to the user.

We have not tested the thermodynamic consistency of our interpolation scheme from the tables (Timmes & Arnett 1999).

2.10. Magnetohydrodynamics

PHANTOM implements the smoothed particle magnetohydrodynamics (SPMHD) algorithm described in Price (2012a) and Tricco & Price (2012, 2013), based on the original work by Phillips & Monaghan (1985) and Price & Monaghan (2004a, 2004b, 2005). PHANTOM was previously used to test a vector potential formulation (Price 2010), but this algorithm has been subsequently removed from the code due to its poor performance (see Price 2010).

The important difference between PHANTOM and the GADGET implementation of SPMHD (Dolag & Stasyszyn 2009; Bürzle et al. 2011a, 2011b), which also implements the core algorithms from Price & Monaghan (2004a, 2004b, 2005), is our use of the divergence-cleaning algorithm from Tricco & Price (2012, 2013) and Tricco, Price, & Bate (2016a). This is vital for preserving the divergence-free (no monopoles) condition on the magnetic field.

For recent applications of PHANTOM to MHD problems, see e.g. Tricco et al. (2016b), Dobbs et al. (2016), Bonnerot et al. (2017), Forgan, Price, & Bonnell (2017), and Wurster et al. (2016, 2017).

2.10.1. Equations of magnetohydrodynamics

PHANTOM solves the equations of MHD in the form

$$\frac{{\rm d}v^i}{{\rm d}t} = -\frac{1}{\rho}\frac{\partial M^{ij}}{\partial x^j} + \Pi_{\rm shock} + f^i_{\rm divB} + a^i_{\rm ext} + a^i_{\rm sink-gas} + a^i_{\rm selfgrav}, \tag{165}$$

$$\frac{{\rm d}u}{{\rm d}t} = -\frac{P}{\rho}(\nabla \cdot \boldsymbol{v}) + \Lambda_{\rm shock} - \frac{\Lambda_{\rm cool}}{\rho}, \tag{166}$$

$$\frac{{\rm d}}{{\rm d}t}\left(\frac{\boldsymbol{B}}{\rho}\right) = \frac{1}{\rho}\left[(\boldsymbol{B}\cdot\nabla)\boldsymbol{v} - \nabla\psi + \boldsymbol{D}_{\rm diss}\right], \tag{167}$$

$$\frac{{\rm d}}{{\rm d}t}\left(\frac{\psi}{c_{\rm h}}\right) = -c_{\rm h}(\nabla\cdot\boldsymbol{B}) - \frac{1}{2}\frac{\psi}{c_{\rm h}}(\nabla\cdot\boldsymbol{v}) - \frac{\psi}{c_{\rm h}\tau_{\rm c}}, \tag{168}$$

where $\boldsymbol{B}$ is the magnetic field, ψ is a scalar used to control the divergence error in the magnetic field (see Section 2.10.8), and $\mathcal{D}_{\rm diss}$ represents magnetic dissipation terms (Sections 2.10.5 and 2.11). The Maxwell stress tensor, M_{ij}, is given by

$$M^{ij} = \left(P + \frac{1}{2}\frac{B^2}{\mu_0}\right)\delta^{ij} - \frac{B^i B^j}{\mu_0}, \tag{169}$$

where δ^{ij} is the Kronecker delta and μ_0 is the permeability of free space. A source term related to the numerically induced divergence of the magnetic field, given by

$$f^i_{\rm divB} \equiv -\frac{B^i}{\rho}(\nabla\cdot\boldsymbol{B}) \tag{170}$$

is necessary to prevent the tensile instability in SPMHD (Phillips & Monaghan 1985; Monaghan 2000; Børve, Omang, & Trulsen 2001; Price 2012a). With this source term, the equation set for ideal MHD in the absence of the divergence cleaning field, ψ, is formally the same as in the Powell et al. (1999) eight-wave scheme (Price 2012a), meaning that the divergence errors in the magnetic field are advected by the flow, but not dissipated, unless cleaning is used.

2.10.2. Discrete equations

The discrete version of (165) follows the same procedure as for physical viscosity (Section 2.7), i.e.

$$\frac{{\rm d}v^i_a}{{\rm d}t} = -\sum_b m_b\left[\frac{M^{ij}_a}{\Omega_a\rho_a^2}\frac{\partial W_{ab}(h_a)}{\partial x^j_a} + \frac{M^{ij}_b}{\Omega_b\rho_b^2}\frac{\partial W_{ab}(h_b)}{\partial x^j_a}\right] + \Pi^a_{\rm shock} + f^i_{{\rm divB},a} + a^i_{{\rm ext},a} + a^i_{\rm sink-gas} + a^i_{\rm selfgrav}, \tag{171}$$

where M^{ij}_a is defined according to (169), $f_{\rm divB}$ is a correction term for stability (discussed below), and accelerations due to external forces are as described in Section 2.4.

Equations (167) and (168) are discretised according to (Price & Monaghan 2005; Tricco & Price 2012; Tricco et al. 2016a)

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\boldsymbol{B}}{\rho}\right)_a = &-\frac{1}{\Omega_a\rho_a^2}\sum_b m_b \boldsymbol{v}_{ab}\left[\boldsymbol{B}_a\cdot\nabla_a W_{ab}(h_a)\right]\\ &-\sum_b m_b\left[\frac{\psi_a}{\Omega_a\rho_a^2}\nabla_a W_{ab}(h_a)+\frac{\psi_b}{\Omega_b\rho_b^2}\nabla_a W_{ab}(h_b)\right]\\ &+\frac{1}{\rho_a}\mathcal{D}^a_{\rm diss},\end{aligned} \tag{172}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\psi}{c_{\rm h}}\right)_a = &\frac{c_{\rm h}^a}{\Omega_a\rho_a}\sum_b m_b\boldsymbol{B}_{ab}\cdot\nabla_a W_{ab}(h_a)\\ &+\frac{\psi_a}{2c_{\rm h}^a\Omega_a\rho_a}\sum_b m_b\boldsymbol{v}_{ab}\cdot\nabla_a W_{ab}(h_a)-\frac{\psi_a}{c_{\rm h}^a\tau_{\rm c}^a}.\end{aligned} \tag{173}$$

The first term in (173) uses the divergence of the magnetic field discretised according to

$$(\nabla\cdot\boldsymbol{B})_a = -\frac{1}{\Omega_a\rho_a}\sum_b m_b\left(\boldsymbol{B}_a-\boldsymbol{B}_b\right)\cdot\nabla_a W_{ab}(h_a), \tag{174}$$

which is therefore the operator we use when measuring the divergence error (c.f. Tricco & Price 2012).

2.10.3. Code units

An additional unit is required when magnetic fields are included to describe the unit of magnetic field. We adopt code units such that $\mu_0 = 1$, as is common practice. The unit scalings for the magnetic field can be determined from the definition of the Alfvén speed,

$$v_{\rm A} \equiv \sqrt{\frac{B^2}{\mu_0\rho}}. \tag{175}$$

Since the Alfvén speed has dimensions of length per unit time, this implies a unit for the magnetic field, $u_{\rm mag}$, given by

$$u_{\rm mag} = \left(\frac{\mu_0 u_{\rm mass}}{u_{\rm dist}u_{\rm time}}\right)^{\frac{1}{2}}. \tag{176}$$

Converting the magnetic field in the code to physical units therefore only requires specifying μ_0 in the relevant unit system. In particular, it avoids the differences between SI and cgs units in how the charge unit is defined, since μ_0 is dimensionless and equal to 4π in cgs units but has dimensions that involve charge in SI units.

2.10.4. Tensile instability correction

The correction term $f_{\rm divB}$ is necessary to avoid the tensile instability—a numerical instability where particles attract each other along field lines—in the regime where the magnetic pressure exceeds the gas pressure, that is, when plasma $\beta \equiv P/\frac{1}{2}B^2 < 1$ (Phillips & Monaghan 1985). The correction term is computed using the symmetric divergence operator (Børve et al. 2001; Price 2012a; Tricco & Price 2012)

$$\begin{aligned}f^i_{{\rm divB},a} = -\hat{B}^i_a\sum_b m_b&\left[\frac{\boldsymbol{B}_a\cdot\nabla_a W_{ab}(h_a)}{\Omega_a\rho_a^2}\right.\\ &\left.+\frac{\boldsymbol{B}_b\cdot\nabla_a W_{ab}(h_b)}{\Omega_b\rho_b^2}\right].\end{aligned} \tag{177}$$

Since this term violates momentum conservation to the extent that the $\nabla\cdot\boldsymbol{B}$ term is non-zero, several authors have proposed ways to minimise its effect. Børve et al. (2004) showed that stability could be achieved with $\hat{B}^i = \frac{1}{2}B^i$ and also proposed a scheme for scaling this term to zero for $\beta > 1$. Barnes, Kawata, & Wu (2012) similarly advocated using a factor of $\frac{1}{2}$ in this term. However, Tricco & Price (2012) showed that this could lead to problematic effects (their Figure 12). In Phantom, we use

$$\hat{B}^i = \begin{cases} B^i & \beta < 2,\\ [(10-\beta)B^i]/8 & 2 < \beta < 10,\\ 0 & \text{otherwise}\end{cases} \tag{178}$$

to provide numerical stability in the strong field regime while maintaining conservation of momentum when $\beta > 10$. This also helps to reduce errors in the MHD wave speed caused by the correction term (Iwasaki 2015).

2.10.5. Shock capturing

The shock capturing term in the momentum equation for MHD is identical to (39) and (40) except that the signal speed becomes (Price & Monaghan 2004a, 2005; Price 2012a)

$$v_{{\rm sig},a} = \alpha^{\rm AV}_a v_a + \beta^{\rm AV}|\boldsymbol{v}_{ab}\cdot\hat{\boldsymbol{r}}_{ab}|, \tag{179}$$

where

$$v_a = \sqrt{c_{{\rm s},a}^2 + v_{{\rm A},a}^2} \tag{180}$$

is the fast magnetosonic speed. Apart from this, the major difference to the hydrodynamic case is the addition of an artificial resistivity term to capture shocks and discontinuities in the magnetic field (i.e. current sheets). This is given by

$$\mathcal{D}^a_{\rm diss} = \frac{\rho_a}{2}\sum_b m_b\left[\frac{v^B_{{\rm sig},a}}{\rho_a^2}\frac{F_{ab}(h_a)}{\Omega_a}+\frac{v^B_{{\rm sig},b}}{\rho_b^2}\frac{F_{ab}(h_b)}{\Omega_b}\right]\boldsymbol{B}_{ab}, \tag{181}$$

where $v^B_{{\rm sig},a}$ is an appropriate signal speed (see below) multiplied by a dimensionless coefficient, α^B. The corresponding contribution to the thermal energy from the resistivity term in (42) is given by

$$\begin{aligned}\Lambda_{\rm artres} = -\frac{1}{4}\sum_b m_b&\left[\frac{v^B_{{\rm sig},a}}{\rho_a^2}\frac{F_{ab}(h_a)}{\Omega_a}\right.\\ &\left.+\frac{v^B_{{\rm sig},b}}{\rho_b^2}\frac{F_{ab}(h_b)}{\Omega_b}\right](\boldsymbol{B}_{ab})^2.\end{aligned} \tag{182}$$

As with the artificial viscosity, (181) and (182) are the SPH representation of a physical resistivity term, $\eta\nabla^2\boldsymbol{B}$, but with a coefficient that is proportional to resolution (Price & Monaghan 2004a). The resistive dissipation rate from the shock capturing term is given by

$$\eta \approx \frac{1}{2}\alpha^B v_{\rm sig}^B |r_{ab}|, \tag{183}$$

where $|r_{ab}| \propto h$.

2.10.6. Switch to reduce resistivity

PHANTOM previously used the method proposed by Tricco & Price (2013) to reduce the dissipation in the magnetic field away from shocks and discontinuities. The signal velocity, $v_{\rm sig}^B$, was set equal to the magnetosonic speed [Equation (180)] multiplied by the dimensionless coefficient α_B, which was set according to

$$\alpha_a^B = \min\left(h_a \frac{|\nabla \boldsymbol{B}_a|}{|\boldsymbol{B}_a|}, \alpha_{\rm max}^B\right), \tag{184}$$

where $\alpha_{\rm max}^B = 1.0$ by default and $|\nabla \boldsymbol{B}_a|$ is the two-norm of the gradient tensor, i.e. the root mean square of all nine components of this tensor. Unlike the viscous dissipation, this is set based on the instantaneous values of h and $\boldsymbol{B}$ and there is no source/decay equation involved, as Tricco & Price (2013) found it to be unnecessary. Since α^B is proportional to resolution, from (183), we see that this results in dissipation that is effectively second order ($\propto h^2$). When individual particle timesteps were used, inactive particles retained their value of α^B from the last timestep they were active.

More recently, we have found that a better approach, similar to that used for artificial conductivity, is to simply set $\alpha^{\rm B} = 1$ for all particles and set the signal speed for artificial resistivity according to

$$v_{\rm sig}^B = |\boldsymbol{v}_{ab} \times \hat{\boldsymbol{r}}_{ab}|. \tag{185}$$

We find that this produces better results on all of our tests (Section 5.6), in particular, producing zero numerical dissipation on the current loop advection test (Section 5.6.5). As with the Tricco & Price (2013) switch, it gives second-order dissipation in the magnetic field (demonstrated in Section 5.6.1, Figure 26). This is now the default treatment for artificial resistivity in PHANTOM.

2.10.7. Conservation properties

The total energy when using MHD is given by

$$E = \sum_a m_a \left(\frac{1}{2}v_a^2 + u_a + \Phi_a + \frac{1}{2}\frac{B_a^2}{\mu_0 \rho_a}\right). \tag{186}$$

Hence, total energy conservation, in the absence of divergence cleaning, corresponds to

$$\frac{{\rm d}E}{{\rm d}t} = \sum_a m_a \left[\boldsymbol{v}_a \cdot \frac{{\rm d}\boldsymbol{v}_a}{{\rm d}t} + \frac{{\rm d}u_a}{{\rm d}t} + \frac{{\rm d}\Phi_a}{{\rm d}t} + \frac{B_a^2}{2\mu_0\rho_a^2}\frac{{\rm d}\rho_a}{{\rm d}t} + \frac{\boldsymbol{B}_a}{\mu_0}\cdot\frac{\rm d}{{\rm d}t}\left(\frac{\boldsymbol{B}}{\rho}\right)_a\right] = 0. \tag{187}$$

Neglecting the $f_{\rm divB}$ correction term for the moment, substituting (171), (35), and (172) into (187) with the ideal MHD and shock capturing terms included demonstrates that the total energy is exactly conserved, using the same argument as the one given in Section 2.2.6 (detailed workings can be found in Price & Monaghan 2004b). The total linear momentum is also exactly conserved following a similar argument as in Section 2.2.11. However, the presence of the $f_{\rm divB}$ correction term, though necessary for numerical stability, violates the conservation of both momentum and energy in the strong field regime (in the weak field regime, it is switched off and conservation is preserved). The severity of this non-conservation is related to the degree in which divergence errors are present in the magnetic field, hence inadequate divergence control (see below) usually manifests as a loss of momentum conservation in the code (see Tricco & Price 2012, for details).

2.10.8. Divergence cleaning

We adopt the 'constrained' hyperbolic/parabolic divergence cleaning algorithm described by Tricco & Price (2012) and Tricco et al. (2016a) to preserve the divergence-free condition on the magnetic field. This formulation addresses a number of issues with earlier formulations by Dedner et al. (2002) and Price & Monaghan (2005).

The main idea of the scheme is to propagate divergence errors according to a damped wave equation (Dedner et al. 2002; Price & Monaghan 2005). This is facilitated by introducing a new scalar field, ψ, which is coupled to the magnetic field in (167) and evolved according to (168).

Tricco et al. (2016a) generalised the method of Dedner et al. (2002) to include the case where the hyperbolic wave speed, $c_{\rm h}$, varies in time and space. This is the approach we use in PHANTOM. The resulting 'generalised wave equation' may be derived by combining the relevant term in (167) with (168) to give (Tricco et al. 2016a)

$$\frac{\rm d}{{\rm d}t}\left[\frac{1}{\sqrt{\rho}c_{\rm h}}\frac{\rm d}{{\rm d}t}\left(\frac{\psi}{\sqrt{\rho}c_{\rm h}}\right)\right] - \frac{\nabla^2\psi}{\rho} + \frac{\rm d}{{\rm d}t}\left[\frac{1}{\sqrt{\rho}c_{\rm h}}\left(\frac{\psi}{\sqrt{\rho}c_{\rm h}\tau_{\rm c}}\right)\right] = 0. \tag{188}$$

When $c_{\rm h}$, ρ, τ_c, and the fluid velocity are constant, this reduces to the usual damped wave equation in the form

$$\frac{\partial^2\psi}{\partial t^2} - c_{\rm h}^2\nabla^2\psi + \frac{1}{\tau_{\rm c}}\frac{\partial\psi}{\partial t} = 0. \tag{189}$$

The same equation holds for the evolution of $\nabla\cdot\boldsymbol{B}$ itself, i.e.

$$\frac{\partial^2(\nabla\cdot\boldsymbol{B})}{\partial t^2} - c_{\rm h}^2\nabla^2(\nabla\cdot\boldsymbol{B}) + \frac{1}{\tau_{\rm c}}\frac{\partial(\nabla\cdot\boldsymbol{B})}{\partial t} = 0, \tag{190}$$

from which it is clear that $c_{\rm h}$ represents the speed at which divergence errors are propagated and $\tau_{\rm c}$ is the decay timescale over which divergence errors are removed.

Tricco & Price (2012) formulated a 'constrained' SPMHD implementation of hyperbolic/parabolic cleaning which guarantees numerical stability of the cleaning. The constraint imposed by Tricco & Price (2012) is that, in the absence of damping, any energy removed from the magnetic field during cleaning must be conserved by the scalar field, ψ. This enforces particular choices of numerical operators for $\nabla\cdot\boldsymbol{B}$ and $\nabla\psi$ in (172) and (173), respectively, in particular that they form a conjugate pair of difference and symmetric derivative operators. This guarantees that the change of magnetic energy is negative definite in the presence of the parabolic term (see below).

In PHANTOM, we set the cleaning speed, $c_{\rm h}$, equal to the fast magnetosonic speed [Equation (180)] so that its timestep constraint is satisfied already by (72), as recommended by Tricco & Price (2013). The decay timescale is set according to

$$\tau_{\rm c}^a \equiv \frac{h_a}{\sigma_{\rm c} c_{{\rm h},a}}, \tag{191}$$

where the dimensionless factor $\sigma_{\rm c}$ sets the ratio of parabolic to hyperbolic cleaning. This is set to $\sigma_{\rm c} = 1.0$ by default, which was empirically determined by Tricco & Price (2012) to provide optimal reduction of divergence errors in three dimensions.

The divergence cleaning dissipates energy from the magnetic field at a rate given by (Tricco & Price 2012)

$$\left(\frac{{\rm d}E}{{\rm d}t}\right)_{\rm cleaning} = -\sum_a m_a \frac{\psi_a^2}{\mu_0 \rho_a c_{{\rm h},a}^2 \tau_{\rm c}^a}. \tag{192}$$

In general, this is so small compared to other dissipation terms (e.g. resistivity for shock capturing) that it is not worth monitoring (Tricco et al. 2016a). This energy is not added as heat, but simply removed from the calculation.

2.10.9. *Monitoring of divergence errors and over-cleaning*

The divergence cleaning algorithm is guaranteed to either conserve or dissipate magnetic energy, and cleans divergence errors to a sufficient degree for most applications. However, the onus is on the user to ensure that divergence errors are not affecting simulation results. This may be monitored by the dimensionless quantity

$$\epsilon_{\rm divB} \equiv \frac{h|\nabla\cdot\boldsymbol{B}|}{|\boldsymbol{B}|}. \tag{193}$$

The maximum and mean values of this quantity should be used to check the fidelity of simulations that include magnetic fields. A good rule-of-thumb is that the mean should remain $\lesssim 10^{-2}$ for the simulation to remain qualitatively unaffected by divergence errors.

The cleaning wave speed can be arbitrarily increased to improve the effectiveness of the divergence cleaning according to

$$c_{{\rm h},a} = f_{\rm clean} v_a, \tag{194}$$

where $f_{\rm clean}$ is an 'over-cleaning' factor (by default, $f_{\rm clean} =$ 1, i.e. no 'over-cleaning'). Tricco et al. (2016a) showed that increasing $f_{\rm clean}$ leads to further reduction in divergence errors, without affecting the quality of obtained results, but with an accompanying computational expense associated with a reduction in the timestep size.

2.11. Non-ideal magnetohydrodynamics

PHANTOM implements non-ideal MHD including terms for Ohmic resistivity, ambipolar (ion-neutral) diffusion and the Hall effect. Algorithms and tests are taken from Wurster, Price, & Ayliffe (2014) and Wurster, Price, & Bate (2016). See Wurster et al. (2016, 2017) and Wurster, Bate, & Price (2018) for recent applications. Our formulation of non-ideal SPMHD in PHANTOM is simpler than the earlier formulation proposed by Hosking & Whitworth (2004) because we consider only one set of particles, representing a mixture of charged and uncharged species. Ours is similar to the implementation by Tsukamoto, Iwasaki, & Inutsuka (2013) and Tsukamoto et al. (2015).

2.11.1. *Equations of non-ideal MHD*

We assume the strong coupling or 'heavy ion' approximation (see e.g. Wardle & Ng 1999; Shu et al. 2006; Pandey & Wardle 2008), which neglects ion pressure and assumes $\rho_i \ll \rho_n$, where the subscripts i and n refer to the ionised and neutral fluids, respectively. In this case, (167) contains three additional terms in the form

$$\frac{\rm d}{{\rm d}t}\left(\frac{\boldsymbol{B}}{\rho}\right)_{\rm nimhd} = -\frac{1}{\rho}\nabla\times\left[\frac{\boldsymbol{J}}{\sigma_e} + \frac{\boldsymbol{J}\times\boldsymbol{B}}{en_e} - \frac{(\boldsymbol{J}\times\boldsymbol{B})\times\boldsymbol{B}}{\gamma_{\rm AD}\rho_i}\right], \tag{195}$$

where σ_e is the electrical conductivity, n_e is the number density of electrons, e is the charge on an electron, and $\gamma_{\rm AD}$ is the collisional coupling constant between ions and neutrals (Pandey & Wardle 2008). We write this in the form

$$\frac{\rm d}{{\rm d}t}\left(\frac{\boldsymbol{B}}{\rho}\right)_{\rm nimhd} = -\frac{1}{\rho}\nabla\times\Big[\eta_{\rm O}\boldsymbol{J} + \eta_{\rm H}\boldsymbol{J}\times\hat{\boldsymbol{B}} - \eta_{\rm AD}(\boldsymbol{J}\times\hat{\boldsymbol{B}})\times\hat{\boldsymbol{B}}\Big], \tag{196}$$

where $\hat{\boldsymbol{B}}$ is the unit vector in the direction of $\boldsymbol{B}$ such that $\eta_{\rm O}$, $\eta_{\rm AD}$, and $\eta_{\rm Hall}$ are the coefficients for resistive and ambipolar diffusion and the Hall effect, respectively, each with dimensions of area per unit time.

To conserve energy, we require the corresponding resistive and ambipolar heating terms in the thermal energy equation in the form

$$\left(\frac{{\rm d}u}{{\rm d}t}\right)_{\rm nimhd} = \frac{\eta_{\rm O}}{\rho}J^2 + \frac{\eta_{\rm AD}}{\rho}\left[J^2 - (\boldsymbol{J}\cdot\hat{\boldsymbol{B}})^2\right]. \tag{197}$$

The Hall term is non-dissipative, being dispersive rather than diffusive, so does not enter the energy equation.

We currently neglect the 'Biermann battery' term (Biermann 1950) proportional to $\nabla P_e/(en_e)$ in our non-ideal MHD implementation, both because it is thought to be negligible in the interstellar medium (Pandey & Wardle 2008) and because numerical implementations can produce incorrect results (Graziani et al. 2015). This term is mainly important in generating seed magnetic fields for subsequent dynamo processes (e.g. Khomenko et al. 2017).

2.11.2. Discrete equations

Our main constraint is that the numerical implementation of the non-ideal MHD terms should exactly conserve energy, which is achieved by discretising in the form (Wurster et al. 2014)

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\boldsymbol{B}}{\rho}\right)_{\mathrm{nimhd},a} = -\sum_b \left[\frac{\boldsymbol{D}_a}{\Omega_a \rho_a^2} \times \nabla_a W_{ab}(h_a) + \frac{\boldsymbol{D}_b}{\Omega_b \rho_b^2} \times \nabla_a W_{ab}(h_b)\right], \tag{198}$$

where

$$\boldsymbol{D} = -\eta_{\mathrm{O}}\boldsymbol{J} - \eta_{\mathrm{H}}(\boldsymbol{J} \times \hat{\boldsymbol{B}}) + \eta_{\mathrm{AD}}[(\boldsymbol{J} \times \hat{\boldsymbol{B}}) \times \hat{\boldsymbol{B}}]. \tag{199}$$

The corresponding term in the energy equation is given by

$$\left(\frac{\mathrm{d}u_a}{\mathrm{d}t}\right)_{\mathrm{nimhd}} = -\frac{\boldsymbol{D}_a \cdot \boldsymbol{J}_a}{\rho_a}, \tag{200}$$

where the magnetic current density is computed alongside the density evaluation according to

$$\boldsymbol{J} = \frac{1}{\Omega_a \rho_a} \sum_b m_b (\boldsymbol{B}_a - \boldsymbol{B}_b) \times \nabla_a W_{ab}(h_a). \tag{201}$$

Non-ideal MHD therefore utilises a ‘two first derivatives’ approach, similar to the formulation of physical viscosity described in Section 2.7.1. This differs from the ‘direct second derivatives’ approach used for our artificial resistivity term, and in previous formulations of physical resistivity (Bonafede et al. 2011). In practice, the differences are usually minor. Our main reason for using two first derivatives for non-ideal MHD is that it is easier to incorporate the Hall effect and ambipolar diffusion terms.

2.11.3. Computing the non-ideal MHD coefficients

To self-consistently compute the coefficients η_{O}, η_{H}, and η_{AD} from the local gas properties, we use the NICIL library (Wurster 2016) for cosmic ray ionisation chemistry and thermal ionisation. We refer the reader to Wurster (2016) and Wurster et al. (2016) for details, since this is maintained and versioned as a separate package.

2.11.4. Timestepping

With explicit timesteps, the timestep is constrained in a similar manner to other terms, using

$$\Delta t = \frac{C_{\mathrm{nimhd}} h^2}{\max(\eta_{\mathrm{O}}, \eta_{\mathrm{AD}}, |\eta_{\mathrm{H}}|)}, \tag{202}$$

where $C_{\mathrm{nimhd}} = 1/(2\pi)$ by default. This can prove prohibitive, so we employ the so-called ‘super-timestepping’ algorithm from Alexiades, Amiez, & Gremaud (1996) to relax the stability criterion for the Ohmic and ambipolar terms (only). The implementation is described in detail in Wurster et al. (2016). Currently, the Hall effect is always timestepped explicitly in the code.

2.12. Self-gravity

PHANTOM includes self-gravity between particles. By self-gravity, we mean a solution to Poisson’s equation

$$\nabla^2 \Phi = 4\pi G \rho(\boldsymbol{r}), \tag{203}$$

where Φ is the gravitational potential and ρ represents a continuous fluid density. The corresponding acceleration term in the equation of motion is

$$\boldsymbol{a}_{\mathrm{selfgrav}} = -\nabla\Phi. \tag{204}$$

Since (203) is an elliptic equation, implying instant action, it requires a global solution. This solution is obtained in PHANTOM by splitting the acceleration into ‘short-range’ and ‘long-range’ contributions.

$$\boldsymbol{a}^a_{\mathrm{selfgrav}} = \boldsymbol{a}^a_{\mathrm{short}} + \boldsymbol{a}^a_{\mathrm{long}}, \tag{205}$$

where the ‘short-range’ interaction is computed by direct summation over nearby particles, and the ‘long-range’ interaction is computed by hierarchical grouping of particles using the *k*d-tree.

The distance at which the gravitational acceleration is treated as ‘short-’ or ‘long-range’ is determined for each node–node pair, *n*–*m*, either by the tree opening criterion,

$$\theta^2 < \left(\frac{s_m}{r_{nm}}\right)^2, \tag{206}$$

where $0 \leqslant \theta \leqslant 1$ is the tree opening parameter, or by nodes whose smoothing spheres intersect,

$$r_{nm}^2 < \left[s_n + s_m + \max(R_{\mathrm{kern}} h^n_{\mathrm{max}}, R_{\mathrm{kern}} h^m_{\mathrm{max}})\right]^2. \tag{207}$$

Here, s is the node size, which is the minimum radius about the centre of mass that contains all the particles in the node, and h_{max} is the maximum smoothing length of the particles within the node. Node pairs satisfying either of these criteria have the particles contained within them added to a trial neighbour list, used for computing the short-range gravitational acceleration. Setting $\theta = 0$, therefore, leads to the gravitational acceleration computed entirely as ‘short-range’, that is, only via direction summation, while $\theta = 1$ gives the fastest possible, but least accurate, gravitational force evaluation. The code default is $\theta = 0.5$.

2.12.1. Short-range interaction

How to solve (203) on particles is one of the most widely misunderstood problems in astrophysics. In SPH or collisionless *N*-body simulations (i.e. stars and dark matter), the particles do not represent physical point-mass particles, but rather interpolation points in a density field that is assumed to be continuous. Hence, one needs to first reconstruct the density field on the particles, then solve (203) in a manner which is consistent with this (e.g. Hernquist & Barnes 1990).

How to do this consistently using a spatially adaptive softening length was demonstrated by Price & Monaghan (2007), since an obvious choice is to use the iterative kernel summation in (3) to both reconstruct the density field and set the

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

softening length, i.e.[6]

$$\rho_a = \sum_b m_b W_{ab}(\epsilon_a); \qquad \epsilon_a = h_{\rm fac}(m_a/\rho_a)^{1/3}. \tag{208}$$

It can then be shown that the gravitational potential consistent with this choice is given by

$$\Phi_a = -G\sum_b m_b \phi(|\boldsymbol{r}_a - \boldsymbol{r}_b|, \epsilon_a), \tag{209}$$

where ϕ is the softening kernel derived from the density kernel via Poisson's equation (Section 2.12.2). For a variable smoothing length, energy conservation necessitates taking an average of the kernels, i.e.

$$\Phi_a = -G\sum_b m_b \left[\frac{\phi_{ab}(\epsilon_a) + \phi_{ab}(\epsilon_b)}{2}\right]. \tag{210}$$

Price & Monaghan (2007) showed how the equations of motion could then be derived from a Lagrangian in order to take account of the softening length gradient terms, giving equations of motion in the form

$$\begin{aligned}\boldsymbol{a}^a_{\rm selfgrav} &= -\nabla\Phi_a, \\ &= -G\sum_b m_b \left[\frac{\phi'_{ab}(\epsilon_a) + \phi'_{ab}(\epsilon_b)}{2}\right]\hat{\boldsymbol{r}}_{ab} \\ &\quad - \frac{G}{2}\sum_b m_b \left[\frac{\zeta_a}{\Omega^\epsilon_a}\nabla_a W_{ab}(\epsilon_a) + \frac{\zeta_b}{\Omega^\epsilon_b}\nabla_a W_{ab}(\epsilon_b)\right],\end{aligned} \tag{211}$$

where Ω^ε and ζ are correction terms necessary for energy conservation, with Ω as in (5) but with h replaced by ε, and ζ defined as

$$\zeta_a \equiv \frac{\partial \epsilon_a}{\partial \rho_a}\sum_b m_b \frac{\partial \phi_{ab}(\epsilon_a)}{\partial \epsilon_a}. \tag{212}$$

The above formulation satisfies all of the conservation properties, namely conservation of linear momentum, angular momentum, and energy.

The short-range acceleration is evaluated for each particle in the leaf node n by summing (211) over the trial neighbour list obtained by node–node pairs that satisfy either of the criteria in (206) or (207). For particle pairs separated outside the softening radius of either particle, the short range interaction reduces to

$$\boldsymbol{a}^a_{{\rm short}, r > {\rm R_{kern}}\epsilon} = -G\sum_b m_b \frac{\boldsymbol{r}_a - \boldsymbol{r}_b}{|\boldsymbol{r}_a - \boldsymbol{r}_b|^3}. \tag{213}$$

We use this directly for such pairs to avoid unnecessary evaluations of the softening kernel.

It is natural in SPH to set the gravitational softening length equal to the smoothing length $\varepsilon = h$, since both derive from the same density estimate. Indeed, Bate & Burkert (1997) showed that this is a crucial requirement to resolve gas fragmentation correctly. For pure N-body calculations, Price & Monaghan (2007) also showed that setting the (variable) softening length in the same manner as the SPH smoothing length (Sections 2.1.3–2.1.4) results in a softening that is always close to the 'optimal' fixed softening length (Merritt 1996; Athanassoula et al. 2000; Dehnen 2001). In collisionless N-body simulations, this has been found to increase the resolving power, giving results at lower resolution comparable to those obtained at higher resolution with a fixed softening length (Bagla & Khandai 2009; Iannuzzi & Dolag 2011). It also avoids the problem of how to 'choose' the softening length, negating the need for 'rules of thumb' such as the one given by Springel (2005) where the softening length is chosen to be 1/40 of the average particle spacing in the initial conditions.

2.12.2. *Functional form of the softening kernel*

The density kernel and softening potential kernel are related using Poisson's equation (203), i.e.

$$W(r,\epsilon) = \frac{1}{4\pi r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial\phi}{\partial r}\right), \tag{214}$$

where $r \equiv |\boldsymbol{r} - \boldsymbol{r}'|$. Integrating this equation gives the softening kernels used for the force and gravitational potential. As with the standard kernel functions (Section 2.1.5), we define the potential and force kernels in terms of dimensionless functions of the scaled interparticle separation, $q \equiv r/h$, according to

$$\phi(r,\epsilon) \equiv \frac{1}{\epsilon}\tilde{\phi}(q), \tag{215}$$

$$\phi'(r,\epsilon) \equiv \frac{1}{\epsilon^2}\tilde{\phi}'(q), \tag{216}$$

where the dimensionless force kernel is obtained from the density kernel $f(q)$ (Section 2.1.5–2.1.6) using

$$\tilde{\phi}'(q) = \frac{4\pi}{q^2}C_{\rm norm}\int f(q')q'^2{\rm d}q', \tag{217}$$

with the integration constant set such that $\tilde{\phi}'(q) = 1/q^2$ at $q = R_{\rm kern}$. The potential function is

$$\tilde{\phi}(q) = \int \tilde{\phi}'(q')dq'. \tag{218}$$

The derivative of ϕ with respect to ε required in (212) is also written in terms of a dimensionless function, i.e.

$$\frac{\partial\phi(r,\epsilon)}{\partial\epsilon} \equiv \frac{1}{\epsilon^2}h(q), \tag{219}$$

where from differentiating (215), we have

$$h(q) = -\tilde{\phi}(q) - q\tilde{\phi}'(q). \tag{220}$$

Included in the PHANTOM source code is a PYTHON script using the SYMPY library to solve (217), (218), and (220) using symbolic integration to obtain the softening kernel from the density kernel. This makes it straightforward to obtain the otherwise laborious expressions needed for each kernel [expressions for the cubic spline kernel are in Appendix A of Price & Monaghan (2007) and for the quintic spline in Appendix A of Hubber et al. (2011)]. Figure 3 shows the functional form of the softening kernels for each of the kernels

[6] Strictly one should use the number density instead of the mass density when computing the softening length via (208), but as we enforce equal masses for each particle type in PHANTOM, the two methods are equivalent.

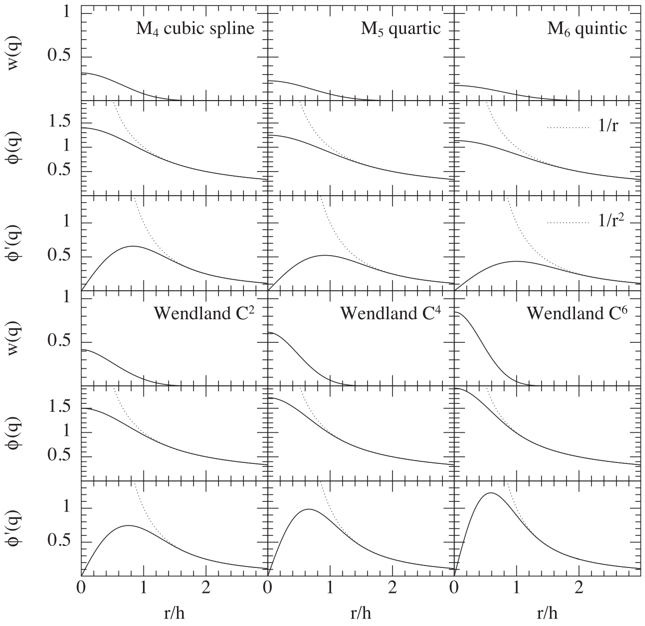

Figure 3. Functional form of the softening kernel functions $-\phi(r, h)$ and $\phi'(r, h)$ used to compute the gravitational force in PHANTOM, shown for each of the available kernel functions $w(r, h)$ (see Figure 1). Dotted lines show the functional form of the unsoftened potential $(-1/r)$ and force $(1/r^2)$ for comparison.

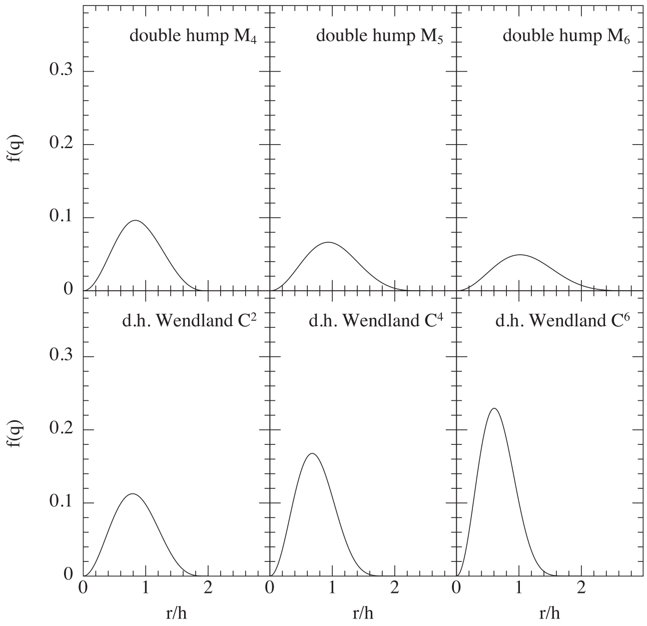

Figure 4. Double hump smoothing kernels $D(r, h)$ available in PHANTOM, used in the computation of the dust–gas drag force.

available in PHANTOM. The kernel function $f(q)$ is shown for comparison (top row in each case).

2.12.3. Softening of gravitational potential due to stars, dark matter, and dust

In the presence of collisionless components (e.g. stars, dark matter, and dust), we require estimates of the density in order to set the softening lengths for each component. We follow the generalisation of the SPH density estimate to multi-fluid systems described by Laibe & Price (2012a) where the density (and hence the softening length and Ω^{ϵ}) for each component is computed in the usual iterative manner (Section 2.1.4), but using only neighbours of the same type (c.f. Section 2.13.3). That is, the softening length for stars is determined based on the local density of stars, the softening length for dark matter is based on the local density of dark matter particles, and so on. The gravitational interaction both within each type and between different types is then computed using (211). This naturally symmetrises the softening between different types, ensuring that momentum, angular momentum, and energy are conserved.

2.12.4. Long-range interaction

At long range, that is $r > R_{\rm kern}\varepsilon_a$ and $r > R_{\rm kern}\varepsilon_b$, the second term in (211) tends to zero since $\zeta = 0$ for $q \geqslant R_{\rm kern}$, while the first term simply becomes $1/r^2$. Computing this via direct summation would have an associated $\mathcal{O}(N^2)$ computational cost, thus we adopt the usual practice of using the kd-tree to reduce this cost to $\mathcal{O}(N \log N)$.

The main optimisation in PHANTOM compared to a standard tree code (e.g. Hernquist & Katz 1989; Benz et al. 1990) is that we compute the long-range gravitational interaction once *per leaf-node* rather than once *per-particle* and that we use Cartesian rather than spherical multipole expansions to enable this (Dehnen 2000b; Gafton & Rosswog 2011).

The long-range acceleration on a given leaf node n consists of a sum over distant nodes m that satisfy neither (206) nor (207).

$$\boldsymbol{a}_n = \sum_m \boldsymbol{a}_{nm}. \tag{221}$$

The acceleration along the node centres, between a given pair n and m, is given (using index notation) by

$$a^i_{nm} = -\frac{GM_m}{r^3}r^i + \frac{1}{r^4}\left(\hat{r}^k Q^m_{ik} - \frac{5}{2}\hat{r}^i \mathcal{Q}^m\right), \tag{222}$$

where $r^i \equiv x^i_n - x^i_m$ is the relative position vector, $\hat{r}$ is the corresponding unit vector, M_m is the total mass in node m, Q^m_{ij} is the quadrupole moment of node m, and repeated indices imply summation. We define the following scalar and vector quantities for convenience:

$$\mathcal{Q} \equiv \hat{r}^i\hat{r}^j Q_{ij}, \tag{223}$$

$$\mathcal{Q}_i \equiv \hat{r}^j Q_{ij}. \tag{224}$$

Alongside the acceleration, we compute six independent components of the first derivative matrix:

$$\frac{\partial a^i_n}{\partial r^j} = \sum_m \frac{\partial a^i_{nm}}{\partial r^j}, \tag{225}$$

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

where

$$\frac{\partial a^i_{nm}}{\partial r^j} = \frac{GM_m}{r^3}\left[3\hat{r}^i\hat{r}^j - \delta^{ij}\right] + \frac{1}{r^5}\left[Q^m_{ij} + \left(\frac{35}{2}\hat{r}^i\hat{r}^j - \frac{5}{2}\delta_{ij}\right)Q^m - 5\hat{r}^i Q^m_j - 5\hat{r}^j Q^m_i\right], \quad (226)$$

and ten independent components of the second derivatives, given by

$$\frac{\partial^2 a^i_{nm}}{\partial r^j \partial r^k} = -\frac{3GM_m}{r^4}\left[5\hat{r}^i\hat{r}^j\hat{r}^k - \delta_{jk}\hat{r}^i - \delta_{ik}\hat{r}^j - \delta_{ij}\hat{r}^k\right] + \frac{1}{r^6}\left[-5(\hat{r}^k Q^m_{ij} + \hat{r}^i Q^m_{jk} + \hat{r}^j Q^m_{ik}) - \frac{315}{2}\hat{r}^i\hat{r}^j\hat{r}^k Q^m + \frac{35}{2}\left(\delta_{ij}\hat{r}^k + \delta_{ik}\hat{r}^j + \delta_{jk}\hat{r}^i\right)Q^m + 35\left(\hat{r}^j\hat{r}^k Q^m_i + \hat{r}^i\hat{r}^k Q^m_j + \hat{r}^i\hat{r}^j Q^m_k\right) - 5(\delta_{ij}Q^m_k + \delta_{ik}Q^m_j + \delta_{jk}Q^m_i)\right]. \quad (227)$$

The acceleration on each individual particle inside the leaf node n is then computed using a second-order Taylor series expansion of $\boldsymbol{a}^n_{\rm node}$ about the node centre, i.e.

$$a^i_{{\rm long},a} = a^i_n + \Delta x^j \frac{\partial a^i_n}{\partial r^j} + \frac{1}{2}\Delta x^j \Delta x^k \frac{\partial^2 a^i_n}{\partial r^j \partial r^k}, \quad (228)$$

where $\Delta x^i_a \equiv x^i_a - x^i_0$ is the relative distance of each particle from the node centre of mass. Pseudo-code for the resulting force evaluation is shown in Figure A5.

The quadrupole moments are computed during the tree build using

$$Q_{ij} = \sum_a m_a\left[3\Delta x^i \Delta x^j - (\Delta x)^2\delta^{ij}\right], \quad (229)$$

where the sum is over all particles in the node. Since Q is a symmetric tensor, only six independent quantities need to be stored (Q_{xx}, Q_{xy}, Q_{xz}, Q_{yy}, Q_{yz} and Q_{zz}).

The current implementation in PHANTOM is $\mathcal{O}(N_{\rm leafnodes}\log N_{\rm part})$ rather than the $\mathcal{O}(N)$ treecode implementation proposed by Dehnen (2000b), since we do not currently implement the symmetric node–node interactions required for $\mathcal{O}(N)$ scaling. Neither does our treecode conserve linear momentum to machine precision, except when $\theta = 0$. Implementing these additional features would be desirable.

2.13. Dust–gas mixtures

Modelling dust–gas mixtures is the first step in the ‘grand challenge’ of protoplanetary disc modelling (Haworth et al. 2016). The public version of PHANTOM implements dust–gas mixtures using two approaches. One models the dust and gas as two separate types of particles (two-fluid), as presented in Laibe & Price (2012a, 2012b), and the other, for small grains, using a single type of particle that represents the combination of dust and gas together (one-fluid), as described in Price & Laibe (2015a). Various combinations of these algorithms have been used in our recent papers using PHANTOM, including Dipierro et al. (2015, 2016), Ragusa et al. (2017), and Tricco et al. (2017) (see also Hutchison et al. 2016).

In the two-fluid implementation, the dust and gas are treated as two separate fluids coupled by a drag term with only explicit timestepping. In the one-fluid implementation, the dust is treated as part of the mixture, with an evolution equation for the dust fraction.

2.13.1. Continuum equations

The two-fluid formulation is based on the continuum equations in the form:

$$\frac{\partial \rho_{\rm g}}{\partial t} + (\boldsymbol{v}_{\rm g}\cdot\nabla)\rho_{\rm g} = -\rho_{\rm g}(\nabla\cdot\boldsymbol{v}_{\rm g}), \quad (230)$$

$$\frac{\partial \rho_{\rm d}}{\partial t} + (\boldsymbol{v}_{\rm d}\cdot\nabla)\rho_{\rm d} = -\rho_{\rm d}(\nabla\cdot\boldsymbol{v}_{\rm d}), \quad (231)$$

$$\frac{\partial \boldsymbol{v}_{\rm g}}{\partial t} + (\boldsymbol{v}_{\rm g}\cdot\nabla)\boldsymbol{v}_{\rm g} = -\frac{\nabla P}{\rho_{\rm g}} + \frac{K}{\rho_{\rm g}}(\boldsymbol{v}_{\rm d} - \boldsymbol{v}_{\rm g}), \quad (232)$$

$$\frac{\partial \boldsymbol{v}_{\rm d}}{\partial t} + (\boldsymbol{v}_{\rm d}\cdot\nabla)\boldsymbol{v}_{\rm d} = -\frac{K}{\rho_{\rm d}}(\boldsymbol{v}_{\rm d} - \boldsymbol{v}_{\rm g}), \quad (233)$$

$$\frac{\partial u}{\partial t} + (\boldsymbol{v}_{\rm g}\cdot\nabla)u = -\frac{P}{\rho_{\rm g}}(\nabla\cdot\boldsymbol{v}_{\rm g}) + \frac{\Lambda_{\rm drag}}{\rho_{\rm g}}, \quad (234)$$

where the subscripts g and d refer to gas and dust properties, K is a drag coefficient and the drag heating is

$$\Lambda_{\rm drag} \equiv K(\boldsymbol{v}_{\rm d} - \boldsymbol{v}_{\rm g})^2. \quad (235)$$

The implementation in PHANTOM currently neglects any thermal coupling between the dust and the gas (see Laibe & Price 2012a), aside from the drag heating. Thermal effects are important for smaller grains since they control the heating and cooling of the gas (e.g. Dopcke et al. 2011). Currently, the internal energy (u) of dust particles is simply set to zero.

2.13.2. Stopping time

The stopping time

$$t_{\rm s} = \frac{\rho_{\rm g}\rho_{\rm d}}{K(\rho_{\rm g} + \rho_{\rm d})} \quad (236)$$

is the characteristic timescale for the exponential decay of the differential velocity between the two phases caused by the drag. In the code, $t_{\rm s}$ is by default specified in physical units, which means that code units need to be defined appropriately when simulating dust–gas mixtures.

2.13.3. Two-fluid dust–gas in SPH

In the two-fluid method, the mixture is discretised into two distinct sets of ‘dust’ and ‘gas’ particles. In the following, we adopt the convention from Monaghan & Kocharyan (1995) that subscripts a, b, and c refer to gas particles while i, j, and k refer to dust particles. Hence, (230)–(231) are discretised with a density summation *over neighbours of the same type* (c.f. Section 2.12.3), giving

$$\rho_a = \sum_b m_b W_{ab}(h_a); \qquad h_a = h_{\rm fact}\left(\frac{m_a}{\rho_a}\right)^{1/3}, \tag{237}$$

for a gas particle, and

$$\rho_i = \sum_j m_j W_{ij}(h_i); \qquad h_i = h_{\rm fact}\left(\frac{m_i}{\rho_i}\right)^{1/3}, \tag{238}$$

for a dust particle. The kernel used for density is the same as usual (Section 2.1.6). We discretise the equations of motion for the gas particles, (232), using

$$\left(\frac{{\rm d}\boldsymbol{v}_a}{{\rm d}t}\right)_{\rm drag} = -3\sum_j m_j \frac{\boldsymbol{v}_{aj}\cdot\hat{\boldsymbol{r}}_{aj}}{(\rho_a+\rho_j)t^{\rm s}_{aj}}\hat{\boldsymbol{r}}_{aj}D_{aj}(h_a), \tag{239}$$

and for dust, (233), using

$$\left(\frac{{\rm d}\boldsymbol{v}_i}{{\rm d}t}\right)_{\rm drag} = -3\sum_b m_b \frac{\boldsymbol{v}_{ib}\cdot\hat{\boldsymbol{r}}_{ib}}{(\rho_i+\rho_b)t^{\rm s}_{ib}}\hat{\boldsymbol{r}}_{ib}D_{ib}(h_b), \tag{240}$$

where D is a ‘double hump’ kernel, defined in Section 2.13.4. The drag heating term in the energy equation, (234), is discretised using

$$\left(\frac{{\rm d}u}{{\rm d}t}\right)_{\rm drag} = \frac{\Lambda_{\rm drag}}{\rho_{\rm g}} = 3\sum_j m_j \frac{(\boldsymbol{v}_{aj}\cdot\hat{\boldsymbol{r}}_{aj})^2}{(\rho_a+\rho_j)t^{\rm s}_{aj}}D_{aj}(h_a). \tag{241}$$

Notice that gas properties are only defined on gas particles and dust properties are defined only on dust particles, greatly simplifying the algorithm. Buoyancy terms caused by dust particles occupying a finite volume (Monaghan & Kocharyan 1995; Laibe & Price 2012a) are negligible in astrophysics because typical grain sizes (μm) are negligible compared to simulation scales of ~au or larger.

2.13.4. Drag kernels

Importantly, we use a ‘double-hump’ shaped kernel function D (Fulk & Quinn 1996) instead of the usual bell-shaped kernel W when computing the drag terms. Defining D in terms of a dimensionless kernel function as previously (c.f. Section 2.1.5).

$$D(r,h) = \frac{\sigma}{h^3}g(q), \tag{242}$$

then the double hump kernels are defined from the usual kernels according to

$$g(q) = q^2 f(q), \tag{243}$$

where the normalisation constant σ is computed by enforcing the usual normalisation condition

$$\int D(r,h){\rm d}V = 1. \tag{244}$$

Figure 4 shows the functional forms of the double hump kernels used in Phantom. Using double hump kernels for the drag terms was found by Laibe & Price (2012a) to give a factor of 10 better accuracy at no additional cost. The key feature is that these kernels are zero at the origin putting more weight in the outer parts of the kernel where the velocity difference is large. This also solves the problem of how to define the unit vector in the drag terms (239)–(241)—it does not matter since D is also zero at the origin.

2.13.5. Stopping time in SPH

The stopping time is defined between a pair of particles, using the properties of gas and dust defined on the particle of the respective type, i.e.

$$t^{\rm s}_{aj} = \frac{\rho_a\rho_j}{K_{aj}(\rho_a+\rho_j)}. \tag{245}$$

The default prescription for the stopping time in Phantom automatically selects a physical drag regime appropriate to the grain size, as described below and in Laibe & Price (2012b). Options to use either a constant K or a constant $t_{\rm s}$ between pairs are also implemented, useful for testing and debugging (c.f. Section 5.9).

2.13.6. Epstein drag

To determine the appropriate physical drag regime, we use the procedure suggested by Stepinski & Valageas (1996) where we first evaluate the Knudsen number

$$K_{\rm n} = \frac{9\lambda_{\rm g}}{4s_{\rm grain}}, \tag{246}$$

where $s_{\rm grain}$ is the grain size and $\lambda_{\rm g}$ is the gas mean free path (see Section 2.13.8, below, for how this is evaluated). For $K_n \geqslant 1$, the drag between a particle pair is computed using the generalised formula for Epstein drag from Kwok (1975), as described in Paardekooper & Mellema (2006) and Laibe & Price (2012b), giving

$$K_{aj} = \rho^a_{\rm g}\rho^j_{\rm d}\frac{4}{3}\sqrt{\frac{8\pi}{\gamma}}\frac{s^2_{\rm grain}}{m_{\rm grain}}c^a_{\rm s}f, \tag{247}$$

where

$$m_{\rm grain} \equiv \frac{4}{3}\pi\rho_{\rm grain}s^3_{\rm grain}, \tag{248}$$

and $\rho_{\rm grain}$ is the intrinsic grain density, which is 3 g/cm^3 by default. The variable f is a correction for supersonic drift velocities given by (Kwok 1975)

$$f \equiv \sqrt{1+\frac{9\pi}{128}\frac{\Delta v^2}{c^2_{\rm s}}}, \tag{249}$$

where $\Delta v \equiv |\boldsymbol{v}_{\rm d}-\boldsymbol{v}_{\rm g}| = v^j_{\rm d}-v^a_{\rm g}$. The stopping time is therefore

$$t_{\rm s} = \frac{\rho_{\rm grain}s_{\rm grain}}{\rho c_{\rm s}f}\sqrt{\frac{\pi\gamma}{8}}, \tag{250}$$

where $\rho \equiv \rho_{\rm d}+\rho_{\rm g}$. This formula, (250), reduces to the standard expression for the linear Epstein regime in the limit

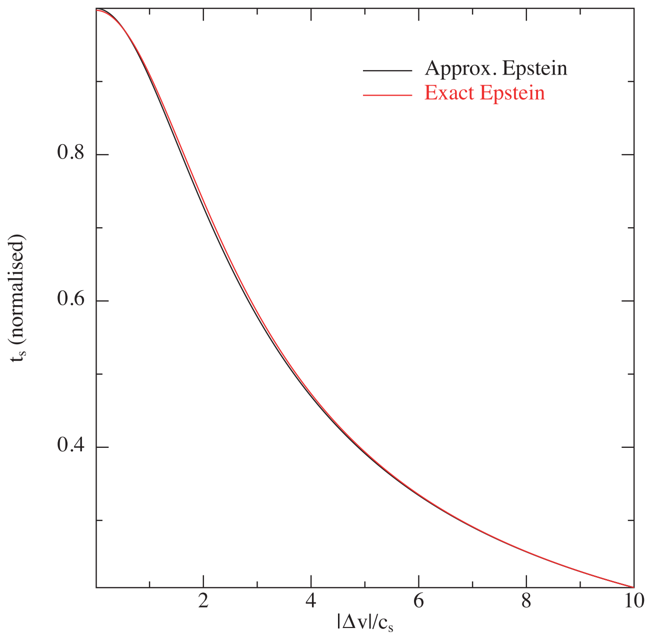

Figure 5. Dependence of the drag stopping time t_s on differential Mach number, showing the increased drag (decrease in stopping time) as the velocity difference between dust and gas increases. The black line shows the analytic approximation we employ [Equation (250)] which may be compared to the red line showing the exact expression from Epstein (1924). The difference is less than 1% everywhere.

where the drift velocity is small compared to the sound speed (i.e. $f \to 1$). Figure 5 shows the difference between the above simplified prescription and the exact expression for Epstein drag (Epstein 1924; c.f. equations (11) and (38) in Laibe & Price 2012b) as a function of $\Delta v/c_s$, which is less than 1% everywhere.

2.13.7. Stokes drag

For $K_n < 1$, we adopt a Stokes drag prescription, describing the drag on a sphere with size larger than the mean free path of the gas (Fan & Zhu 1998). Here, we use (Laibe & Price 2012b)

$$K_{aj} = \rho_g^a \rho_d^j \frac{1}{2} C_D \frac{\pi s_{\rm grain}^2}{m_{\rm grain}} |\Delta v|, \tag{251}$$

where the coefficient C_D is given by (Fassio & Probstein 1970) (see Whipple 1972; Weidenschilling 1977)

$$C_D = \begin{cases} 24 R_e^{-1}, & R_e < 1, \\ 24 R_e^{-0.6}, & 1 < R_e < 800, \\ 0.44, & R_e > 800, \end{cases} \tag{252}$$

where R_e is the local Reynolds number around the grain

$$R_e \equiv \frac{2 s_{\rm grain} |\Delta v|}{\nu}, \tag{253}$$

and ν is the microscopic viscosity of the gas (see below, not to be confused with the disc viscosity). Similar formulations of Stokes drag can be found elsewhere (see e.g. discussion in Woitke & Helling 2003 and references therein). The stopping time in the Stokes regime is therefore given by

$$t_s = \frac{8 \rho_{\rm grain} s_{\rm grain}}{3 \rho |\Delta v| C_D}, \tag{254}$$

where it remains to evaluate ν and λ_g.

2.13.8. Kinematic viscosity and mean free path

We evaluate the microscopic kinematic viscosity, ν, assuming gas molecules interact as hard spheres, following Chapman & Cowling (1970). The viscosity is computed from the mean free path and sound speed according to

$$\nu = \sqrt{\frac{2}{\pi \gamma}} c_s \lambda_g, \tag{255}$$

with the mean free path defined by relating this expression to the expression for the dynamic viscosity of the gas (Chapman & Cowling 1970) given by

$$\mu_\nu = \frac{5m}{64 \sigma_s} \sqrt{\frac{\pi}{\gamma}} c_s, \tag{256}$$

with $\mu_\nu = \rho_g \nu$, giving

$$\lambda_g = \frac{5\pi}{64\sqrt{2}} \frac{1}{n_g \sigma_s}, \tag{257}$$

where $n_g = \rho_g/m$ is the number density of molecules and σ_s is the collisional cross section. To compute this, Phantom currently assumes the gas is molecular Hydrogen, such that the mass of each molecule and the collisional cross section are given by

$$m = 2 m_H, \tag{258}$$

$$\sigma_s = 2.367 \times 10^{-15} {\rm cm}^2. \tag{259}$$

2.13.9. Stokes/Epstein transition

At the transition between the two regimes, assuming $R_e < 1$, (254) reduces to

$$t_s = \frac{2 \rho_{\rm grain} s_{\rm grain}^2}{9 \rho c_s \lambda_g} \sqrt{\frac{\pi \gamma}{2}}, \tag{260}$$

which is the same as the Epstein drag in the subsonic regime when $\lambda_g = 4 s_{\rm grain}/9$, i.e. $K_n = 1$. That this transition is indeed continuous in the code is demonstrated in Figure 6, which shows the transition from Epstein to Stokes drag and also through each of the Stokes regimes in (252) by varying the grain size while keeping the other parameters fixed. For simplicity, we assumed a fixed $\Delta v = 0.01 c_s$ in this plot, even though in general one would expect Δv to increase with stopping time [see Equation (270)].

2.13.10. Self-gravity of dust

With self-gravity turned on, dust particles interact in the same way as stars or dark matter (Section 2.12.3), with a softening length equal to the smoothing length determined from the density of neighbouring dust particles. Dust particles can be accreted by sink particles (Section 2.8.2), but a sink cannot

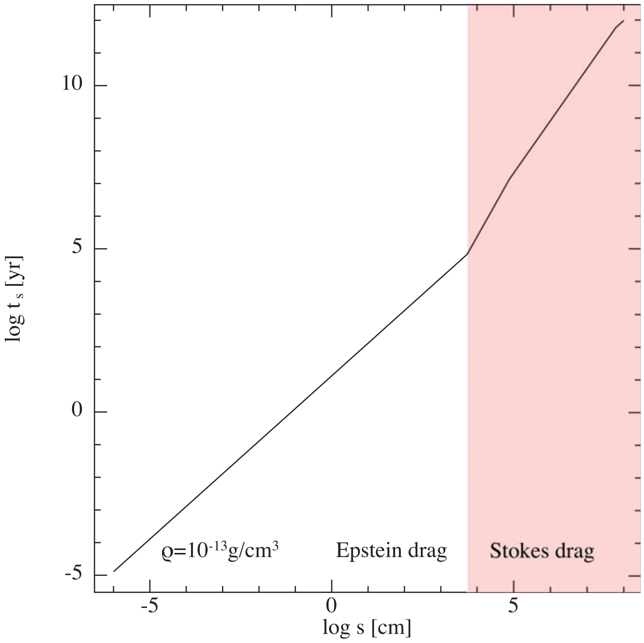

Figure 6. Drag stopping time t_s (in years) as a function of grain size, showing the continuous transition between the Epstein and Stokes drag regimes. The example shown assumes fixed density $\rho = 10^{-13}\mathrm{g/cm}^3$ and sound speed $c_s = 6 \times 10^4$ cm/s with subsonic drag $\Delta v = 0.01 c_s$ and material density $\rho_{\rm grain} = 3\mathrm{g/cm}^3$.

currently be created from dust particles (Section 2.8.4). There is currently no mechanism in the code to handle the collapse of dust to form a self-gravitating object independent of the gas.

2.13.11. *Timestep constraint*

For the two-fluid method, the timestep is constrained by the stopping time according to

$$\Delta t^a_{\rm drag} = \min_j (t_s^{aj}). \tag{261}$$

This requirement, alongside the spatial resolution requirement $h \lesssim c_s t_s$ (Laibe & Price 2012a), means the two-fluid method becomes both prohibitively slow and increasingly inaccurate for small grains. In this regime, one should switch to the one-fluid method, as described below.

2.13.12. *Continuum equations: One-fluid*

In Laibe & Price (2014a), we showed that the two-fluid equations, (230)–(234), can be rewritten as a single fluid mixture using a change of variables given by

$$\rho \equiv \rho_{\rm g} + \rho_{\rm d}, \tag{262}$$

$$\epsilon \equiv \rho_{\rm d}/\rho, \tag{263}$$

$$\boldsymbol{v} \equiv \frac{\rho_{\rm g}\boldsymbol{v}_{\rm g} + \rho_{\rm d}\boldsymbol{v}_{\rm d}}{\rho_{\rm g} + \rho_{\rm d}}, \tag{264}$$

$$\Delta\boldsymbol{v} \equiv \boldsymbol{v}_{\rm d} - \boldsymbol{v}_{\rm g}, \tag{265}$$

where ρ is the combined density, ε is the dust fraction, $\boldsymbol{v}$ is the barycentric velocity, and $\Delta\boldsymbol{v}$ is the differential velocity between the dust and gas.

In Laibe & Price (2014a), we derived the full set of evolution equations in these variables, and in Laibe & Price (2014c), implemented and tested an algorithm to solve these equations in SPH. However, using a fluid approximation cannot properly capture the velocity dispersion of large grains, as occurs for example when large planetesimals stream simultaneously in both directions through the midplane of a protoplanetary disc. For this reason, the one-fluid equations are better suited to treating small grains, where the stopping time is shorter than the computational timestep. In this limit, we employ the 'terminal velocity approximation' (e.g. Youdin & Goodman 2005) and the evolution equations reduce to (Laibe & Price 2014a; Price & Laibe 2015a)

$$\frac{{\rm d}\rho}{{\rm d}t} = -\rho(\nabla\cdot\boldsymbol{v}), \tag{266}$$

$$\frac{{\rm d}\boldsymbol{v}}{{\rm d}t} = -\frac{\nabla P}{\rho} + \boldsymbol{a}_{\rm ext}, \tag{267}$$

$$\frac{{\rm d}\epsilon}{{\rm d}t} = -\frac{1}{\rho}\nabla\cdot\left[\epsilon(1-\epsilon)\rho\Delta\boldsymbol{v}\right], \tag{268}$$

$$\frac{{\rm d}u}{{\rm d}t} = -\frac{P}{\rho}(\nabla\cdot\boldsymbol{v}) + \epsilon(\Delta\boldsymbol{v}\cdot\nabla)u, \tag{269}$$

where

$$\Delta\boldsymbol{v} \equiv t_s\left(\boldsymbol{a}_{\rm d} - \boldsymbol{a}_{\rm g}\right), \tag{270}$$

where $\boldsymbol{a}_{\rm d}$ and $\boldsymbol{a}_{\rm g}$ refers to any acceleration acting only on the dust or gas phase, respectively. For the simple case of pure hydrodynamics, the only difference is the pressure gradient, giving

$$\Delta\boldsymbol{v} \equiv t_s\frac{\nabla P}{\rho_{\rm g}} = \frac{t_s}{(1-\epsilon)}\frac{\nabla P}{\rho}, \tag{271}$$

such that (268) becomes

$$\frac{{\rm d}\epsilon}{{\rm d}t} = -\frac{1}{\rho}\nabla\cdot\left(\epsilon t_s\nabla P\right). \tag{272}$$

Importantly, the one-fluid dust algorithm does not result in any heating term in du/dt due to drag, because this term is $\mathcal{O}(\Delta\boldsymbol{v}^2)$ and thus negligible (Laibe & Price 2014a).

2.13.13. *Visualisation of one-fluid results*

Finally, when visualising results of one-fluid calculations, one must reconstruct the properties of the dust and gas in post-processing. We use

$$\rho_{\rm g} = (1-\epsilon)\rho, \tag{273}$$

$$\rho_{\rm d} = \epsilon\rho, \tag{274}$$

$$\boldsymbol{v}_{\rm g} = \boldsymbol{v} - \epsilon\Delta\boldsymbol{v}, \tag{275}$$

$$\boldsymbol{v}_{\rm d} = \boldsymbol{v} + (1-\epsilon)\Delta\boldsymbol{v}. \tag{276}$$

To visualise the one-fluid results in a similar manner to those from the two-fluid method, we reconstruct a set of 'dust' and 'gas' particles with the same positions but with the respective properties of each type of fluid particle copied onto them. We compute $\Delta \boldsymbol{v}$ from (271) using the pressure gradient computed using (34), multiplied by the stopping time and the gas fraction. See Price & Laibe (2015a) for more details.

2.13.14. One-fluid dust–gas implementation

Our implementation of the one-fluid method in PHANTOM follows Price & Laibe (2015a) with a few minor changes and corrections[7]. In particular, we use the variable $s = \sqrt{\epsilon\rho}$ described in Appendix B of Price & Laibe (2015a) to avoid problems with negative dust fractions. The evolution equation (268) expressed in terms of s is given by

$$\frac{{\rm d}s}{{\rm d}t} = -\frac{1}{2s}\nabla\cdot\left(\frac{\rho_{\rm g}\rho_{\rm d}}{\rho}\Delta\boldsymbol{v}\right) - \frac{s}{2}(\nabla\cdot\boldsymbol{v}), \tag{277}$$

which for the case of hydrodynamics becomes

$$\begin{aligned}\frac{{\rm d}s}{{\rm d}t} &= -\frac{1}{2s}\nabla\cdot\left(\frac{s^2}{\rho}t_{\rm s}\nabla P\right) - \frac{s}{2}(\nabla\cdot\boldsymbol{v}),\\ &= -\frac{1}{2}\nabla\cdot\left(\frac{s}{\rho}t_{\rm s}\nabla P\right) - \frac{t_{\rm s}}{2\rho}\nabla P\cdot\nabla s - \frac{s}{2}(\nabla\cdot\boldsymbol{v}).\end{aligned} \tag{278}$$

The SPH discretisation of this equation is implemented in the form

$$\begin{aligned}\frac{{\rm d}s_a}{{\rm d}t} &= -\frac{1}{2}\sum_b \frac{m_b s_b}{\rho_b}\left(\frac{t_{{\rm s},a}}{\rho_a} + \frac{t_{{\rm s},b}}{\rho_b}\right)(P_a - P_b)\frac{\overline{F}_{ab}}{|r_{ab}|}\\ &\quad + \frac{s_a}{2\rho_a\Omega_a}\sum_b m_b \boldsymbol{v}_{ab}\cdot\nabla_a W_{ab}(h_a),\end{aligned} \tag{279}$$

where $\overline{F}_{ab} \equiv \frac{1}{2}[F_{ab}(h_a) + F_{ab}(h_b)]$. The thermal energy equation, (269), takes the form

$$\frac{{\rm d}u}{{\rm d}t} = -\frac{P}{\rho}(\nabla\cdot\boldsymbol{v}) + \frac{s^2 t_{\rm s}}{\rho\rho_{\rm g}}\nabla P\cdot\nabla u, \tag{280}$$

the first term of which is identical to (35) and the second term of which is discretised in PHANTOM according to

$$-\frac{\rho_a}{2\rho_a^{\rm g}}\sum_b m_b\frac{s_a s_b}{\rho_a\rho_b}\left(\frac{t_{{\rm s},a}}{\rho_a} + \frac{t_{{\rm s},b}}{\rho_b}\right)(P_a - P_b)(u_a - u_b)\frac{\overline{F}_{ab}}{|r_{ab}|}. \tag{281}$$

2.13.15. Conservation of dust mass

Conservation of dust mass with the one-fluid scheme is in principle exact because (Price & Laibe 2015a)

$$\frac{\rm d}{{\rm d}t}\left(\sum_a \frac{m_a}{\rho_a}s_a^2\right) = \sum_a m_a\left(\frac{2s_a}{\rho_a}\frac{{\rm d}s_a}{{\rm d}t} - \frac{s_a^2}{\rho_a^2}\frac{{\rm d}\rho_a}{{\rm d}t}\right) = 0. \tag{282}$$

In practice, some violation of this can occur because although the above algorithm guarantees positivity of the dust fraction, it does not guarantee that ε remains less than unity. Under this circumstance, which occurs only rarely, we set $\varepsilon = \max(\varepsilon, 1)$ in the code. However, this violates the conservation of dust mass. This specific issue has been recently addressed in detail in the study by Ballabio et al. (2018). Therefore, in the latest code, there are two main changes as follows:

- Rather than evolve $s = \sqrt{\epsilon\rho}$, in the most recent code, we instead evolve a new variable $s' = \sqrt{\rho_{\rm d}/\rho_{\rm g}}$. This prevents the possibility of $\varepsilon > 1$.
- We limit the stopping time such that the timestep from the one fluid algorithm does not severely restrict the computational performance.

For details of these changes, we refer the reader to Ballabio et al. (2018).

2.13.16. Conservation of energy and momentum

Total energy with the one-fluid scheme can be expressed via

$$E = \sum_a m_a\left[\frac{1}{2}v_a^2 + (1 - \epsilon_a)u_a\right], \tag{283}$$

which is conserved exactly by the discretisation since

$$\begin{aligned}\sum_a m_a\Bigg[\boldsymbol{v}_a\cdot\frac{{\rm d}\boldsymbol{v}_a}{{\rm d}t} + (1 - \epsilon_a)\frac{{\rm d}u_a}{{\rm d}t}\\ - u_a\left(\frac{2s_a}{\rho_a}\frac{{\rm d}s_a}{{\rm d}t} - \frac{s_a^2}{\rho_a^2}\frac{{\rm d}\rho_a}{{\rm d}t}\right)\Bigg] = 0.\end{aligned} \tag{284}$$

Conservation of linear and angular momentum also hold since the discretisation of the momentum equation is identical to the hydrodynamics algorithm.

2.13.17. Timestep constraint

For the one-fluid method, the timestep is constrained by the *inverse* of the stopping time according to

$$\Delta t^a_{\rm drag} = C_{\rm force}\frac{h^2}{\epsilon t_{\rm s} c_{\rm s}^2}. \tag{285}$$

This becomes prohibitive precisely in the regime where the one-fluid method is no longer applicable ($t_{\rm s} > t_{\rm Cour}$; see Laibe & Price 2014a), in which case one should switch to the two-fluid method instead. There is currently no mechanism to automatically switch regimes, though this is possible in principle and may be implemented in a future code version.

2.14. Interstellar medium (ISM) physics

2.14.1. Cooling function

The cooling function in PHANTOM is based on a set of FORTRAN modules written by Glover & Mac Low (2007), updated further by Glover et al. (2010). It includes cooling from atomic lines (H I), molecular excitation (H_2), fine structure metal lines (Si I, Si II, O I, C I, and C II), gas–dust collisions, and polycyclic aromatic hydrocarbon (PAH) recombination (see Glover & Mac Low 2007 for references on each process).

[7] One should be aware that we derived several of the above equations incorrectly in Appendix B of Price & Laibe (2015a) (see Price & Laibe 2015b). The above equations are the correct versions and reflect what is implemented in PHANTOM.

Table 2. Heating and cooling processes in the PHANTOM cooling module.

Process	Description	Reference
H I (atomic) cooling	Electron collisional excitation/ resonance line emission	Sutherland & Dopita (1993)
H_2 (molecular) cooling	Vibrational/rotational excitation cooling by collisions with H, He and H_2	Le Bourlot et al. (1999)
Fine structure cooling	C II, Si II, and O I collisions with H, H_2, free e^-, and H^+	Glover & Jappsen (2007)
CO rotational cooling	CO collisions with H_2, H, and free e^-	Neufeld & Kaufman (1993), Neufeld, Lepp, & Melnick (1995)
Recombination cooling	Free e^- recombining with ionised gas on PAH and dust grain surfaces	Wolfire et al. (2003)
Gas-grain cooling	Dust–gas collisional heat transfer	Hollenbach & McKee (1989)
Cosmic-ray heating	Energy deposition associated with cosmic ray ionisation	Goldsmith & Langer (1978)
Photo-electric heating	UV e^- excitation from dust and PAH	Wolfire et al. (2003)

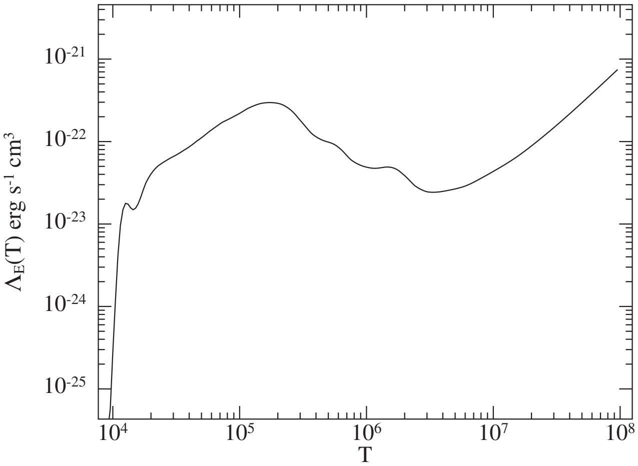

Figure 7. Emissivity, Λ_E (erg s^{-1} cm^3) as a function of temperature for the ISM cooling assuming default abundances appropriate for the warm neutral medium (WNM). Note that as we treat cooling from atomic hydrogen using a full non-equilibrium treatment, the behaviour of Λ_E close to 10^4 K is highly sensitive to the electron fraction, which in the case shown here is much smaller than it would be in collisional ionisation equilibrium. Values of Λ_E below 10^4 K depend strongly on the current chemical state of the gas and are not shown in this plot.

Heating is provided by cosmic rays and the photoelectric effect. The strength of the cooling is a function of the temperature, density, and various chemical abundances. Table 2 summarises these various heating and cooling processes. Figure 7 shows the resulting emissivity $\Lambda_E(T)$ for temperatures between 10^4 and 10^8 K. The cooling rate per unit volume is related to the emissivity according to

$$\Lambda_{\rm cool} \equiv n^2 \Lambda_E(T)\, {\rm erg\ s^{-1} cm^{-3}}, \tag{286}$$

where n is the number density. The cooling in the energy equation corresponds to

$$\left(\frac{{\rm d}u}{{\rm d}t}\right)_{\rm cool} = -\frac{\Lambda_{\rm cool}}{\rho}. \tag{287}$$

Table 3. Default fractional abundances for C, O, Si, and e^- in the ISM cooling and chemistry modules. Abundances are taken from Sembach et al. (2000) appropriate for the warm neutral medium (WNM). These are lower than solar because it is assumed some fraction of the metals are locked up in dust rather than being available in the gas phase.

Element	Abundance
C	1.4×10^{-4}
O	3.2×10^{-4}
Si	1.5×10^{-5}
e^-	2×10^{-4}

These routines were originally adapted for use in SPHNG (Dobbs et al. 2008) and result in an approximate two-phase ISM with temperatures of 100 K and 10 000 K. Note that the cooling depends on a range of parameters (dust-to-gas ratio, cosmic ray ionisation rate, etc.), many of which can be specified at runtime. Table 3 lists the default abundances, which are set to values appropriate for the Warm Neutral Medium taken from Sembach et al. (2000). The abundances listed in Table 3, along with the dust-to-gas ratio, are the only metallicity-dependent parameters that can be changed at runtime. An option for cooling appropriate to the zero metallicity early universe is also available.

2.14.2. *Timestep constraint from cooling*

When cooling is used, we apply an additional timestep constraint in the form

$$\Delta t^a_{\rm cool} = C_{\rm cool}\left|\frac{u}{({\rm d}u/{\rm d}t)_{\rm cool}}\right|, \tag{288}$$

where $C_{\rm cool} = 0.3$ following Glover & Mac Low (2007). The motivation for this additional constraint is to not allow the

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

Table 4. Processes and references for the Phantom ISM chemistry module tracing the evolution of H, H_2, and CO.

Reaction	Description	Reference
$H^+ + e^- + \text{grain} \rightarrow H + \text{grain}$	Grain surface recombination	Weingartner & Draine (2001)
$H^+ + e^- \rightarrow H + \gamma$	Gas-phase recombination	Ferland et al. (1992)
$H + e^- \rightarrow H^+ + 2e^-$	e^- collisional ionisation	Abel et al. (1997)
$H + \text{c.r.} \rightarrow H^+ + e^-$	Cosmic ray ionisation	Glover & Mac Low (2007)
$H + H + \text{grain} \rightarrow H_2 + \text{grain}$	Grain surface formation	Bergin et al. (2004)
$H_2 + \gamma \rightarrow 2H$	UV photodissociation	Draine & Bertoldi (1996)
$H_2 + \text{c.r.} \rightarrow H_2^+ + e^-$	Cosmic ray ionisation*	Bergin et al. (2004)
$C^+ + H_2 \rightarrow CH_2^+ + \gamma$	Radiative association	Nelson & Langer (1997)
$CH_2^+ + \text{various} \rightarrow CH_X + \text{various}$	Rapid neutralisation†	–
$CH_Z + O \rightarrow CO + H$	Gas phase formation	–
$CH_Z + \gamma \rightarrow C + H$	UV photodissociation	–
$CO + \gamma \rightarrow C^+ + O + e^-$	UV photodissociation††	–

* H_2^+ ions produced by cosmic ray ionisation of H_2 are assumed to dissociatively recombine to H + H, so that the effective reaction in the code is actually $H_2 + \text{c.r.} \rightarrow H + H$.
† Process is intermediate and is assumed rather than fully represented.
†† C is not present in our chemistry, but is assumed to rapidly photoionise to C^+.

cooling to completely decouple from the other equations in the code (Suttner et al. 1997), and to avoid cooling instabilities that may be generated by numerical errors (Ziegler, Yorke, & Kaisig 1996).

Cooling is currently implemented only with explicit timestepping of (287), where u is evolved alongside velocity in our leapfrog timestepping scheme. However, the substepping scheme described below for the chemical network (Section 2.14.4) is also used to update the cooling on the chemical timestep, meaning that the cooling can evolve on a much shorter timestep than the hydrodynamics when it is used in combination with chemistry, which it is by default. Implementation of an implicit cooling method, namely the one proposed by Townsend (2009), is under development.

2.14.3. Chemical network

A basic chemical network is included for ISM gas that evolves the abundances of H, H^+, e^-, and the molecules H_2 and CO. The number density of each species, n_X, is evolved using simple rate equations of the form

$$\frac{dn_X}{dt} = C_X - D_X n_X, \tag{289}$$

where C_X and D_X are creation and destruction coefficients for each species. In general, C_X and D_X are functions of density, temperature, and abundances of other species. The number density of each species, X, is time integrated according to

$$n_X(t + \Delta t) = n_X(t) + \frac{dn_X}{dt}\Delta t. \tag{290}$$

There are in effect only three species to evolve (H, H_2, and CO), as the H^+ and e^- abundances are directly related to the H abundance.

The chemistry of atomic hydrogen is effectively the same as in Glover & Mac Low (2007). H is created by recombination in the gas phase and on grain surfaces, and destroyed by cosmic ray ionisation and free electron collisional ionisation. H_2 chemistry is based on the model of Bergin et al. (2004), with H_2 created on grain surfaces and destroyed by photo-dissociation and cosmic rays (see Dobbs et al. 2008 for computational details).

The underlying processes behind CO chemistry are more complicated, and involve many intermediate species in creating CO from C and O by interactions with H species. Instead of following every intermediate step, we use the model of Nelson & Langer (1997) (see Pettitt et al. 2014 for computational details). CO is formed by a gas phase reaction from an intermediate CH_Z step after an initial reaction of C^+ and H_2 (where Z encompasses many similar type species). CO and CH_Z are subject to independent photo-destruction, which far outweighs the destruction by cosmic rays. Abundances of C^+ and O are used in the CO chemistry, and their abundance is simply the initial value excluding what has been used up in CO formation. Glover & Clark (2012) test this and a range of simpler and more complicated models, and show that the model adopted here is sufficient for large-scale simulations of CO formation, although it tends to over-produce CO compared to more sophisticated models.

The details for each reaction in the H, H_2, and CO chemistry are given in Table 4, with relevant references for each process.

2.14.4. Timestep constraint from chemistry

H chemistry is evolved on the cooling timestep, since the timescale on which the H^+ abundance changes significantly is generally comparable to or longer than the cooling time. This is not true for H_2 and CO. Instead, these species are evolved using a chemical timestepping criterion, where (290) is subcycled during the main time step at the interval $\Delta t_{\rm chem}$. If the abundance is decreasing, then the chemical timestep is

$$\Delta t_{\rm chem} = -\frac{1}{10}\frac{n_X}{(C_X - D_X n_X)}, \tag{291}$$

i.e., 10% of the time needed to completely deplete the species. If the abundance is increasing,

$$\Delta t_{\rm chem} = \frac{\Delta t_{\rm hydro}}{200}, \tag{292}$$

where $\Delta t_{\rm hydro}$ is the timestep size for the hydrodynamics, and was found to be an appropriate value by test simulations. These updated abundances feed directly into the relevant cooling functions. Although the cooling function includes Si I and C I, the abundances of these elements are set to zero in the current chemical model.

2.15. Particle injection

We implement several algorithms for modelling inflow boundary conditions (see Toupin et al. 2015a, 2015b for recent applications) This includes injecting SPH particles in spherical winds from sink particles (both steady and time dependent), in a steady Cartesian flow and for injection at the L_1 point between a pair of sink particles to simulate the formation of accretion discs by Roche Lobe overflow in binary systems.

3 INITIAL CONDITIONS

3.1. Uniform distributions

The simplest method for initialising the particles is to set them on a uniform Cartesian distribution. The lattice arrangement can be cubic (equal particle spacing in each direction, $\Delta x = \Delta y = \Delta z$), close-packed ($\Delta y = \sqrt{3/4}\Delta x$, $\Delta z = \sqrt{6}/3\Delta x$, repeated every three layers in z), hexagonal close-packed (as for close-packed but repeated every two layers in z), or uniform random. The close-packed arrangements are the closest to a 'relaxed' particle distribution, but require care with periodic boundary conditions due to the aspect ratio of the lattice. The cubic lattice is not a relaxed arrangement for the particles, but is convenient and sufficient for problems where the initial conditions are quickly erased (e.g. driven turbulence). For problems where initial conditions matter, it is usually best to relax the particle distribution by evolving the simulation for a period of time with a damping term (Section 3.6). This is the approach used, for example, in setting up stars in hydrostatic equilibrium (Section 3.4).

3.2. Stretch mapping

General non-uniform density profiles may be set up using 'stretch mapping' (Herant 1994). The procedure for spherical distributions is the most commonly used (e.g. Fryer, Hungerford, & Rockefeller 2007; Rosswog & Price 2007; Rosswog, Ramirez-Ruiz, & Hix 2009), but we have generalised the method for any density profile that is along one coordinate direction (e.g. Price & Monaghan 2004b). Starting with particles placed in a uniform distribution, the key is that a particle should keep the same relative position in the mass distribution. For each particle with initial coordinate x_0 in the relevant coordinate system, we solve the equation

$$f(x) = \frac{M(x)}{M(x_{\rm max})} - \frac{x_0 - x_{\rm min}}{(x_{\rm max} - x_{\rm min})} = 0, \tag{293}$$

where $M(x)$ is the desired density profile integrated along the relevant coordinate direction, i.e.

$$M(x) \equiv \int_{x_{\rm min}}^{x} \rho(x')\mathrm{d}S(x')\mathrm{d}x', \tag{294}$$

where the area element $\mathrm{d}S(x')$ depends on the geometry and the chosen direction, given by

$$dS(x) = \begin{cases} 1 & \text{Cartesian or cyl./sph. along } \phi, \theta \text{ or } z, \\ 2\pi x & \text{cylindrical along r}, \\ 4\pi x^2 & \text{spherical along r}. \end{cases} \tag{295}$$

We solve (293) for each particle using Newton–Raphson iterations

$$x = x - \frac{f(x)}{f'(x)}, \tag{296}$$

where

$$f'(x) = \frac{\rho(x)\mathrm{d}S(x)}{M(x_{\rm max})}, \tag{297}$$

iterating until $|f(x)| < 10^{-9}$. The Newton–Raphson iterations have second-order convergence, but may fail in extreme cases. Therefore, if the root finding has failed to converge after 30 iterations, we revert to a bisection method, which is only first-order but guaranteed to converge.

Stretch mapping is implemented in such a way that only the desired density function need be specified, either as through an analytic expression (implemented as a function call) or as a tabulated dataset. Since the mass integral in (294) may not be known analytically, we compute this numerically by integrating the density function using the trapezoidal rule.

The disadvantage of stretch mapping is that in spherical or cylindrical geometry it produces defects in the particle distribution arising from the initial Cartesian distribution of the particles. In this case, the particle distribution should be relaxed into a more isotropic equilibrium state before starting the simulation. For stars, for example, this may be done by simulating the star in isolation with artificial damping added (Section 3.6). Alternative approaches are to relax the simulation using an external potential chosen to produce the desired density profile in equilibrium (e.g. Zurek & Benz 1986; Nagasawa, Nakamura, & Miyama 1988) or to iteratively 'cool' the particle distribution to generate 'optimal' initial conditions (Diehl et al. 2015).

3.3. Accretion discs

3.3.1. Density field

The accretion disc setup module uses a Monte Carlo particle placement (details in Section A.7) in cylindrical geometry to

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

construct density profiles of the form

$$\rho(x, y, z) = \Sigma_0 f_s \left(\frac{R}{R_{\rm in}}\right)^{-p} \exp\left(\frac{-z^2}{2H^2}\right), \tag{298}$$

where Σ_0 is the surface density at $R = R_{\rm in}$ (if $f_s = 1$), $H \equiv c_s/\Omega$ is the scale height (with $\Omega \equiv \sqrt{GM/R^3}$), p is the power-law index (where $p = 3/2$ by default following Lodato & Pringle 2007), and $f_s \equiv (1 - \sqrt{R_{\rm in}/R})$ is an optional factor to smooth the surface density near the inner disc edge.

Several authors have argued that a more uniform particle placement is preferable for setting up discs in SPH (Cartwright, Stamatellos, & Whitworth 2009; Vanaverbeke et al. 2009). This may be important if one is interested in transient phenomena at the start of a simulation, but otherwise the particle distribution settles to a smooth density distribution within a few orbits (c.f. Lodato & Price 2010).

3.3.2. Velocity field

The orbital velocities of particles in the disc are set by solving the equation for centrifugal equilibrium, i.e.

$$v_\phi^2 = \frac{GM}{R} - f_p - 2v_\phi f_{\rm BH}, \tag{299}$$

where the correction from radial pressure gradients is given by

$$f_p = -c_s^2(R)\left(\frac{3}{2} + p + q + \frac{1}{2f_s}\right), \tag{300}$$

where q is the index of the sound speed profile such that $c_s(R) = c_{\rm s,\,in}(R/R_{\rm in})^{-q}$ and $f_{\rm BH}$ is a correction used for discs around a spinning black hole (Nealon et al. 2015)

$$f_{\rm BH} = -\frac{2a}{c^3}\left(\frac{GM}{R}\right)^2, \tag{301}$$

where a is the black hole spin parameter. The latter assumes Lense–Thirring precession is implemented as in Section 2.4.5. Where self-gravity is used, M is the enclosed mass at a given radius $M(< R)$, otherwise it is simply the mass of the central object. Using the enclosed mass for the self-gravitating case is an approximation since the disc is not spherically symmetric, but the difference is small and the disc relaxes quickly into the true equilibrium. Equation (299) is a quadratic for v_ϕ which we solve analytically.

3.3.3. Warps

Warps are applied to the disc (e.g. Lodato & Price 2010; Nealon et al. 2015) by rotating the particles about the y-axis by the inclination angle i [in general, a function of radius $i \equiv i(R)$], according to

$$x' = x\cos(i) + z\sin(i), \tag{302}$$

$$y' = y, \tag{303}$$

$$z' = -x\sin(i) + z\cos(i), \tag{304}$$

with the velocities similarly adjusted using

$$v_x' = v_x\cos(i) + v_z\sin(i), \tag{305}$$

$$v_y' = v_y, \tag{306}$$

$$v_z' = -v_x\sin(i) + v_z\cos(i). \tag{307}$$

3.3.4. Setting an α-disc viscosity

The simplest approach to mimicking an α-disc viscosity in SPH is to employ a modified shock viscosity term, setting the desired $\alpha_{\rm SS}$ according to (124) as described in more detail in Section 2.6.1. Since the factor $\langle h\rangle/H$ is dependent both on resolution and temperature profile (i.e. the q-index), it is computed in the initial setup by taking the desired $\alpha_{\rm SS}$ as input in order to give the required $\alpha_{\rm AV}$. Although this does not guarantee that $\alpha_{\rm SS}$ is constant with radius and time (this is only true with specific choices of p and q and if the disc is approximately steady), it provides a simple way to prescribe the disc viscosity.

3.4. Stars and binary stars

We implement a general method for setting up 'realistic' stellar density profiles, based on either analytic functions (e.g. polytropes) or tabulated data files output from stellar evolution codes [see Iaconi et al. (2017) for a recent application of this to common envelope evolution].

The basic idea is to set up a uniform density sphere of particles and set the density profile by stretch mapping (see above). The thermal energy of the particles is set so that the pressure gradient is in hydrostatic equilibrium with the self-gravity of the star for the chosen EOS. We then relax the star into equilibrium for several dynamical times using a damping parameter (Section 3.6), before re-launching the simulation with the star on an appropriate orbit.

For simulating red giants, it is preferable to replace the core of the star by a sink particle (see Passy et al. 2012; Iaconi et al. 2017). When doing so, one should set the accretion radius of the sink to zero and set a softening length for the sink particle consistent with the original core radius (see Section 2.8.1).

3.5. Galactic initial conditions

In addition to simulating ISM gas in galactic discs with analytic stellar potentials, one may represent bulge-halo-disc components by collisionless N-body particles (see Section 2.12.3). To replace a potential with a resolved system requires care with the initial conditions (i.e. position, velocity, and mass). If setup incorrectly the system will experience radial oscillations and undulations in the rotation curve, which will have strong adverse effects on the gas embedded within. We include algorithms for initialising the static-halo models of Pettitt et al. (2015) (which used the SPHNG SPH code). These initial conditions require the NFW profile to be active and care must be taken to ensure the mass and scale lengths correspond to the rotation curve

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

used to generate the initial conditions. Other codes may alternatively be used to seed multi-component N-body disc galaxies (e.g. Kuijken & Dubinski 1995a; Boily, Kroupa, & Peñarrubia-Garrido 2001; McMillan & Dehnen 2007; Yurin & Springel 2014), including MAGALIE (Boily et al. 2001) and GALIC (Yurin & Springel 2014) for which we have implemented format readers.

The gas in galactic scale simulations can be setup either in a uniform surface density disc, or according to the Milky Way's specific surface density. The latter is based on the radial functions given in Wolfire et al. (1995). As of yet, we have not implemented a routine for enforcing hydrostatic equilibrium (Springel, Di Matteo, & Hernquist 2005; Wang et al. 2010), this may be included in a future update.

3.6. Damping

To relax a particle distribution into equilibrium, we adopt the standard approach (e.g. Gingold & Monaghan 1977) of adding an external acceleration in the form

$$\boldsymbol{a}^{a}_{\rm ext,damp} = -f_{\rm d}\boldsymbol{v}, \tag{308}$$

such that a percentage of the kinetic energy is removed each timestep. The damping parameter, $f_{\rm d}$, is specified by the user. A sensible value for $f_{\rm d}$ is of order a few percent (e.g. $f_{\rm d} = 0.03$) such that a small fraction of the kinetic energy is removed over a Courant timescale.

4 SOFTWARE ENGINEERING

No code is completely bug free (experience is the name everyone gives to their mistakes; Wilde 1892). However, we have endeavoured to apply the principles of good software engineering to PHANTOM. These include

1. a modular structure,
2. unit tests of important modules,
3. nightly builds,
4. automated nightly tests,
5. automated code maintenance scripts,
6. version control with GIT,
7. wiki documentation, and
8. a bug database and issue tracker.

Together these simplify the maintenance, stability, and usability of the code, meaning that PHANTOM can be used direct from the development repository without fear of regression, build failures, or major bugs.

Specific details of how the algorithms described in Section 2 are implemented are given in the Appendix. Details of the test suite are given in Appendix A.9.

5 NUMERICAL TESTS

Unless otherwise stated, we use the M_6 quintic spline kernel with $h_{\rm fac} = 1.0$ by default, giving a mean neighbour number of 113 in 3D. Almost all of the test results are similar when adopting the cubic spline kernel with $h_{\rm fac} = 1.2$ (requiring ≈58 neighbours), apart from the tests with the one-fluid dust method where the quintic is required. Since most of the algorithms used in PHANTOM have been extensively tested elsewhere, our aim is merely to demonstrate that the implementation in the code is correct, and to illustrate the typical results that should be achieved on these tests when run by the user. The input files used to run the entire test suite shown in the paper are available on the website, so it should be straightforward for even a novice user to reproduce our results.

Unless otherwise indicated, we refer to dimensionless $\mathcal{L}_1$ and $\mathcal{L}_2$ norms when referencing errors, computed according to

$$\mathcal{L}_1 \equiv \frac{1}{NC_0}\sum_{i=1}^{N}|y_i - y_{\rm exact}|, \tag{309}$$

$$\mathcal{L}_2 \equiv \sqrt{\frac{1}{NC_0}\sum_{i=1}^{N}|y_i - y_{\rm exact}|^2}, \tag{310}$$

where $y_{\rm exact}$ is the exact or comparison solution interpolated or computed at the location of each particle i and N is the number of points. The norms are the standard error norms divided by a constant, which we set to the maximum value of the exact solution within the domain, $C_0 = \max(y_{\rm exact})$, in order to give a dimensionless quantity. Otherwise quoting the absolute value of $\mathcal{L}_1$ or $\mathcal{L}_2$ is meaningless. Dividing by a constant has no effect when considering convergence properties. These are the default error norms computed by SPLASH (Price 2007).

Achieving formal convergence in SPH is more complicated than in mesh-based codes where linear consistency is guaranteed (see Price 2012a). The best that can be achieved with regular (positive, symmetric) kernels is second-order accuracy away from shocks provided the particles remain well ordered (Monaghan 1992). The degree to which this remains true depends on the smoothing kernel and the number of neighbours. Our tests demonstrate that formal second-order convergence can be achieved with PHANTOM on certain problems (e.g. Section 5.6.1). More typically, one obtains something between first- and second-order convergence in smooth flow depending on the degree of spatial disordering of the particles. The other important difference compared to mesh-based codes is that there is no intrinsic numerical dissipation in SPH due to its Hamiltonian nature—numerical dissipation terms must be explicitly added. We perform all tests with these terms included.

We use timestep factors of $C_{\rm Cour} = 0.3$ and $C_{\rm force} = 0.25$ by default for all tests (Section 2.3.2).

5.1. Hydrodynamics

5.1.1. Sod shock tube

Figure 8 shows the results of the standard Sod (1978) shock tube test, performed in 3D using $[\rho, P] = [1, 1]$ in the 'left state' ($x \leqslant 0$) and $[\rho, P] = [0.125, 0.1]$ for the 'right state'

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

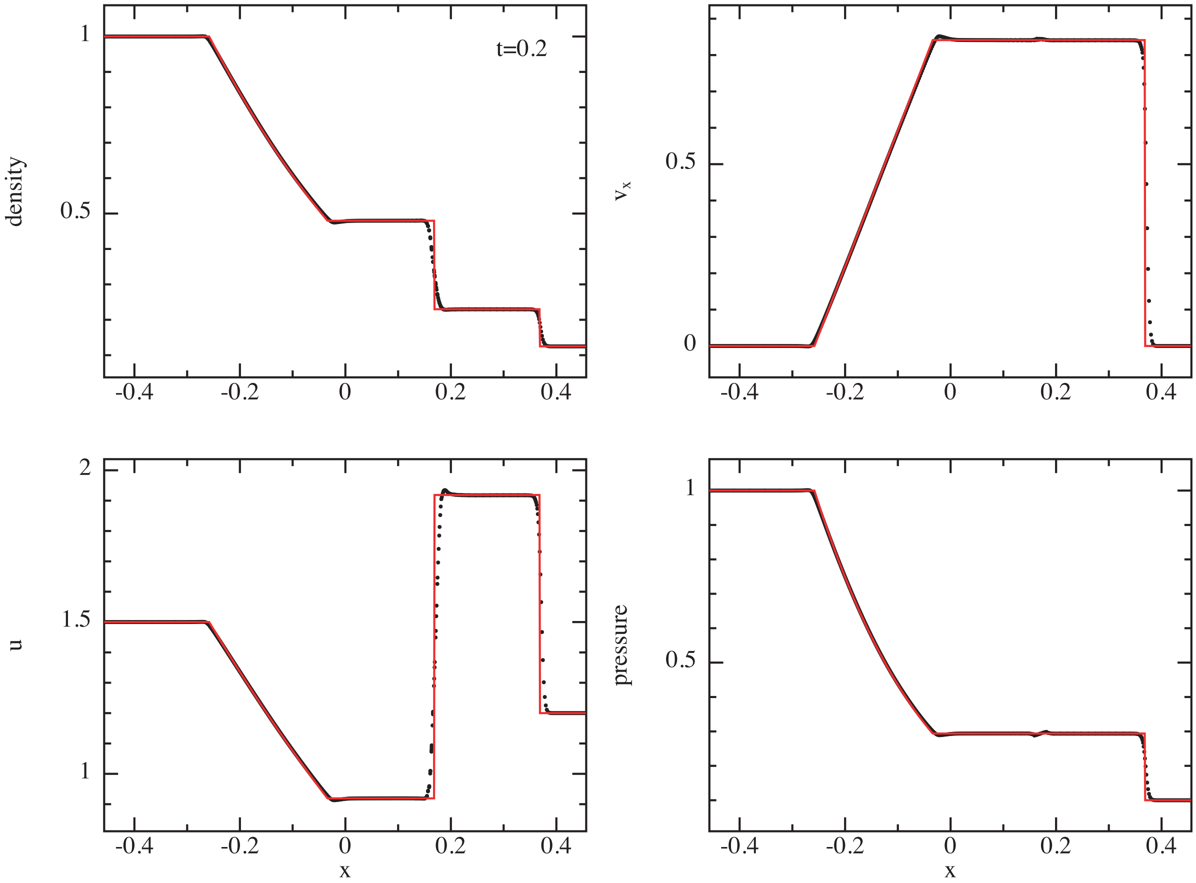

Figure 8. Results of the Sod shock tube test in 3D, showing projection of all particles (black dots) compared to the analytic solution (red line). The problem is set up with $[\rho, P] = [1, 1]$ for $x \leqslant 0$ and $[\rho, P] = [0.125, 0.1]$ for $x > 0$ with $\gamma = 5/3$, with zero velocities and no magnetic field. The density contrast is initialised using equal mass particles placed on a close packed lattice with 256 × 24 × 24 particles initially in $x \in [-0.5, 0]$ and 128 × 12 × 12 particles initially in $x \in [0, 0.5]$. The results are shown with constant $\alpha^{\rm AV} = 1$.

($x > 0$) with a discontinuity initially at $x = 0$ and zero initial velocity and magnetic field. We perform the test using an adiabatic EOS with $\gamma = 5/3$ and periodic boundaries in y and z. While many 1D solutions appear in the literature, only a handful of results on this test have been published for SPH in 3D (e.g. Dolag et al. 2005; Hubber et al. 2011; Beck et al. 2016; a 2D version is shown in Price 2012a). The tricky part in a 3D SPH code is how to set up the density contrast. Setting particles on a cubic lattice is a poor choice of initial condition since this is not a stable arrangement for the particles (Morris 1996b, 1996a; Lombardi et al. 1999; Børve et al. 2004). The approach taken in Springel (2005) (where only the density was shown, being the easiest to get right) was to relax the two halves of the box into a stable arrangement using the gravitational force run with a minus sign, but this is time consuming.

Here we take a simpler approach which is to set the particles initially on a close-packed lattice (Section 3.1), since this is close to the relaxed arrangement (e.g. Lombardi et al. 1999). To ensure continuity of the lattice across periodic boundaries, we fix the number of particles in the y (z) direction to the nearest multiple of 2 (3) and adjust the spacing in the x-direction accordingly to give the correct density in each subdomain. We implement the boundary condition in the x-direction by tagging the first and last few rows of particles in the x direction as boundary particles, meaning that their particle properties are held constant. The results shown in Figure 8 use 256 × 24 × 24 particles initially in $x \in [-0.5, 0]$ and 128 × 12 × 12 particles in $x \in [0, 0.5]$ with code defaults for the artificial conductivity [$\alpha_u = 1$ with $v^u_{\rm sig}$ given by equation (43)] and artificial viscosity ($\alpha^{\rm AV} = 1$, $\beta^{\rm AV} = 2$). The results are identical whether global or individual particle timesteps (Section 2.3.4) are used. Figure 9 shows the results when code defaults for viscosity are also employed, resulting in a time-dependent $\alpha^{\rm AV}$ (see lower right panel). There is more noise in the velocity field around the contact discontinuity in this case, but the results are otherwise identical.

The $\mathcal{L}_2$ errors for the solutions shown in Figure 8 are 0.0090, 0.0022, 0.0018, and 0.0045 for the density, velocity, thermal energy, and pressure, respectively. With dissipation switches turned on (Figure 9), the corresponding errors are 0.009, 0.0021, 0.0019, and 0.0044, respectively. That is, our solutions are within 1% of the analytic solution for all four quantities in both cases, with the density profile showing the strongest difference (mainly due to smoothing of the contact discontinuity).

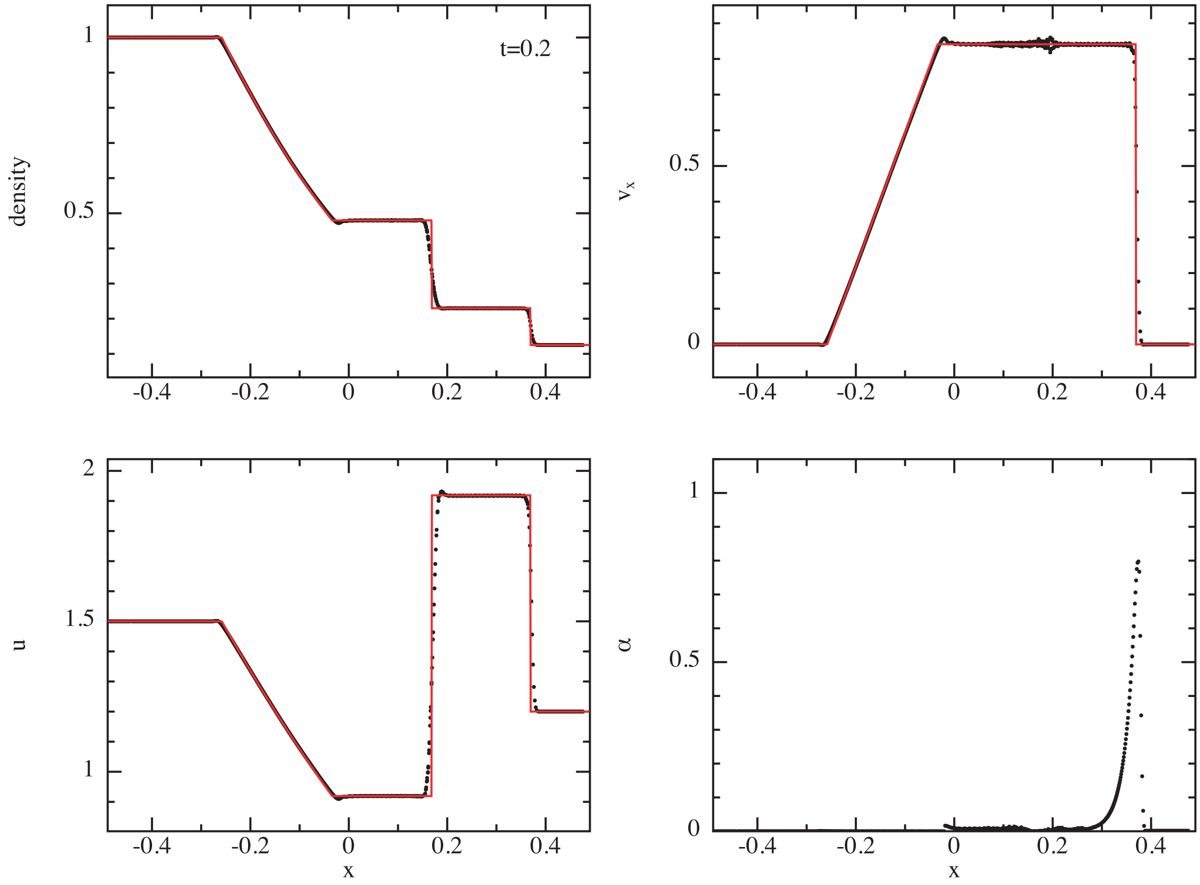

Figure 9. As in Figure 8, but with code defaults for all dissipation terms ($\alpha^{\rm AV} \in [0, 1]$; $\alpha_u = 1$). These leave more noise in the velocity field behind the shock but provide second-order convergence in smooth flow. The lower right panel in this case shows the resultant values for the viscosity parameter $\alpha^{\rm AV}$.

Puri & Ramachandran (2014) compared our shock capturing algorithm with other SPH variants, including Godunov SPH. Our scheme was found to be the most robust of those tested.

5.1.2. Blast wave

As a more extreme version of the shock test, Figures 10 and 11 show the results of the blast wave problem from Monaghan (1997), set up initially with $[\rho, P] = [1, 1000]$ for $x \leqslant 0$ and $[\rho, P] = [1.0, 0.1]$ for $x > 0$ and with $\gamma = 1.4$ (appropriate to air). As previously, we set the particles on a close-packed lattice with a discontinuous initial pressure profile. We employ 800 × 12 × 12 particles in the domain $x \in [-0.5, 0.5]$. Results are shown at $t = 0.01$. This is a more challenging problem than the Sod test due to the higher Mach number. As previously, we show results with both $\alpha^{\rm AV} = 1$ (Figure 10) and with the viscosity switch $\alpha \in [0, 1]$. Both calculations use $\alpha_u = 1$ with (43) for the signal speed in the artificial conductivity. For the solution shown in Figure 10, we find normalised $\mathcal{L}_2$ errors of 0.057, 0.063, 0.051, and 0.018 in the density, velocity, thermal energy, and pressure, respectively, compared to the analytic solution. Employing switches (Figure 11), we find corresponding $\mathcal{L}_2$ errors of 0.056, 0.059, 0.052, and 0.017. That is, our solutions are within 6% of the analytic solution at this resolution.

The main source of error is that the contact discontinuity is over-smoothed due to the artificial conductivity, while the velocity profile shows a spurious jump at the location of the contact discontinuity. This glitch is a startup error caused by our use of purely discontinuous initial conditions—it could be removed by adopting smoothed initial conditions but we prefer to perform the more difficult version of this test. There is also noise in the post-shock velocity profile because the viscosity switch does not apply enough dissipation here. As in the previous test, this noise can be reduced by increasing the numerical viscosity, e.g. by adopting a constant $\alpha^{\rm AV}$ (compare Figures 10 and 11).

Figure 12 quantifies the error in energy conservation, showing the error in the total energy as a function of time, i.e. $|E - E_0|/|E_0|$. Energy is conserved to a relative accuracy of better than 2×10^{-6}.

5.1.3. Sedov blast wave

The Sedov–Taylor blast wave (Taylor 1950a, 1950b; Sedov 1959) is a similar test to the previous but with a spherical geometry. This test is an excellent test for the individual

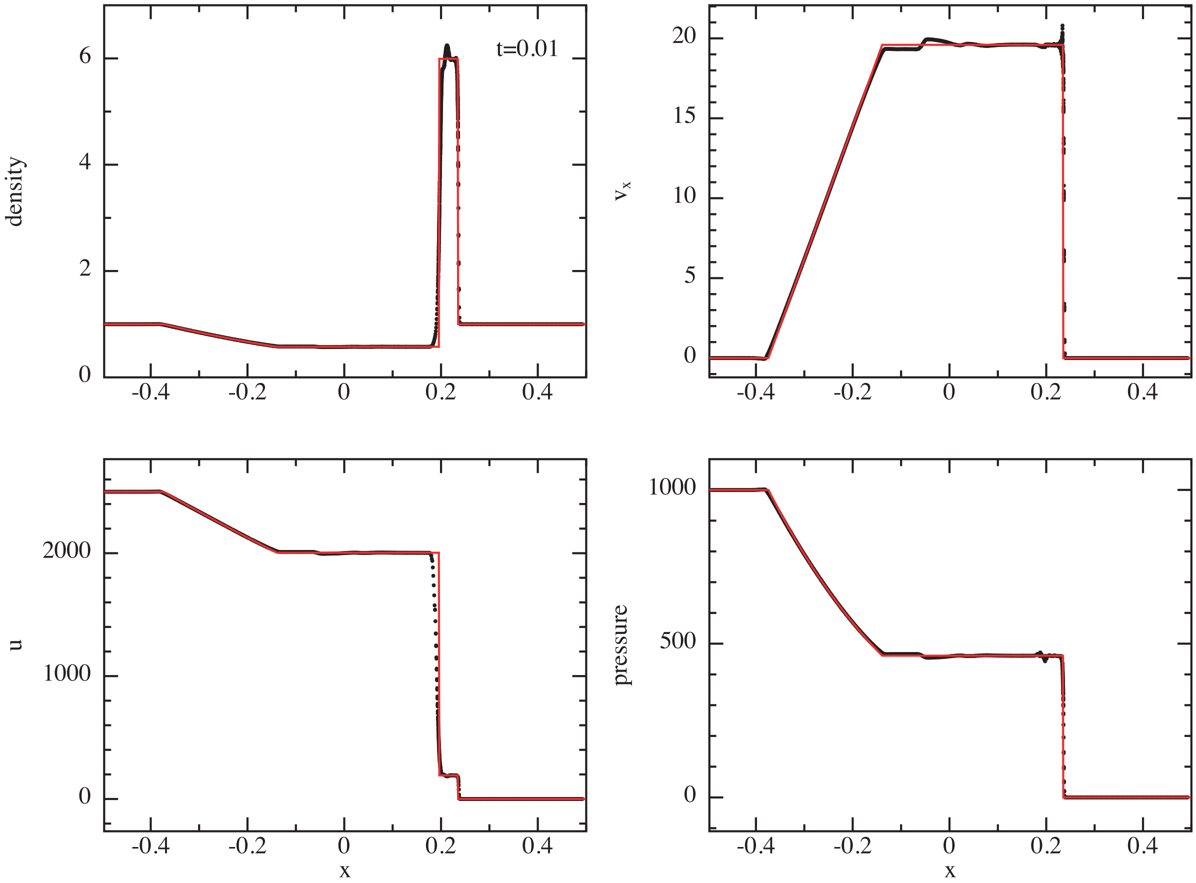

Figure 10. Results of the 3D blast wave test, showing projection of all particles (black dots) compared to the analytic solution (red line). The problem is set up with $[\rho, P] = [1, 1000]$ for $x \leqslant 0$ and $[\rho, P] = [1, 0.1]$ for $x > 0$ with $\gamma = 7/5$, with zero velocities and no magnetic field. We use equal mass particles placed on a close packed lattice with $800 \times 12 \times 12$ particles initially in $x \in [-0.5, 0.5]$. Results are shown with constant $\alpha^{\rm AV} = 1$.

timestepping algorithm, since it involves propagating a blast wave into an ambient medium of ‘inactive’ or ‘asleep’ particles, which can cause severe loss of energy conservation if they are not carefully awoken (Saitoh & Makino 2009). For previous uses of this test with SPH, see e.g. Springel & Hernquist (2002) and Rosswog & Price (2007), and for a comparison between SPH and mesh codes on this test, see Tasker et al. (2008).

We set up the problem in a uniform periodic box $x, y, z \in [-0.5, 0.5]$, setting the thermal energy on the particles to be non-zero in a barely resolved sphere around the origin. We assume an adiabatic EOS with $\gamma = 5/3$. The total energy is normalised such that the total thermal energy in the blast is $E_0 = \Sigma_a m_a u_a = 1$, distributed on the particles within $r < R_{\rm kern} h_0$ using the smoothing kernel, i.e.

$$u_a = \begin{cases} E_0 W(r, h_0), & r/h_0 \leq R_{\rm kern} \\ 0 & r/h_0 > R_{\rm kern}, \end{cases} \tag{311}$$

where $r = \sqrt{x^2 + y^2 + z^2}$ is the radius of the particle and we set h_0 to be twice the particle smoothing length.

We simulate the Sedov blast wave using both global and individual timesteps at several different resolutions. Figure 13 shows the evolution of the relative error in total energy for our suite, while Figure 14 shows the density at $t = 0.1$ compared to the analytical solution given by the solid line. Energy is conserved to better than 1% in all cases. Using higher spatial resolution results in a better match of the post-shock density with the analytic solution. The scatter at the leading edge of the shock is a result of the default artificial conductivity algorithm. Given the initial strong gradient in thermal energy, using more artificial conductivity than our switch permits would reduce the noise on this problem (Rosswog & Price 2007).

5.1.4. Kelvin–Helmholtz instability

Much has been written about Kelvin–Helmholtz instabilities with SPH (e.g. Agertz et al. 2007; Price 2008; Abel 2011; Valdarnini 2012; Read & Hayfield 2012; Hubber, Falle, & Goodwin 2013c). For the present purpose, it suffices to say that the test problems considered by Agertz et al. (2007) and Price (2008) are not well posed. That is, the number of small-scale secondary instabilities will always increase with numerical resolution because high wavenumber modes grow fastest in the absence of physical dissipation or other

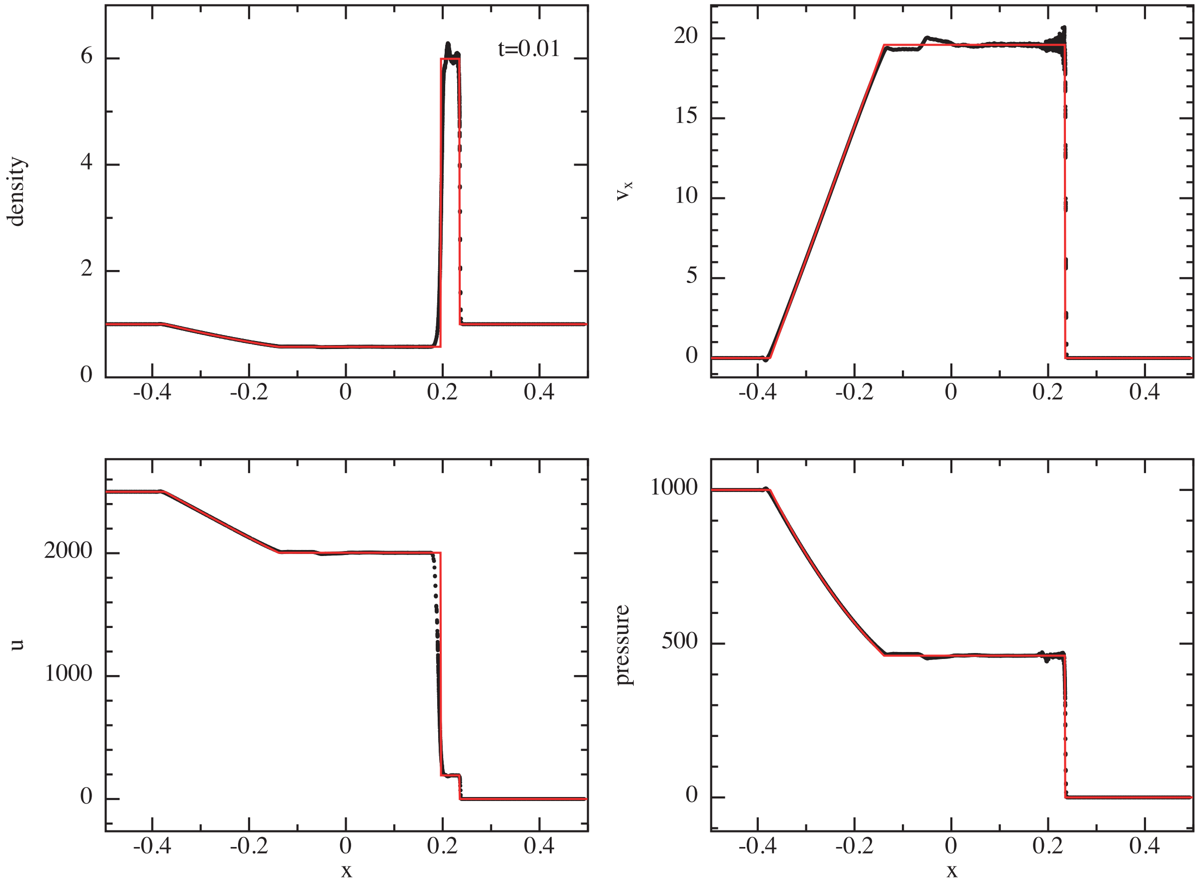

Figure 11. As in Figure 10, but with code defaults for dissipation switches ($\alpha^{\rm AV} \in [0, 1]$; $\alpha_u = 1$). As in Figure 9, the velocity field behind the shock is more noisy with switches applied, but the switches reduce the numerical dissipation away from shocks.

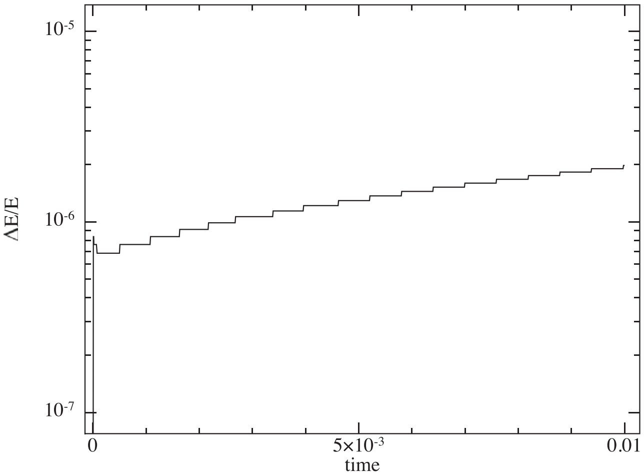

Figure 12. Relative error in energy conservation for the 3D blast wave test. Energy is conserved to an error of 10^{-6} in this test.

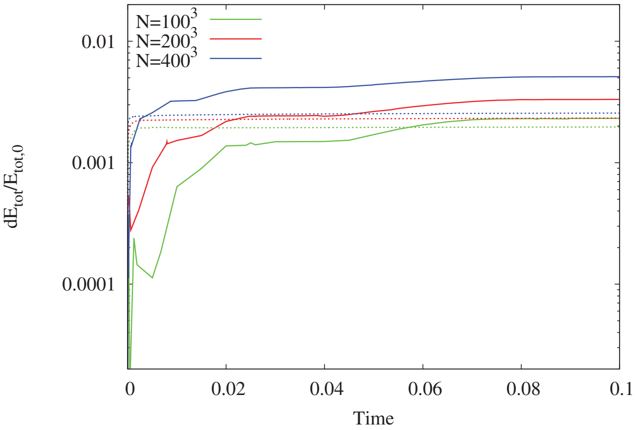

Figure 13. Evolution of the relative error in total energy for the Sedov blast wave problem. Resolutions are given in the legend; solid lines use individual timestepping while dotted lines show global timestepping.

regularising forces such as magnetic fields or surface tension. The ill-posed nature of the test problem has been pointed out by several authors (Robertson et al. 2010; McNally, Lyra, & Passy 2012; Lecoanet et al. 2016), who have each proposed well-posed alternatives.

We adopt the setup from Robertson et al. (2010), similar to the approach by McNally et al. (2012), where the initial density contrast is smoothed using a ramp function. This should suppress the formation of secondary instabilities long enough to allow a single large-scale mode to grow. The density and

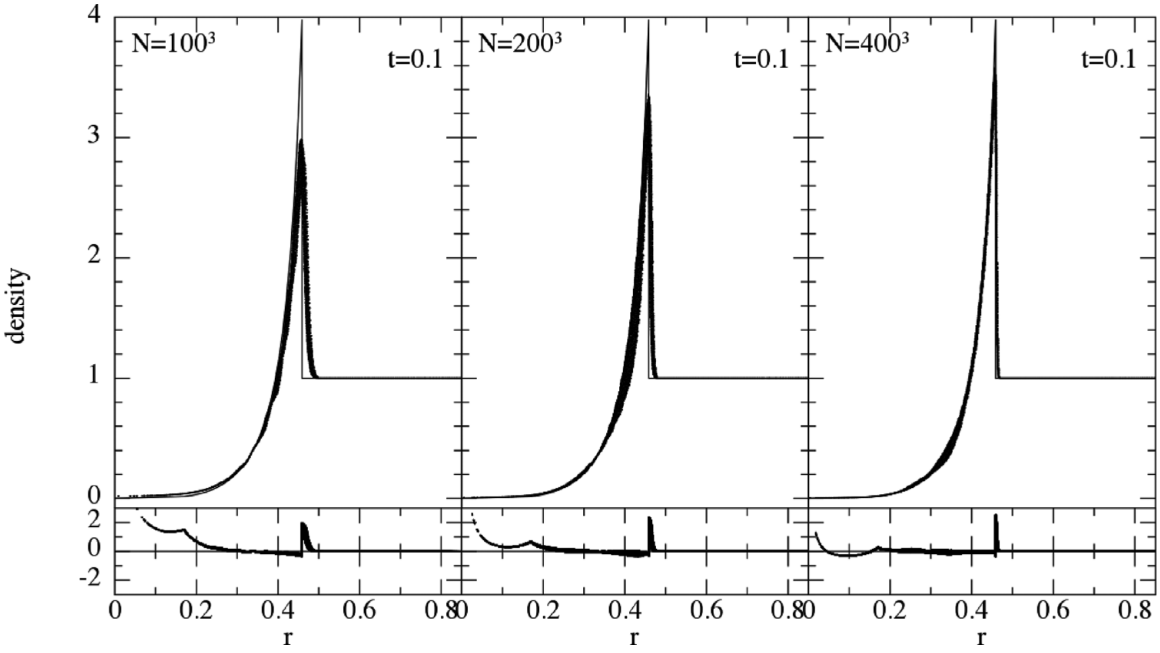

Figure 14. Density as a function of radius in the Sedov blast wave problem at three resolutions. All particles are placed initially on a closepacked lattice, are evolved using individual timesteps and we use the quintic kernel. The analytic solution is given by the solid line, and the bottom panels show the residuals compared to the analytical solution.

shear velocity in the y direction are given by

$$\rho(y) = \rho_1 + R(y)[\rho_2 - \rho_1], \tag{312}$$

and

$$v_x(y) = v_1 + R(y)[v_2 - v_1], \tag{313}$$

where $\rho_1 = 1$, $\rho_2 = 2$, $v_1 = -0.5$, and $v_2 = 0.5$ with constant pressure $P = 2.5$, $\gamma = 5/3$. The ramp function is given by

$$R(y) \equiv \left[1 - f(y)\right]\left[1 - g(y)\right], \tag{314}$$

where

$$f \equiv \frac{1}{1 + \exp\left[2(y - 0.25)/\Delta\right]},$$
$$g \equiv \frac{1}{1 + \exp\left[2(0.75 - y)/\Delta\right]}, \tag{315}$$

and we set $\Delta = 0.25$. Finally, a perturbation is added in the velocity in the y direction, given by

$$v_y = 0.1\sin(2\pi x). \tag{316}$$

The setup in Robertson et al. (2010) is 2D, but since Phantom is a 3D code, we instead set up the problem using a thin 3D box. We first set up a uniform close packed lattice in a periodic box with dimensions $1 \times 1 \times \sqrt{24}/n_x$, where n_x is the initial resolution in the x direction such that the box thickness is set to be exactly 12 particle spacings in the z direction independent of the resolution used in the x and y direction. The box is set between [0,1] in x and y, consistent with the ramp function. We then set up the density profile by stretch mapping in the y direction using (312) as the input function (c.f. Section 3.2).

Figure 15 shows the results of this test, showing a cross section of density at $z = 0$ for three different resolutions (top to bottom) and at the times corresponding to those shown in Robertson et al. (2010). We compute the quantitative difference between the calculations by taking the root mean square difference of the cross section slices shown above interpolated to a 1,024 × 1,024 pixel map. The error between the $n_x = 64$ calculation and the $n_x = 256$ calculation is 1.3×10^{-3}, while this reduces to 4.9×10^{-4} for the $n_x = 128$ calculation. Figure 16 shows the growth of the amplitude of the mode seeded in (316). We follow the procedure described in McNally et al. (2012) to calculate the mode amplitude. At $t = 2$, the amplitude between the $n_x = 128$ and $n_x = 256$ calculations is within 4%. The artificial viscosity and conductivity tend to slow convergence on this problem, so it is a good test of the dissipation switches (we use the default code viscosity switch as discussed in Section 2.2.9).

5.2. External forces

5.2.1. Lense–Thirring precession

We test the implementation of the Lense–Thirring precession by computing the precession induced on a pressure-less disc of particles, as outlined in Nealon et al. (2015). This disc is simulated for one orbit at the outer edge such that the inner part of the disc has precessed multiple times but the outer region has not yet completed a full precession. The precession timescale is estimated by measuring the twist as a function of time for each radial bin; in the inner region, this is the time

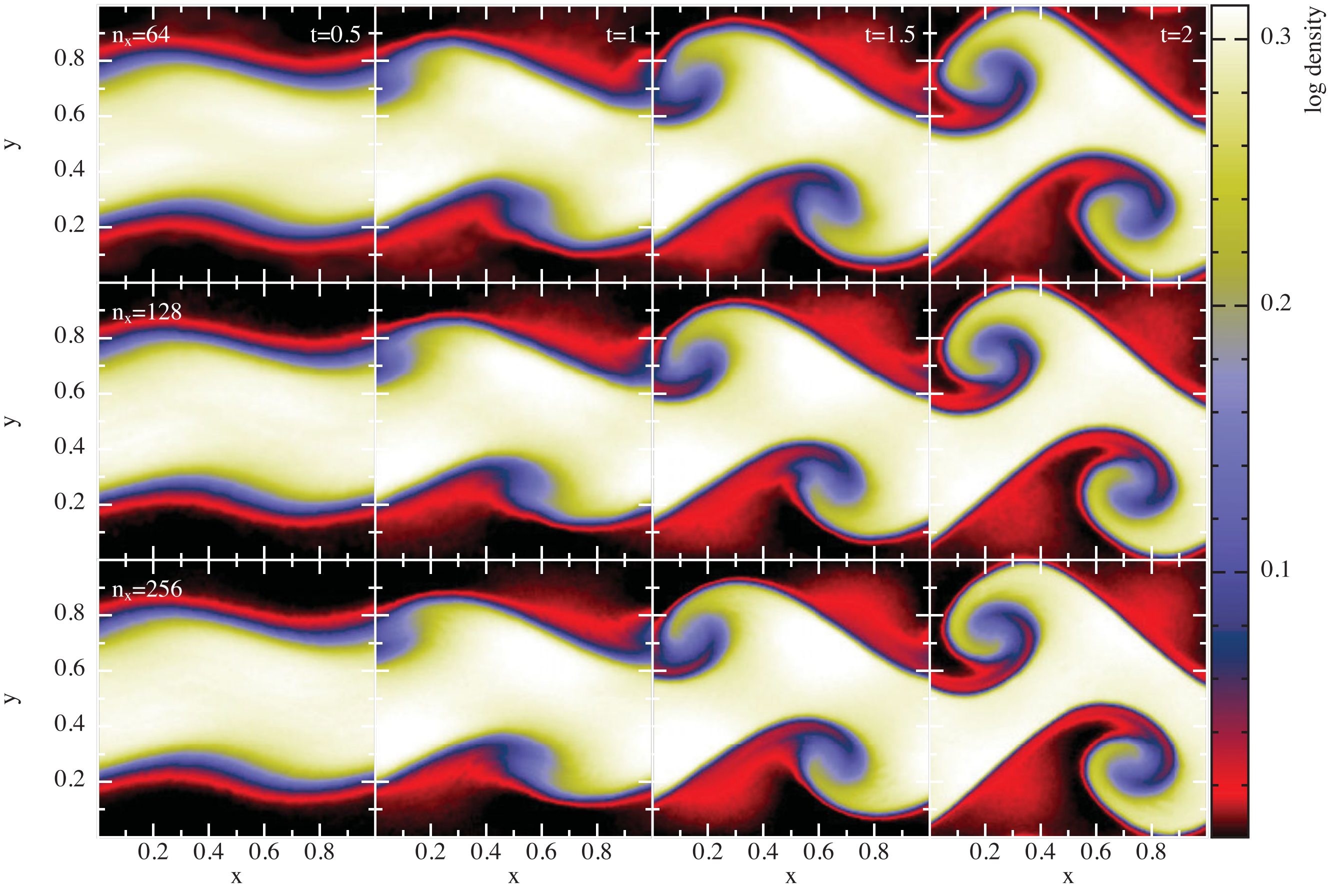

Figure 15. Results of the well-posed Kelvin–Helmholtz instability test from Robertson et al. (2010), shown at a resolution of (from top to bottom) 64 × 74 × 12, 128 × 148 × 12 and 256 × 296 × 12 equal mass SPH particles. We use stretch mapping (Section 3.2) to achieve the initial density profile, consisting of a 2:1 density jump with a smoothed transition.

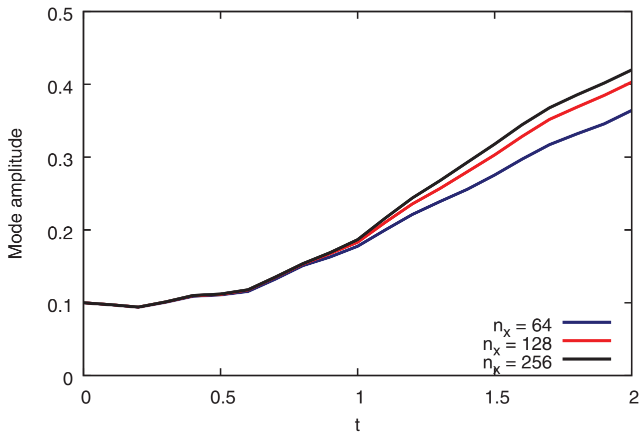

Figure 16. Growth of the amplitude of the seeded mode for the Kevin–Helmholtz instability test. The amplitude at $t = 2$ between the $n_x = 128$ and $n_x = 256$ calculations is within 4%.

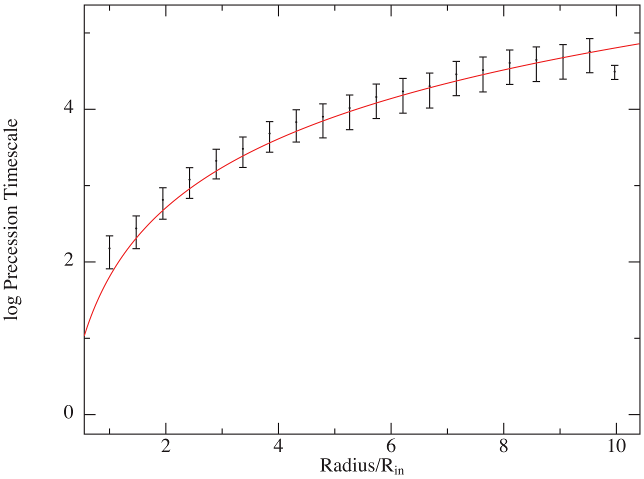

Figure 17. Lense–Thirring precession test from a disc inclined by 30°. Here, the precession timescale is measured from the cumulative twist in the disc and the exact solution, $t_p = R^3/2a$ is represented by the red line.

taken for the twist to go from a minimum (zero twist) to a maximum (fully twisted) and in the outer region the gradient of the twist against time is used to calculate the equivalent time. Figure 17 shows the precession timescale measured from the simulation as a function of the radius compared to the analytically derived precession timescale, with uncertainties derived from the calculation of the gradient.

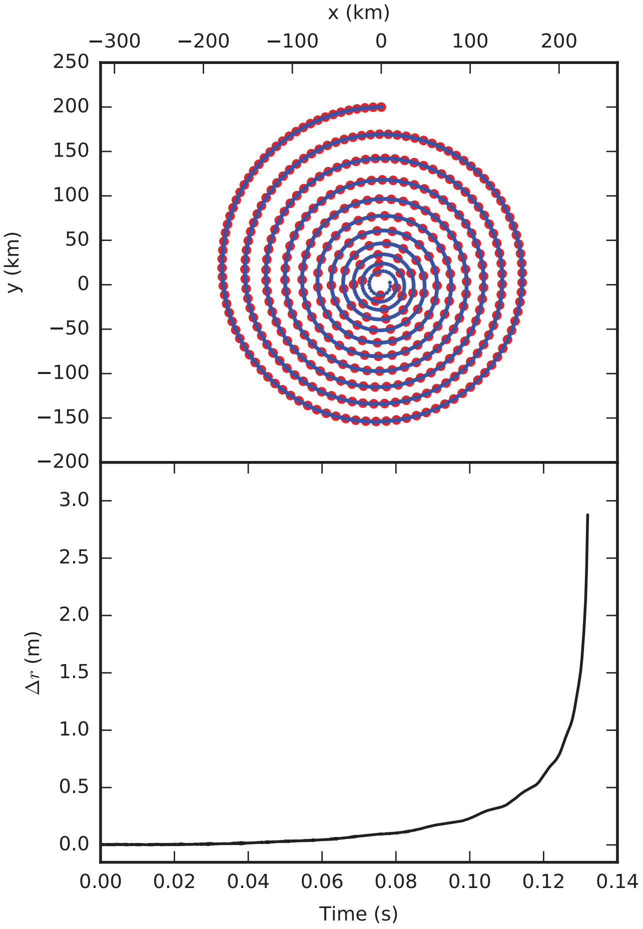

Figure 18. Inspiral of a test particle subjected to Poynting–Robertson drag using PHANTOM (blue curve) compared to the expected solution (red curve). For clarity, only some of the points are plotted. The results are indistinguishable on this scale. Bottom panel shows the distance (Δr) between the PHANTOM particle and the test code particle. This demonstrates that the implementation of Poynting–Robertson drag in PHANTOM is consistent with a fourth-order numerical solution to (106).

5.2.2. *Poynting–Robertson drag*

Figure 18 shows the trajectory of a spherical assembly of 89 pressureless SPH particles subject to Poynting–Robertson drag with a fixed value of $\beta_{\rm PR} = 0.1$, assuming a central neutron star of mass 1.4 $M_\odot$ and 10 km radius and a particle initially orbiting at $R = 200$ km with initial v_ϕ of 0.9 times the Keplerian orbital speed. We compare this to the trajectory of a test particle produced by direct numerical integration of the equations of motion, (106), with a fourth-order Runge Kutta scheme. As shown in Figure 18, there is no significant difference between the codes. We therefore expect that the behaviour of SPH gas or dust particles under the influence of any given β will be correct.

5.2.3. *Galactic potentials*

Figure 19 shows six calculations with gas embedded within different galactic potentials (Section 2.4.4). We set up an isothermal gas disc with $T = 10\,000$ K, with a total gas mass of 1×10^9 $M_\odot$ set up in a uniform surface density disc from 0–10 kpc in radius. A three-part potential model for the Milky Way provides the disc with an axisymmetric rotation curve [bulge plus disc plus halo, the same as Pettitt et al. (2014)]. The top row shows gas exposed to spiral potentials of Cox & Gómez (2002) with three different arm numbers (2, 3, and 4), while the bottom row shows simulations within the bar potential of Wada & Koda (2001) at three different pattern speeds (40, 60, and 80 km s^{-1} kpc^{-1}). All models are shown after approximately one full disc rotation (240 Myr). Gas can be seen to trace the different spiral arm features, with the two armed model in particular showing branches characteristic of such density wave potentials (e.g. Martos et al. 2004). The bars drive arm features in the gas, the radial extent of which is a function of the pattern speed. Also, note the inner elliptical orbits of the bar at the location of the Lindblad resonance which is an effect of the peaked inner rotation curve resulting from the central bulge.

5.3. Accretion discs

SPH has been widely used for studies of accretion discs, ever since the first studies by Artymowicz & Lubow (1994, 1996), Murray (1996), and Maddison, Murray, & Monaghan (1996) showed how to use the SPH artificial viscosity term to mimic a Shakura & Sunyaev (1973) disc viscosity.

5.3.1. *Measuring the disc viscosity*

The simplest test is to measure the disc viscosity from the diffusion rate of the disc surface density. Figure 20 shows the results of an extensive study of this with PHANTOM perfomed by Lodato & Price (2010). For this study, we set up a disc from $R_{\rm in} = 0.5$ to $R_{\rm out} = 10$ with surface density profile

$$\Sigma = \Sigma_0 R^{-p}\left(1 - \sqrt{\frac{R_{\rm in}}{R}}\right), \tag{317}$$

and a locally isothermal EOS $c_{\rm s} = c_{\rm s,0} R^{-q}$. We set $p = 3/2$ and $q = 3/4$ such that the disc is uniformly resolved, i.e. $h/H \sim$ constant (Lodato & Pringle 2007), giving a constant value of the Shakura & Sunyaev (1973) α parameter according to (124). We set $c_{\rm s,0}$ such that the aspect ratio is $H/R = 0.02$ at $R = 1$. We used 2 million particles by default, with several additional calculations performed using 20 million particles. The simulation is performed to $t = 1{,}000$ in code units.

The diffusion rate is measured by fitting the surface density evolution obtained from PHANTOM with the results of a 'ring code' solving the standard 1D diffusion equation for accretion discs (Lynden-Bell & Pringle 1974; Pringle 1981, 1992). Details of the fitting procedure are given in Lodato & Price (2010). In short, we use Newton–Raphson iterations to find the minimum error between the 1D code and the surface density profile from PHANTOM at the end of the simulation, which provides the best fit ($\alpha_{\rm fit}$) and error bars. Figure 20 shows that the measured diffusion rates agree with the expected values to within the error bars. The exception is for

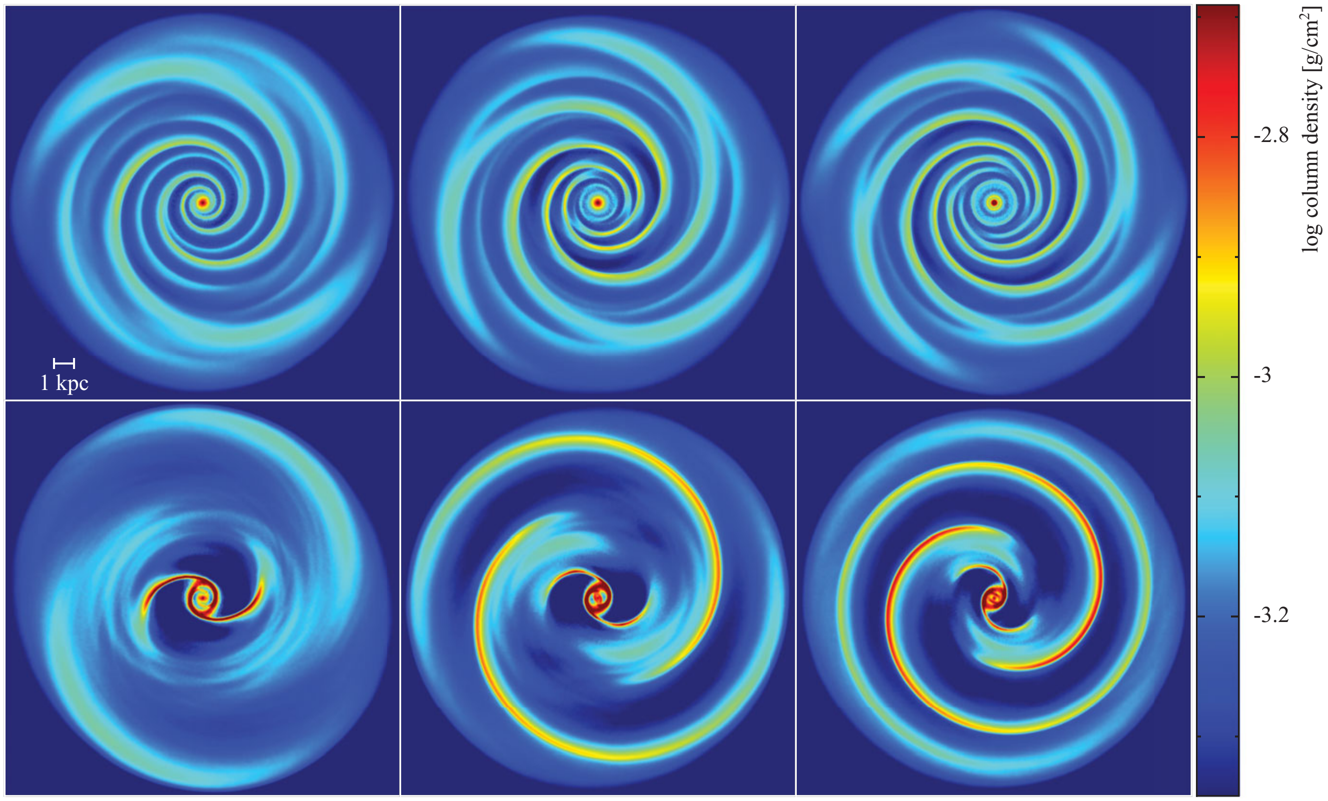

Figure 19. Gas in a galactic disc under the effect of different galactic potentials. Top row shows models with a 2, 3, and 4 armed spiral (left to right) with a pitch angle of 15° and pattern speed of 20 km s^{-1} kpc^{-1}. Bottom row shows a bar potential with pattern speeds of 40, 60, and 80 km s^{-1} kpc^{-1} (left to right).

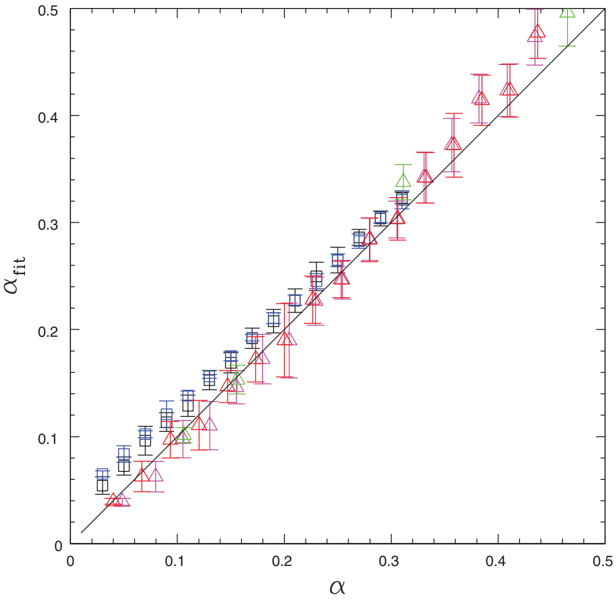

Figure 20. Calibration of the disc viscosity in PHANTOM, comparing the input value of the Shakura–Sunyaev α from (124) (x-axis) with the measured diffusion rate of the surface density by fitting to a 1D code (y-axis). Triangles indicate simulations with the disc viscosity computed using the artificial viscosity (Section 2.6.1), while squares represent simulations using physical viscosity (Section 2.7.1). All simulations use 2 million particles except for the green, cyan, and red triangles which use 20 million particles. Figure taken from Lodato & Price (2010).

low viscosity discs with physical viscosity, where contribution from artificial viscosity becomes significant. Triangles in the figure show the results with disc viscosity computed from the artificial viscosity (Section 2.6.1), while squares represent simulations with physical viscosity set according to (126).

This test demonstrates that the implementation of disc viscosity matches the analytic theory to within measurement errors. This also demonstrates that the translation of the artificial viscosity term according to (124) is correct.

5.3.2. *Warp diffusion*

A more demanding test of disc physics involves the dynamics of warped discs. Extensive analytic theory exists, starting with the linear theory of Papaloizou & Pringle (1983), subsequent work by Pringle (1992), and culminating in the work by Ogilvie (1999) which provides the analytic expressions for the diffusion rate of warps in discs for non-linear values of both disc viscosity and warp amplitude. Importantly, this theory applies in the 'diffusive' regime where the disc viscosity exceeds the aspect ratio, $\alpha > H/R$. For $\alpha \lesssim H/R$, the warp propagation is wave-like and no equivalent non-linear theory exists (see Lubow & Ogilvie 2000; Lubow, Ogilvie, & Pringle 2002).

PHANTOM was originally written to simulate warped discs—with our first study in Lodato & Price (2010) designed to test the Ogilvie (1999) theory in 3D simulations. Figure 21

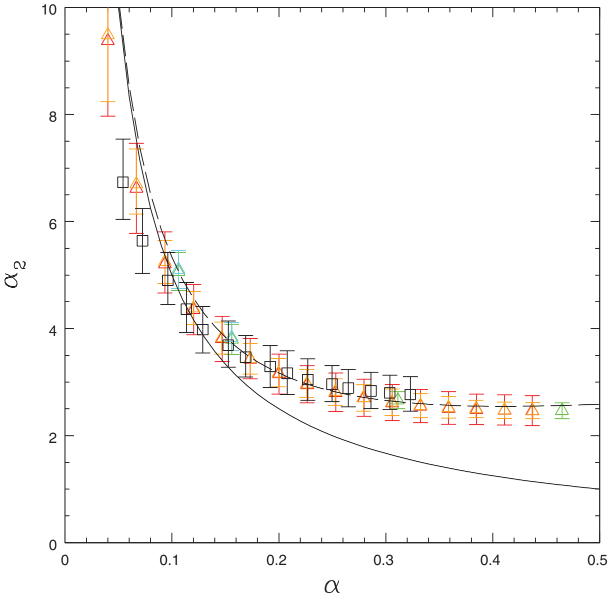

Figure 21. Warp diffusion rate as a function of disc viscosity, showing the PHANTOM results compared to the non-linear theory of Ogilvie (1999) (dashed line). The linear prediction, $\alpha_2 = 1/(2\alpha)$, is shown by the solid line, highlighting the agreement of PHANTOM with the non-linear theory. Colouring of points is as in Figure 20. Figure taken from Lodato & Price (2010).

shows the results of this study, showing the measured warp diffusion rate as a function of disc viscosity. The setup of the simulations is as in the previous test but with a small warp added to the disc, as outlined in Section 3.3 of Lodato & Price (2010). Details of the fitting procedure used to measure the warp diffusion rate are also given in Section 4.2 of that paper. The dashed line shows the non-linear prediction of Ogilvie (1999), namely

$$\alpha_2 = \frac{1}{2\alpha}\frac{4(1+7\alpha^2)}{4+\alpha^2}. \tag{318}$$

Significantly, the PHANTOM results show a measurable difference between the predictions of the non-linear theory and the prediction from linear theory (Papaloizou & Pringle 1983) of $\alpha_2 = 1/(2\alpha)$, shown by the solid black line.

In addition to the results shown in Figure 21, PHANTOM also showed a close match to both the predicted self-induced precession of the warp and to the evolution of non-linear warps (see Figures 13 and 14 in Lodato & Price 2010, respectively). From the success of this initial study, we have used PHANTOM to study many aspects of disc warping, either with isolated warps or breaks (Lodato & Price 2010; Nixon et al. 2012a), warps induced by spinning black holes (Nixon et al. 2012b; Nealon et al. 2015, 2016), and warps in circumbinary (Nixon et al. 2013; Facchini et al. 2013) or circumprimary (Doğan et al. 2015; Martin et al. 2014a, 2014b) discs. In particular, PHANTOM was used to discover the phenomenon of 'disc tearing' where sections of the disc are 'torn' from the disc plane and precess effectively independently (Nixon et al. 2012b, 2013; Nealon et al. 2015).

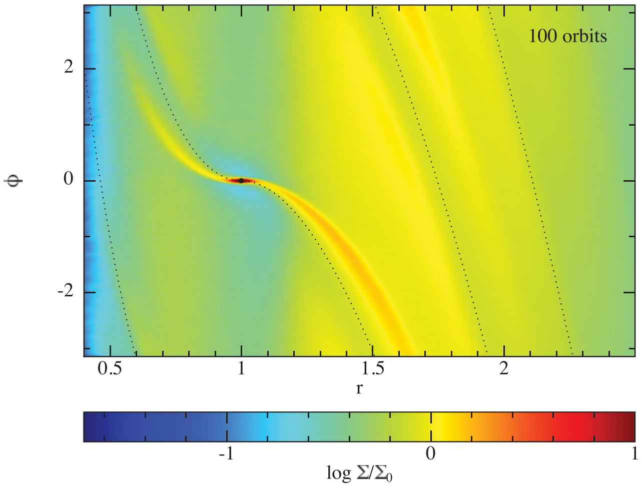

Figure 22. Planet–disc interaction in 3D, showing the 'viscous Jupiter' calculation comparable to the 2D results shown with various grid and SPH codes in Figure 10 of de Val-Borro et al. (2006). The dotted lines shows the estimated position of the planetary shocks from Ogilvie & Lubow (2002). The offset between this solution and the numerical shock position is due to the approximate nature of the analytic solution (see de Val-Borro et al. 2006).

5.3.3. Disc–planet interaction

Although there is no 'exact' solution for planet–disc interaction, an extensive code comparison was performed by de Val-Borro et al. (2006). Figure 22 shows the column density of a 3D PHANTOM calculation, plotted in r–ϕ with the density integrated through the z direction, comparable to the 'viscous Jupiter' setup in de Val-Borro et al. (2006).

Two caveats apply when comparing our results with those in Figure 10 of de Val-Borro et al. (2006). The first is that the original comparison project was performed in 2D and mainly with grid-based codes with specific 'wave damping' boundary conditions prescribed. We chose simply to ignore the prescribed boundary conditions and two dimensionality and instead modelled the disc in 3D with a central accretion boundary at $r = 0.25$ with a free outer boundary, with the initial disc set up from $r = 0.4$ to $r = 2.5$. We used 10^6 SPH particles. Second, the planetary orbit was prescribed on a fixed circular orbit with no accretion onto either the planet or the star. Although we usually use sink particles in PHANTOM to model planet–disc interaction (e.g. Dipierro et al. 2015), for this test we thus employed the fixed binary potential (Section 2.4.2) to enable a direct comparison. We thus used $M = 10^{-3}$ in the binary potential, corresponding to the 'Jupiter' simulation in de Val-Borro et al. (2006) with the planet on a fixed circular orbit at $r = 1$.

As per the original comparison project, we implemented Plummer softening of the gravitational force from the planet

$$\phi_{\rm planet} = -\frac{-m_{\rm planet}}{\sqrt{r^2+\epsilon^2}}, \tag{319}$$

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

where $\varepsilon = 0.6H$. We also implemented the prescribed increase of planet mass with time for the first five orbits according to

$$m_{\rm planet} = \sin^2\left(\frac{\pi t}{10P_{\rm orbit}}\right), \quad (320)$$

where $P_{\rm orbit}$ is the orbital period (2π in code units), though we found this made little or no difference to the results in practice. We assumed a locally isothermal EOS, $c_{\rm s} \propto R^{-0.5}$, such that $H/R \approx 0.05 \approx$ constant. We assumed an initially constant surface density, $\Sigma_0 = 0.002\, M_*/(\pi a^2)$. We also employed a Navier–Stokes viscosity with $\nu = 10^{-5}$ *on top of* the usual settings for shock dissipation with the viscosity switch, namely $\alpha_{\rm AV} \in [0, 1]$ and $\beta_{\rm AV} = 2$. We use the Navier–Stokes viscosity implementation described in Section 2.7.2, which is the code default. We also employed the cubic spline kernel (with $h_{\rm fac} = 1.2$) rather than the quintic to reduce the computational expense (using the quintic made no difference to the results).

Despite the different assumptions, the results in Figure 22 are strikingly similar to those obtained with most of the grid-based codes in de Val-Borro et al. (2006). The main difference is that our gap is shallower, which is not surprising since this is where resolution is lowest in SPH. There is also some difference in the evolution of the surface density, particularly at the inner boundary, due to the difference in assumed boundary conditions. However, the dense flow around the planet and in the shocks appear well resolved compared to the other codes. What is interesting is that the SPH codes used in the original comparison performed poorly on this test. This may be simply due to the low resolution employed, as the two SPH calculations used 250 000 and 300 000 particles, respectively, (though performed only in 2D rather than 3D), but given the extent of other differences between adopted setup and SPH algorithms, it is hard to draw firm conclusions.

As per the original comparison, Figure 22 shows the estimated position of the planetary shocks from Ogilvie & Lubow (2002) plotted as dotted lines, namely

$$\phi(r,t) = \begin{cases} t - \dfrac{2}{3\varepsilon}\left(r^{3/2} - \dfrac{3}{2}\ln r - 1\right); & r > r_{\rm planet}, \\ t + \dfrac{2}{3\varepsilon}\left(r^{3/2} - \dfrac{3}{2}\ln r - 1\right); & r < r_{\rm planet}, \end{cases} \quad (321)$$

where $\varepsilon = 0.05$ is the disc aspect ratio. The disagreement between the shock position and the Ogilvie & Lubow (2002) solution seen in Figure 22 was also found in every simulation shown in de Val-Borro et al. (2006), so more likely reflects the approximate nature of the analytic solution rather than numerical error.

We found this to be a particularly good test of the viscosity limiter, since there is both a shock and a shear flow present. Without the limiter in (46), we found the shock viscosity switch would simply trigger to $\alpha_{\rm AV} \approx 1$. The original Morris & Monaghan (1997) switch (Section 2.2.9) also performs well on this test, suggesting that the velocity divergence is better able to pick out shocks in differentially rotating discs compared to its time derivative.

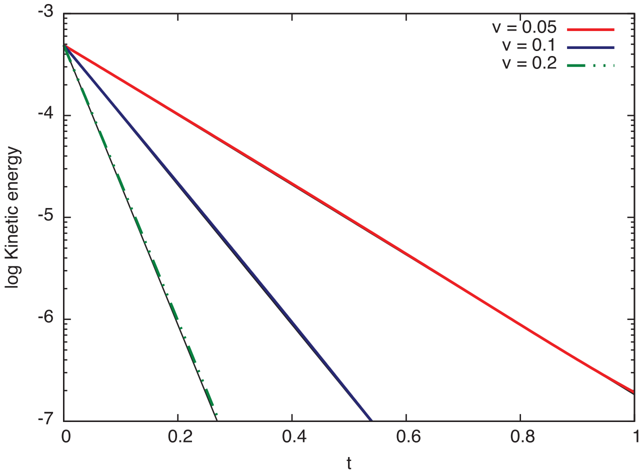

Figure 23. Test of physical Navier–Stokes viscosity in the Taylor–Green vortex using kinematic shear viscosity $\nu = 0.05$, 0.1, 0.2. The exponential decay rate of kinetic energy may be compared to the analytic solution in each case (solid black lines), demonstrating that the calibration of physical viscosity in PHANTOM is correct.

5.4. Physical viscosity

5.4.1. Taylor–Green vortex

The Taylor–Green vortex (Taylor & Green 1937) consists of a series of counter-rotating vortices. We perform this test using four vortices set in a thin 3D slab. The initial velocity fields are given by $v_x = v_0 \sin(2\pi x)\cos(2\pi y)$, $v_y = -v_0 \cos(2\pi x)\sin(2\pi y)$ with $v_0 = 0.1$. The initial density is uniform $\rho = 1$, and an isothermal EOS is used ($P = c_{\rm s}^2\rho$) with speed of sound $c_{\rm s} = 1$. Viscosity will cause each component of the velocity field to decay at a rate $\propto \exp(-16\pi^2\nu t)$, where ν is the kinematic shear viscosity.

Figure 23 shows the kinetic energy for a series of calculations using $\nu = 0.05$, 0.1, 0.2. In each case, the kinetic energy exponentially decays by several orders of magnitude. The corresponding analytic solutions are shown by the solid black lines for comparison, demonstrating that the implementation of physical viscosity in PHANTOM is correct.

5.5. Sink particles

5.5.1. Binary orbit

Figure 24 shows the error in total energy conservation $\Delta E/|E_0|$ for a set of simulations consisting of two sink particles set up in a binary orbit, a common test of N-body integrators (e.g. Hut, Makino, & McMillan 1995; Quinn et al. 1997; Farr & Bertschinger 2007; Dehnen & Read 2011). We fix the initial semi-major axis $a = 1$ with masses $m_1 = m_2 = 0.5$ and with the two sink particles initially at periastron, corresponding to $\boldsymbol{x}_1 = [-m_2/(m_1 + m_2)\Delta, 0, 0]$ and $\boldsymbol{x}_2 = [m_1/(m_1 + m_2)\Delta, 0, 0]$, where $\Delta = a(1 - e)$ is the initial separation. The corresponding initial velocities are $\boldsymbol{v}_1 = [0, -m_2/(m_1 + m_2)|v|, 0]$ and $\boldsymbol{v}_2 = [0, m_1/(m_1 + m_2)|v|, 0]$, where $|v| = \sqrt{a(1 - e^2)(m_1 + m_2)/\Delta}$. The

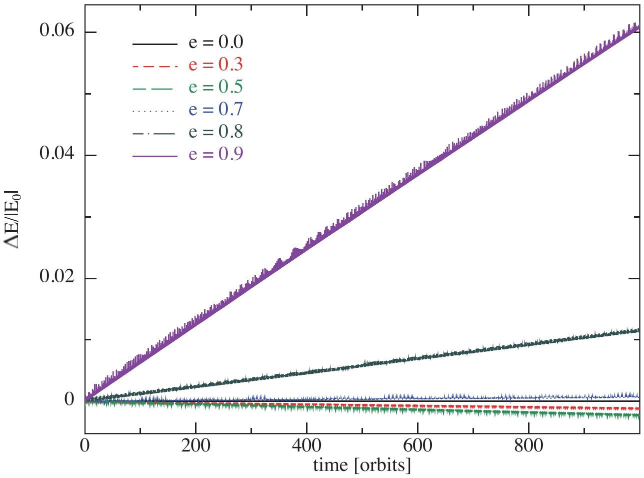

Figure 24. Errors in energy conservation in a sink particle binary integration with code default parameters for the timestep control, showing the energy drift caused by the adaptive timestepping. Angular momentum is conserved to machine precision.

orbital period is thus $P = \sqrt{4\pi^2 a^3/(G(m_1 + m_2))} = 2\pi$ in code units for our chosen parameters. Importantly, we use an adaptive timestep which is not time symmetric so there remains some drift in the energy error which is absent if the timestep is constant (see e.g. Hut et al. 1995; Quinn et al. 1997 and Dehnen & Read 2011 for discussion of this issue).

Figure 24 shows the error in total energy as a function of time for the first 1,000 orbits for calculations with initial eccentricities of $e = 0.0$ (a circular orbit), 0.3, 0.5, 0.7, and 0.9. Energy conservation is worse for more eccentric orbits, as expected, with $\Delta E/|E_0| \sim 6\%$ after 1,000 orbits for our most extreme case ($e = 0.9$). The energy error can be reduced arbitrarily by decreasing the timestep, so this is mainly a test of the default settings for the sink particle timestep control. For this problem, the timestep is controlled entirely by (76), where by default we use $\eta_\Phi = 0.05$, giving 474 steps per orbit for $e = 0.9$. For simulations with more eccentric orbits, we recommend decreasing $C_{\rm force}$ from the default setting of 0.25 to obtain more accurate orbital dynamics.

In addition to calibrating the timestep constraint, Figure 24 also validates the sink particle substepping via the RESPA algorithm (Section 2.3.3) since for this problem the 'gas' timestep is set only by the desired interval between output files (to ensure sufficient output for the figure we choose $\Delta t_{\rm max} = 1$, but we also confirmed that this choice is unimportant for the resultant energy conservation). This means that increasing the accuracy of sink particle interactions adds little or no cost to calculations involving gas particles.

The corresponding plot for angular momentum conservation (not shown) merely demonstrates that angular momentum is conserved to machine precision ($\Delta L/|L| \sim 10^{-15}$), as expected. Importantly, angular momentum remains conserved to machine precision even with adaptive timestepping.

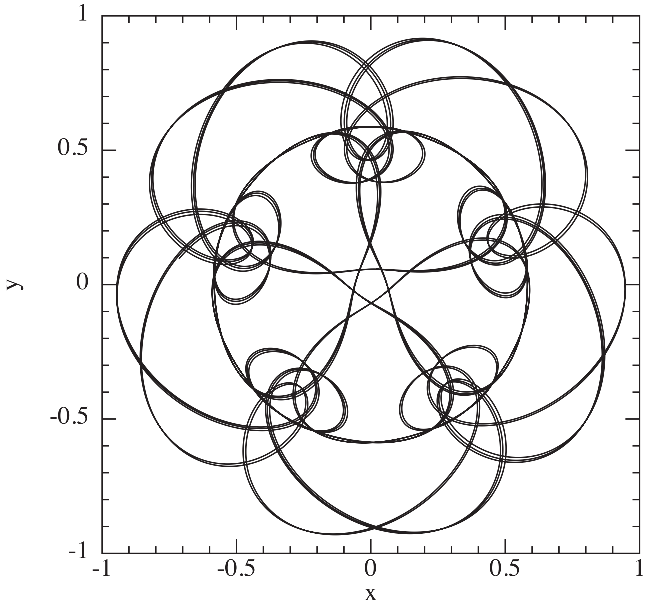

Figure 25. Orbit of a sink particle in the restricted three-body problem from Chin & Chen (2005) using default code parameters. This tests both the time integrator for sink particles and the interaction with a time-dependent binary potential (Section 2.4.2). The trajectory of the sink is plotted every timestep for three periods ($t = 27\pi$).

5.5.2. *Restricted three-body problem*

Chin & Chen (2005) proposed a more demanding test of *N*-body integrators, consisting of a test particle orbiting in the potential of a binary on a fixed circular orbit. We set up this problem with a single sink particle with $\boldsymbol{x} = [0, 0.0580752367, 0]$ and $\boldsymbol{v} = [0.489765446, 0, 0]$, using the time-dependent binary potential as described in Section 2.4.2 with $M = 0.5$. This is therefore a good test of the interaction between a sink particle and external potentials in the code, as well as the sink particle timestepping algorithm. For convenience, we set the sink mass $m = 1$ and accretion radius $r_{\rm acc} = 0.1$, although both are irrelevant to the problem.

Figure 25 shows the resulting orbit using the default code parameters, where we plot the trajectory of the sink particle up to $t = 27\pi$, as in Figures 1 and 2 of Chin & Chen (2005). Considering that we use only a second-order integrator, the orbital trajectory is remarkably accurate, showing no chaotic behaviour and only a slight precession consistent or better than the results with some of the fourth-order schemes shown in their paper (albeit computed with a larger timestep). We are thus satisfied that our time integration scheme and the associated timestep settings can capture complex orbital dynamics with sufficient accuracy.

5.6. Magnetohydrodynamics

5.6.1. *3D circularly polarised Alfvén wave*

Tóth (2000) introduced the circularly polarised Alfvén wave test, an exact non-linear solution to the MHD equations which can therefore be performed using a wave of arbitrarily large

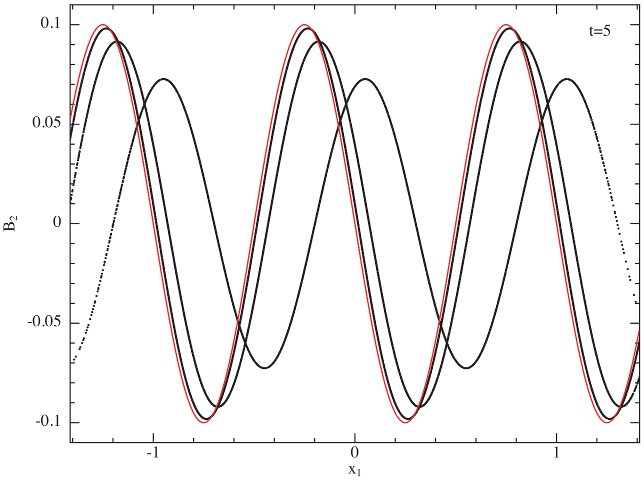

Figure 26. Results of the 3D circularly polarised Alfven wave test after five periods, showing perpendicular component of the magnetic field on all particles (black dots) as a function of distance along the axis parallel to the wave vector. Results are shown using 32 × 18 × 18, 64 × 36 × 39, and 128 × 74 × 78 particles (most to least damped, respectively), compared to the exact solution given by the solid red line. Convergence is shown in Figure 27.

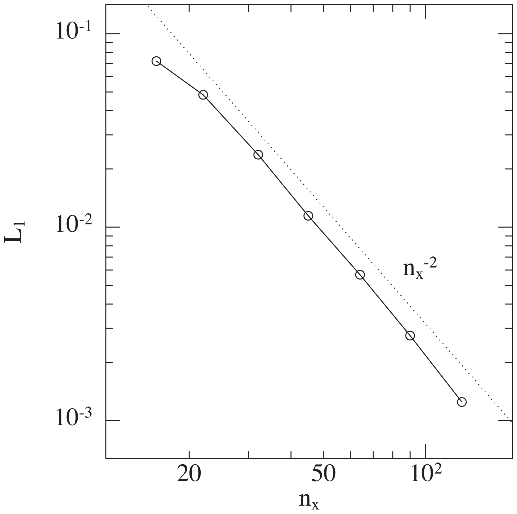

Figure 27. Convergence in the 3D circularly polarised Alfven wave test, showing the L_1 error as a function of the number of particles along the x-axis, n_x alongside the expected slope for second-order convergence (dotted line). Significantly, this demonstrates second-order convergence *with all dissipation switched on*.

amplitude. Most results of this test are shown in 2D (e.g. Price & Monaghan 2005; Rosswog & Price 2007; Price 2012a; Tricco & Price 2013). Here we follow the 3D setup outlined by Gardiner & Stone (2008). We use a periodic domain of size $L \times L/2 \times L/2$, where $L = 3$, with the wave propagation direction defined using angles a and b where $\sin a = 2/3$ and $\sin b = 2/\sqrt{5}$, with the unit vector along the direction of propagation given by $\boldsymbol{r} = [\cos a \cos b, \cos a \sin b, \sin a]$. We use an initial density $\rho = 1$, an adiabatic EOS with $\gamma = 5/3$ and $P = 0.1$. We perform the 'travelling wave test' from Gardiner & Stone (2008), where the wavelength $\lambda = 1$ and the vectors $[v_1, v_2, v_3] = [0, 0.1\sin(2\pi x_1/\lambda), 0.1\cos(2\pi x_1/\lambda)]$ and $[B_1, B_2, B_3] = [1, 0.1\sin(2\pi x_1/\lambda), 0.1\cos(2\pi x_1/\lambda)]$ are projected back into the x, y, and z components using the transformations given by (Gardiner & Stone 2008)

$$
\begin{aligned}
x &= x_1 \cos a \cos b - x_2 \sin b - x_3 \sin a \cos b,\\
y &= x_1 \cos a \sin b + x_2 \cos b - x_3 \sin a \sin b,\\
z &= x_1 \sin a + x_3 \cos a. \qquad (322)
\end{aligned}
$$

Figure 26 shows the results of this test using 32 × 18 × 18, 64 × 36 × 39 and 128 × 74 × 78 particles initially set on a close packed lattice, compared to the exact solution given by the red line (the same as the initial conditions for the wave). We plot the transverse component of the magnetic field B_2 as a function of x_1, where $B_2 \equiv (B_y - 2B_x)/\sqrt{5}$ and $x_1 \equiv (x + 2y + 2z)/3$ for our chosen values of a and b. There is both a dispersive and dissipative error, with the result converging in both phase and amplitude towards the undamped exact solution as the resolution is increased.

Figure 27 shows a convergence study on this problem, showing, as in Gardiner & Stone (2008), the $\mathcal{L}_1$ error as a function of the number of particles in the x-direction. The convergence is almost exactly second order. This is significant because we have performed the test with code defaults for all dissipation and divergence cleaning terms. This plot therefore demonstrates the second-order convergence of both the viscous and resistive dissipation in PHANTOM (see Sections 2.2.9 and 2.10.6). By comparison, the solution shown by Price & Monaghan (2005) (Figure 6 in their paper) was severely damped when artificial resistivity was applied.

As noted by Price & Monaghan (2005) and illustrated in Figure 12 of Price (2012a), this problem is unstable to the SPMHD tensile instability (e.g. Phillips & Monaghan 1985) in the absence of force correction terms since the plasma $\beta \equiv P/\frac{1}{2}B^2 \approx 0.2$. Our results demonstrate that the correction term (Section 2.10.2) effectively stabilises the numerical scheme without affecting the convergence properties.

5.6.2. *MHD shock tubes*

The classic Brio & Wu (1988) shock tube test generalises the Sod shock tube (Section 5.1.1) to MHD. It has provoked debate over the years (e.g. Wu 1988; Dai & Woodward 1994a; Falle & Komissarov 2001; Takahashi, Yamada, & Yamada 2013) because of the presence of a compound slow shock and rarefaction in the solution, which is stable only when the magnetic field is coplanar and there is no perturbation to the tangential (B_z) magnetic field (Barmin, Kulikovskiy, & Pogorelov 1996). Whether or not such solutions can exist in nature remains controversial (e.g. Feng et al. 2007). Nevertheless, it has become a standard benchmark for numerical MHD (e.g. Stone et al. 1992; Dai & Woodward 1994a; Balsara 1998; Ryu & Jones 1995). It was first used to benchmark SPMHD by Price & Monaghan (2004a, 2004b) and 1.5D results on this test with SPMHD, for comparison, can be found in e.g. Price & Monaghan (2005), Rosswog & Price (2007),

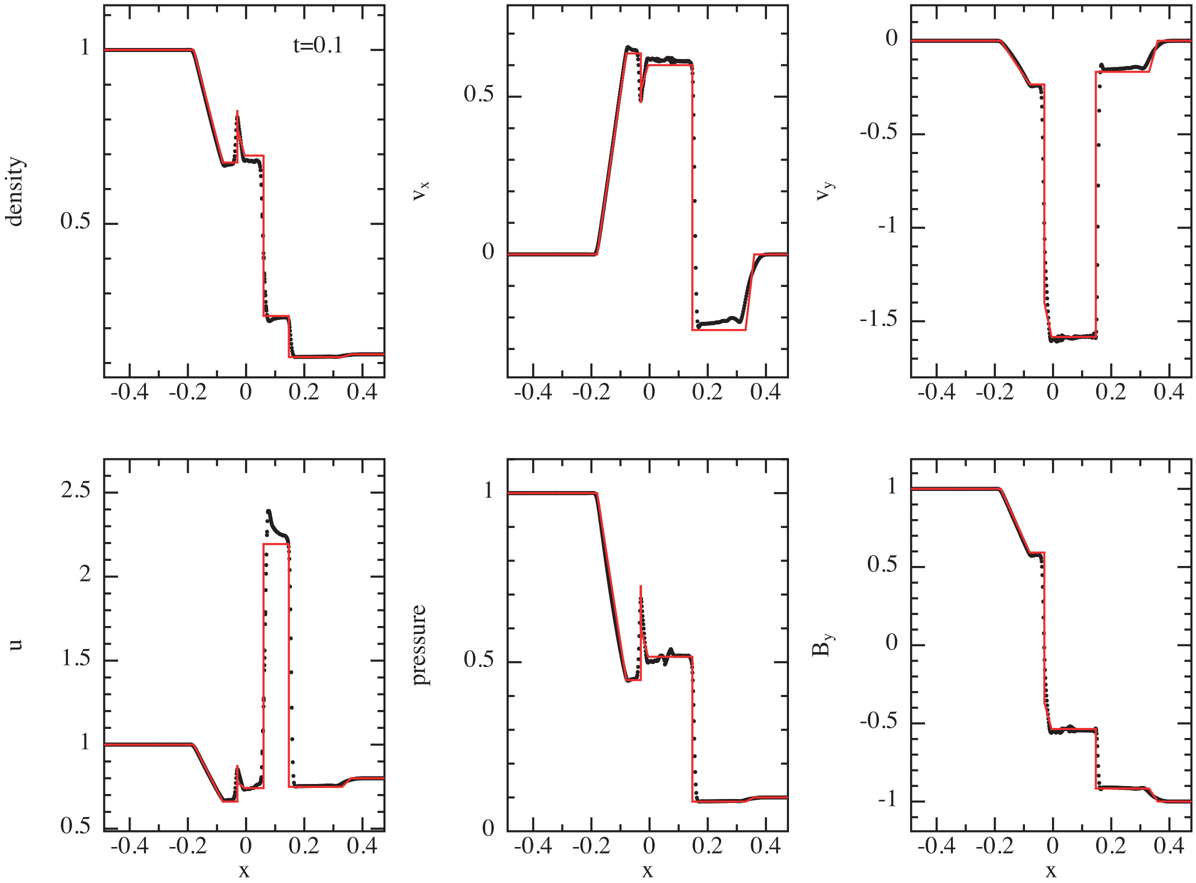

Figure 28. Results of the Brio & Wu (1988) shock tube test in 3D, showing projection of all particles (black dots) compared to the reference solution (red line). The problem is set up with $[\rho, P, B_y, B_z] = [1, 1, 1, 0]$ for $x \leqslant 0$ and $[\rho, P, B_y, B_z] = [0.125, 0.1, -1, 0]$ for $x > 0$ with zero initial velocities, $B_x = 0.75$ and $\gamma = 2$. The density contrast is initialised using equal mass particles placed on a close packed lattice with $256 \times 24 \times 24$ particles initially in $x \in [-0.5, 0]$ and $128 \times 12 \times 12$ particles initially in $x \in [0, 0.5]$.

Dolag & Stasyszyn (2009), Price (2010), and Vanaverbeke, Keppens, & Poedts (2014), with 2D versions shown in Price (2012a), Tricco & Price (2013), and Tricco et al. (2016a). We handle the boundary conditions by setting the first and last few planes of particles to be 'boundary particles' (Section A.2), meaning that the gas properties on these particles are fixed.

Figure 28 shows the results of the Brio & Wu (1988) problem using PHANTOM, performed in 3D with $256 \times 24 \times 24$ particles initially in $x \in [-0.5, 0]$ and $128 \times 12 \times 12$ particles initially in $x \in [0, 0.5]$ set on close packed lattices with purely discontinuous initial conditions in the other variables (see caption). The projection of all particles onto the x-axis are shown as black dots, while the red lines shows a reference solution taken from Balsara (1998). Figure 28 shows the results when a constant $\alpha^{\rm AV} = 1$ is employed, while Figure 29 shows the results with default code parameters, giving second-order dissipation away from shocks. For constant $\alpha^{\rm AV}$ (Figure 28), we find the strongest deviation from the reference solution is in v_x, with $\mathcal{L}_1 = 0.015$ and $\mathcal{L}_2 = 0.065$ at this resolution. The remaining $\mathcal{L}_2$ errors are within 5% of the reference solution, while the $\mathcal{L}_1$ norms are all smaller than 1.5% per cent in the other variables. Similar errors are found with code defaults (Figure 29), with $\mathcal{L}_2 = 0.074$ in v_x and $\mathcal{L}_1$ norms smaller than 1.6% in all variables. That is, our solutions are within 1.6% of the reference solution. Using the default Courant factor of 0.3, total energy is conserved to better than 0.5%, with maximum $|\Delta E|/|E_0| = 4.2 \times 10^{-3}$ up to $t = 0.1$.

Figure 30 shows the result of the 'seven-discontinuity' test from Ryu & Jones (1995). This test is particularly sensitive to over-dissipation by resistivity given the sharp jumps in the transverse magnetic and velocity fields. A reference solution with intermediate states taken from the corresponding table in Ryu & Jones (1995) is shown by the red lines for comparison. Here the boundary particles are moved with a fixed velocity in the x-direction. The largest deviation from the reference solution is in the v_y component, mainly due to the over-dissipation of the small spikes, with $\mathcal{L}_1 = 0.02$ and $\mathcal{L}_2 = 0.07$ at this resolution. The remaining $\mathcal{L}_2$ errors are within 3% of the reference solution, while the $\mathcal{L}_1$ norms are smaller than 0.9% in all other variables.

Finally, Figure 31 shows the results of shock tube 1a from Ryu & Jones (1995). This test is interesting because it has historically proven to be a difficult test for SPMHD codes in

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

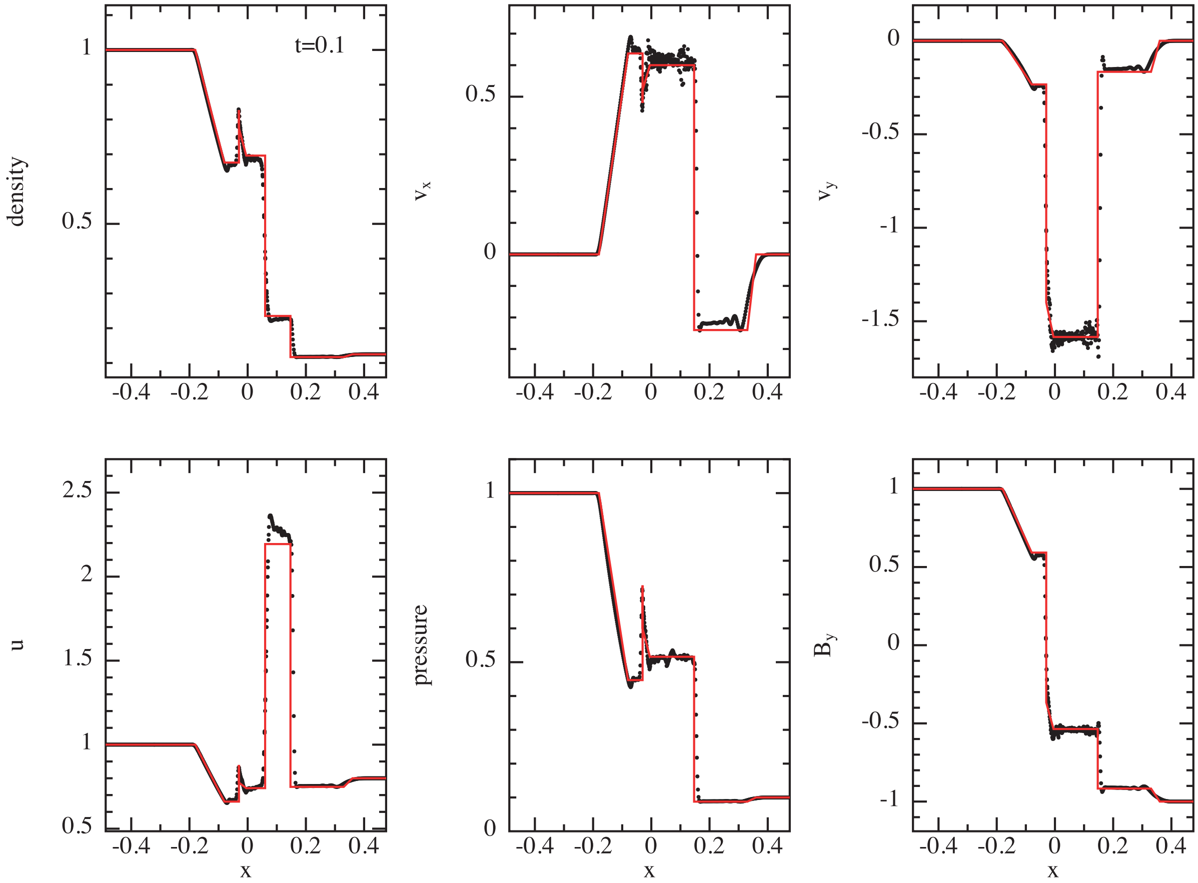

Figure 29. As in Figure 28 but using code defaults which give second-order convergence away from shocks. Some additional noise in the velocity field is visible, while otherwise the solutions are similar.

more than 1D. In particular, to obtain reasonable results on this problem, Price & Monaghan (2005) had to employ both an explicit shear viscosity term and a large neighbour number. Even then the jumps were found to show significant deviation from the analytic solution (see figure 10 in Price & Monaghan 2005). We did find that using $h_{\rm fac} = 1.2$ significantly improved the results of this test compared to our default $h_{\rm fac} = 1.0$ for the quintic kernel. Likewise, we found the results with the cubic spline kernel could be noisy. However, Figure 31 demonstrates that with only this change to the default code parameters, we can obtain results with $\mathcal{L}_1$ errors of better than 4.4% in v_y and less than 0.9% in all other variables at this resolution. The bottom left panel shows the errors induced in B_x, with the largest error ($\mathcal{L}_\infty$) only 0.6%. This is a substantial improvement over the 2D results shown in Price & Monaghan (2005).

5.6.3. Orszag–Tang vortex

The Orszag–Tang vortex (Orszag & Tang 1979; Dahlburg & Picone 1989; Picone & Dahlburg 1991) has been used widely to test astrophysical MHD codes (e.g. Stone et al. 1992; Ryu, Jones, & Frank 1995; Dai & Woodward 1998; Tóth 2000; Londrillo & Del Zanna 2000). Similar to our hydrodynamic tests, we perform a 3D version of the original 2D test problem, similar to the 'thin box' setup used by Dolag & Stasyszyn (2009). Earlier results on this test with 2D SPMHD can be found in Price & Monaghan (2005), Rosswog & Price (2007), Tricco & Price (2012, 2013), Tricco et al. (2016a), and Hopkins & Raives (2016), and in 3D by Dolag & Stasyszyn (2009) and Price (2010).

The setup is a uniform density, periodic box $x, y \in [-0.5, 0.5]$ with boundary in the z direction set to $\pm 2\sqrt{6}/n_x$, where n_x is the initial number of particles in x, in order to setup the 2D problem in 3D (c.f. Section 5.1.4). We use an initial plasma $\beta_0 = 10/3$, initial Mach number $\mathcal{M}_0 = v_0/c_{\rm s,0} = 1$, initial velocity field $[v_x, v_y, v_z] = [-\, v_0 \sin(2\pi y'), v_0 \sin(2\pi x'), 0]$, and magnetic field $[B_x, B_y, B_z] = [-\, B_0 \sin(2\pi y'), B_0 \sin(4\pi x'), 0]$, where $v_0 = 1$, $B_0 = 1/\sqrt{4\pi}$, $x' \equiv x - x_{\rm min}$, and $y' \equiv y - y_{\rm min}$; giving $P_0 = \frac{1}{2} B_0^2 \beta_0 \approx 0.133$ and $\rho_0 = \gamma P_0 \mathcal{M}_0 \approx 0.221$. We use an adiabatic EOS with $\gamma = 5/3$.

Figure 32 shows the results at $t = 0.5$ (top row) and at $t = 1$ (bottom) at resolutions of $n_x = 128$, 256, and 512 particles (left to right). At $t = 0.5$, the main noticeable change as the resolution is increased is that the shocks become more well defined, as does the dense filament consisting of material trapped in the reconnecting layer of magnetic field in the centre of the domain. This current sheet eventually becomes

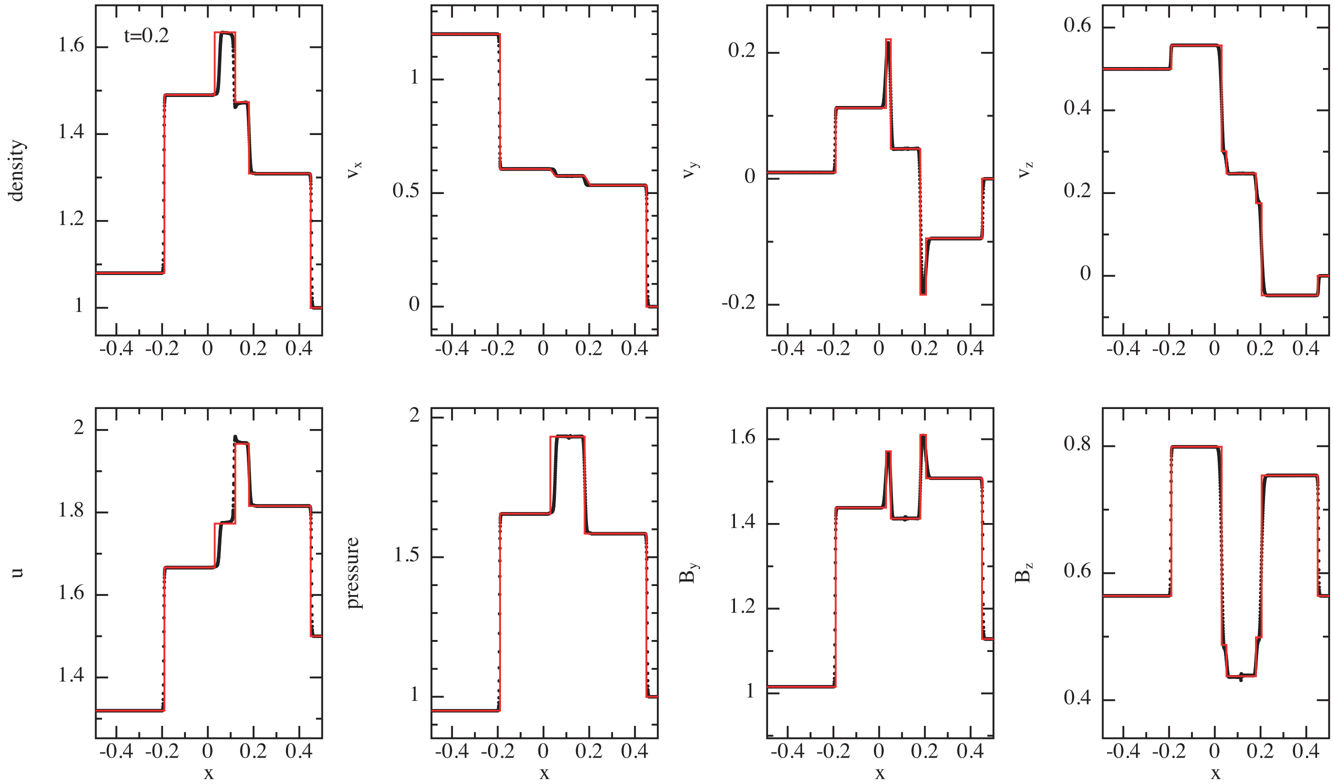

Figure 30. Results of the seven-discontinuity MHD shock tube test 2a from Ryu & Jones (1995) in 3D, showing projection of all particles (black dots) compared to the reference solution (red line). The problem is set up with $[\rho, P, v_x, v_y, v_z, B_x, B_y, B_z] = [1.08, 0.95, 1.2, 0.01, 0.5, 2/\sqrt{4\pi}, 3.6/\sqrt{4\pi}, 2/\sqrt{4\pi}]$ for $x \leqslant 0$ and $[\rho, P, v_x, v_y, v_z, B_x, B_y, B_z] = [1, 1, 0, 0, 0, 2/\sqrt{4\pi}, 4/\sqrt{4\pi}, 2/\sqrt{4\pi}]$ for $x > 0$ with $\gamma = 5/3$. The density contrast is initialised using equal mass particles placed on a close packed lattice with $379 \times 24 \times 24$ particles initially in $x \in [-0.5, 0]$ and $238 \times 12 \times 12$ particles initially in $x \in [0, 0.5]$.

unstable to the tearing mode instability (e.g. Furth, Killeen, & Rosenbluth 1963; Syrovatskii 1981; Priest 1985), seen by the development of small magnetic islands or ‘beads’ at $t = 1$ at high resolution (bottom right panel; c.f. Politano, Pouquet, & Sulem 1989). The appearance of these islands occurs only at high resolution and when the numerical dissipation is small [compare to the results using Euler potentials in 2D shown in Figure 13 of Tricco & Price (2012)], indicating that our implementation of artificial resistivity (Section 2.10.6) and divergence cleaning (Section 2.10.2) are effective in limiting the numerical dissipation.

One other feature worth noting is that the slight ‘ringing’ behind the shock fronts visible in the results of Price & Monaghan (2005) is absent from the low resolution calculation. This is because the Cullen & Dehnen (2010) viscosity switch does a better job of detecting and responding to the shock compared to the previous Morris & Monaghan (1997)-style switch used in that paper. It is also worth noting that the results on this test, in particular, the coherence of the shocks, are noticeably worse *without* artificial resistivity, indicating that a small amount of dissipation in the magnetic field *is* necessary to capture MHD shocks correctly in SPMHD (c.f. Price & Monaghan 2004a, 2005; Tricco & Price 2012).

Figure 33 shows horizontal slices of the pressure at $t = 0.5$, showing cuts along $y = 0.3125$ (top) and $y = 0.4277$ (bottom) following (e.g.) Londrillo & Del Zanna (2000) and Stone et al. (2008). The main difference at higher resolution is that the shocks become sharper and more well defined. Most of the smooth flow regions are converged with $n_x = 256$ (i.e. the red and black lines are indistinguishable), but the parts of the flow where dissipation is important are can be seen to converge more slowly. This is expected.

5.6.4. *MHD rotor problem*

Balsara & Spicer (1999) introduced the ‘MHD rotor problem’ to test the propagation of rotational discontinuities. Our setup follows Tóth (2000)’s ‘first rotor problem’ as used by Price & Monaghan (2005), except that we perform the test in 3D. A rotating dense disc of material with $\rho = 10$ is set up with cylindrical radius $R = 0.1$, surrounded by a uniform periodic box $[x, y] \in [-0.5, 0.5]$ with the z boundary set to $[-\sqrt{6}/(2n_x), \sqrt{6}/(2n_x)]$, or 12 particle spacings on a close packed lattice. The surrounding medium has density $\rho = 1$. Initial velocities are $v_{x,0} = -v_0(y - y_0)/r$ and $v_{y,0} = v_0(x - x_0)/r$ for $r < R$, where $v_0 = 2$ and $r = \sqrt{x^2 + y^2}$. The initial pressure $P = 1$ everywhere while the initial magnetic field is given by $[B_x, B_y, B_z] = [5/\sqrt{4\pi}, 0, 0]$ with $\gamma = 1.4$. We set up the initial density contrast unsmoothed, as in Price & Monaghan (2005), by setting up two uniform lattices of

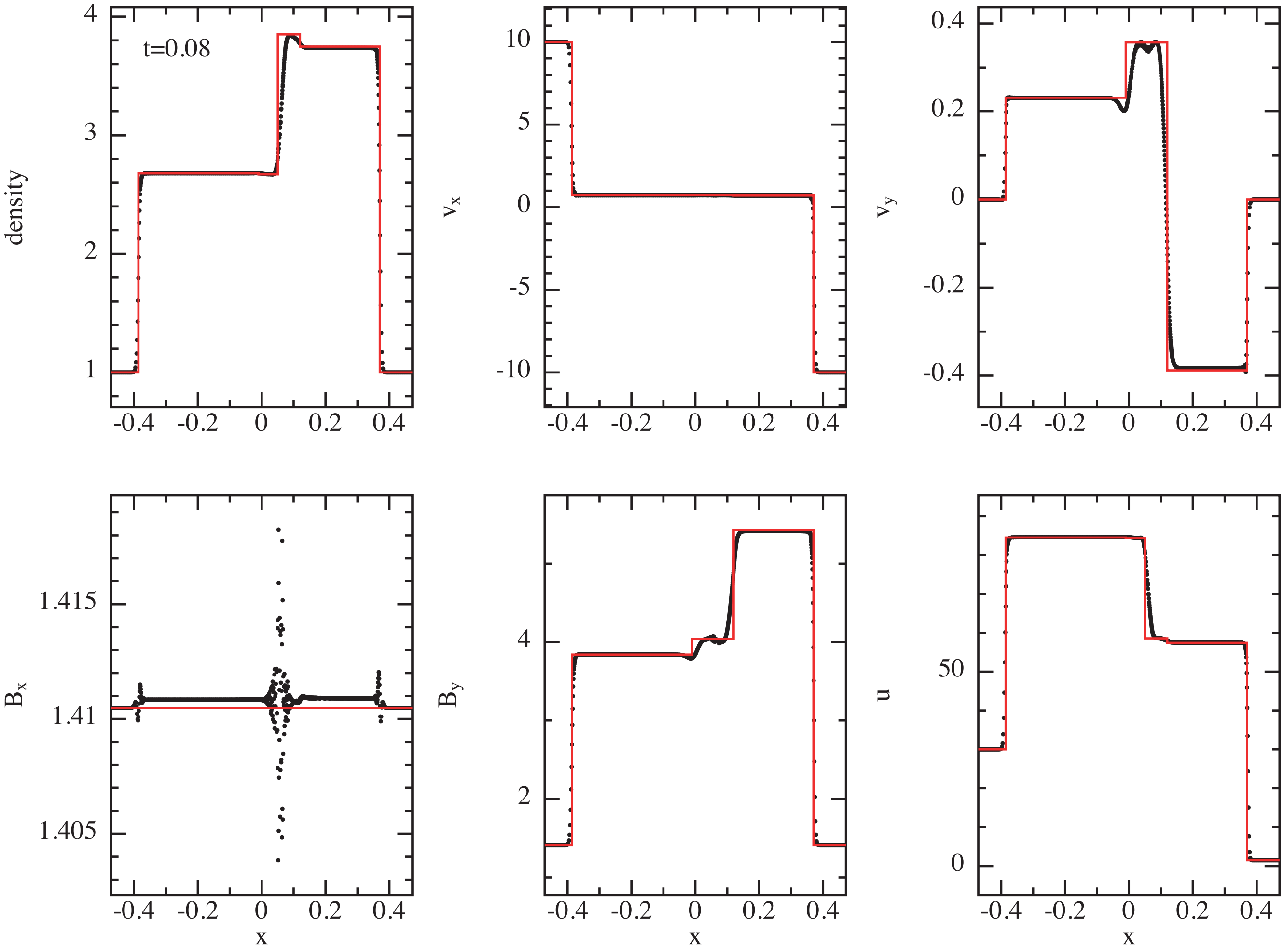

Figure 31. Results of the MHD shock tube test 1a from Ryu & Jones (1995) in 3D, showing projection of all particles (black dots) compared to the reference solution (red line). The problem is set up with $[\rho, P, v_x, v_y, v_z, B_x, B_y, B_z] = [1.0, 20.0, 10, 0, 0, 5/\sqrt{4\pi}, 5/\sqrt{4\pi}, 0]$ for $x \leqslant 0$ and $[\rho, P, v_x, v_y, v_z, B_x, B_y, B_z] = [1, 1, -10, 0, 0, 5/\sqrt{4\pi}, 5/\sqrt{4\pi}, 0]$ for $x > 0$ with $\gamma = 5/3$. We show results using $652 \times 12 \times 12$ particles. This has historically proven difficult for SPMHD codes. We find $\mathcal{L}_1$ within 1% of the reference solution except in v_y (5%).

particles masked to the initial cylinder, with the particle spacing adjusted inside the cylinder by the inverse cube root of the density contrast. At a resolution of $n_x = 256$ particles for the closepacked lattice, this procedure uses 1 145 392 particles, equivalent to a 2D resolution of $\sim 300^2$, inbetween the 200^2 results shown in Tóth (2000) and Price & Monaghan (2005) and the 400^2 used in Stone et al. (2008).

Figure 34 presents the results of this test, showing 30 contours in density, pressure, Mach number, and magnetic pressure using limits identical to those given in Tóth (2000). The symmetry of the solution is preserved by the numerical scheme and the discontinuities are sharp, as discussed in Stone et al. (2008). The contours we obtain with PHANTOM are noticeably less noisy than the earlier SPMHD results given in Price & Monaghan (2005), a result of the improvement in the treatment of dissipation and divergence errors in SPMHD since then (c.f. Section 2.10; see also recent results in Tricco et al. 2016a).

5.6.5. *Current loop advection*

The current loop advection test was introduced by Gardiner & Stone (2005, 2008), and regarded by Stone et al. (2008) as the most discerning of their code tests. We perform this test in 3D, as in the 'first 3D test' from Stone et al. (2008) by using a thin 3D box with non-zero v_z. The field setup is with a vector potential $A_z = A_0(R - r)$ for $r < R$, giving $B_x = -A_0 y/r$, $B_y = A_0 x/r$ and $B_z = 0$, where $r = \sqrt{x^2 + y^2}$, $R = 0.3$ and we use $A_0 = 10^{-3}$, $\rho_0 = 1$, $P_0 = 1$, and an adiabatic EOS with $\gamma = 5/3$. We use a domain $[x, y, z] \in [-1 : 1, -0.5 : 0.5, -\sqrt{6}/(2n_x) : \sqrt{6}/(2n_x)]$ with $[v_x, v_y, v_z] = [2, 1, 0.1/\sqrt{5}]$. The test is difficult mainly because of the cusp in the vector potential gradient at $r = R$ leading to a cylindrical current sheet at this radius. The challenge is to advect this infinite current without change (in numerical codes the current is finite but with a magnitude that increases with resolution). We choose the resolution to be comparable to Stone et al. (2008).

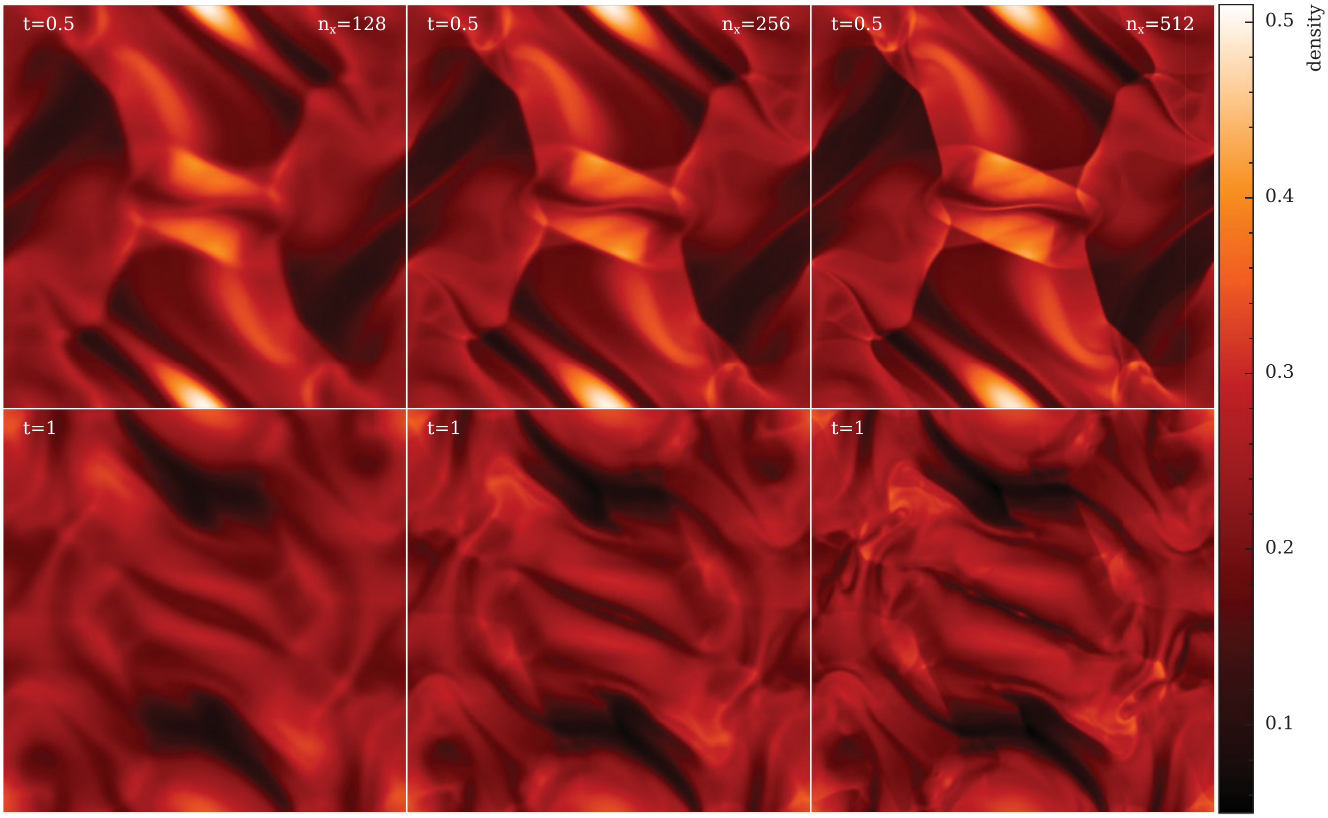

Figure 32. Density in a $z = 0$ cross section of the Orszag-Tang vortex test performed in 3D. Results are shown at $t = 0.5$ (top) and $t = 1$ (bottom) at a resolution of 128 × 148 × 12, 256 × 296 × 12 and 512 × 590 × 12 particles (left to right). Compare e.g. to Figure 4 in Dai & Woodward (1994b) or Figure 22 in Stone et al. (2008), while improvements in the SPMHD method over the last decade can be seen by comparing to Figure 14 in Price & Monaghan (2005).

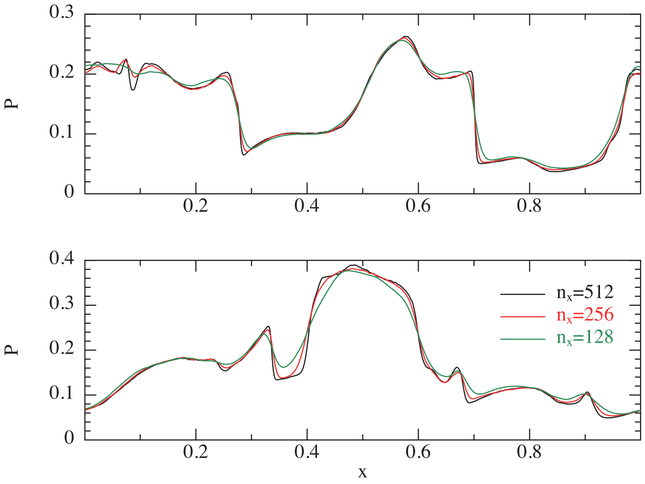

Figure 33. Horizontal slices of pressure shown at t=0.5 in the Orszag–Tang vortex test. We show cuts along $y = 0.3125$ (top) and $y = 0.4277$ (bottom) in the $z = 0$ plane for three different numerical resolutions (see legend).

For SPMHD, this is mainly a test of the shock dissipation and divergence cleaning terms, since in the absence of these terms the advection can be computed to machine precision [c.f. 2D results shown in Rosswog & Price (2007) and Figure 11 of Price (2012a), shown after one thousand crossings of the computational domain]. Figure 35 shows the results of this test in PHANTOM with 128 × 74 × 12 particles after two box crossings, computed *with all dissipation and divergence cleaning terms switched on*, precisely as in the previous tests including the shock tubes (Sections 5.6.2–5.6.4). Importantly, our implementation of artificial resistivity (Section 2.10.6) guarantees that the dissipation is identically zero when there is no relative velocity between the particles, meaning that simple advection of the current loop is not affected by numerical resistivity. However, the problem remains sensitive to the divergence cleaning (Section 2.10.2), in particular to any spurious divergence of $\boldsymbol{B}$ that is measured by the SPMHD divergence operator, (174). For this reason, the results using the quintic kernel, (19), are substantially better than those using the cubic spline, because the initial measurement of $\nabla \cdot \boldsymbol{B}$ is smaller and so the evolution is less affected by the divergence cleaning.

5.6.6. MHD blast wave

The MHD blast wave problem consists of an over-pressurised central region that expands preferentially along the strong magnetic field lines. Our setup uses the 3D initial conditions of Stone et al. (2008), which follows from the work of

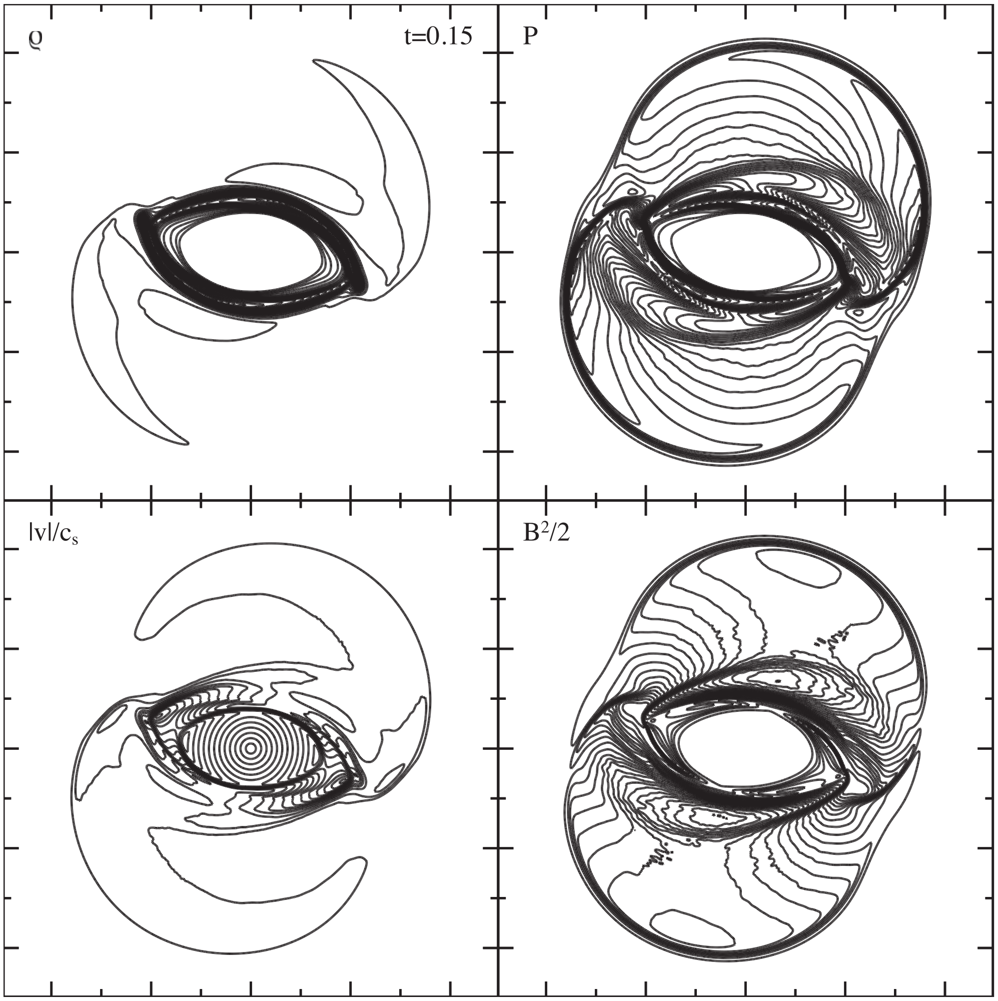

Figure 34. Density, pressure, Mach number, and magnetic pressure shown in a $z = 0$ cross section at $t = 0.15$ in the 3D MHD rotor problem, using $n_x = 256$, equivalent to $\sim 300^2$ resolution elements in 2D. The plots show 30 contours with limits identical to those given by Tóth (2000); $0.483 < \rho < 12.95$, $0.0202 < P < 2.008$, $0 < |v|/c_s < 1.09$, and $0 < \frac{1}{2}B^2 < 2.642$.

Londrillo & Del Zanna (2000) and Balsara & Spicer (1999). For a recent application of SPMHD to this problem, see Tricco & Price (2012) and Tricco et al. (2016a). Set in a periodic box $[x, y, z] \in [-0.5, 0.5]$, the fluid has uniform $\rho = 1$ and $\boldsymbol{B} = [10/\sqrt{2}, 0, 10/\sqrt{2}]$. The pressure is set to $P = 1$, using $\gamma = 1.4$, except for a region in the centre of radius $R = 0.125$ which has its pressure increased to $P = 100$. This yields initial plasma beta $\beta = 2$ inside the blast and $\beta = 0.02$ outside. The particles are arranged on a close-packed triangular lattice using $n_x = 256$.

Figure 36 shows slices through $y = 0$ of density, pressure, magnetic energy density, and kinetic energy density, which may be directly compared to results in Gardiner & Stone (2008). The shock positions and overall structure of the blast wave in all four variables show excellent agreement with the results shown in their paper. The main difference is that their solution appears less smoothed, suggesting that the overall numerical dissipation is lower in ATHENA.

5.6.7. Balsara–Kim supernova-driven turbulence

We reproduce the 'test problem' of Balsara & Kim (2004) (hereafter BK04) modelling supernova-driven turbulence in the interstellar medium. BK04 used this test to argue strongly against the use of divergence cleaning for problems involving strong shocks. Specifically, they compared three different divergence cleaning schemes against a constrained transport method, finding that divergence cleaning was unusable for such problems, with all three divergence cleaning schemes producing strong temporal fluctuations in magnetic energy during the growth phase of the supernova-driven dynamo. The problems were attributed to issues with the non-locality of divergence cleaning.

We follow the setup in BK04 as closely as possible, but several issues make a direct comparison difficult. Chief among these is their use of a physical ISM cooling prescription. We implement a similar algorithm (Section 2.14), but our cooling prescription is not identical (e.g. our implementation includes

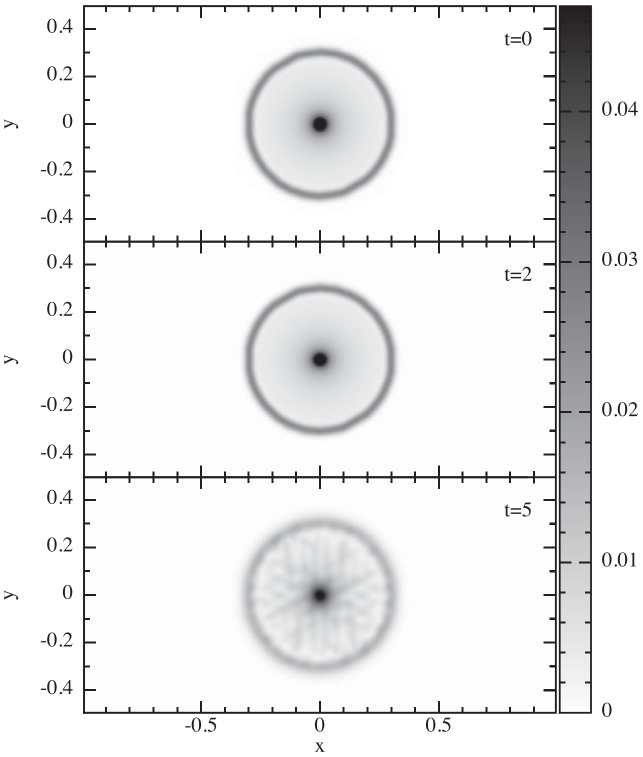

Figure 35. Magnitude of the current density $|\nabla \times \boldsymbol{B}|$ in the current loop advection test performed in 3D, showing comparing the initial conditions (top) to the result after two (middle) and five (bottom) crossings of the box, using 128 × 74 × 12 particles. *Full dissipation, shock capturing, and divergence cleaning terms were applied for this test*, without which the advection is exact to machine precision. The advection is affected mainly by the divergence cleaning acting on the outer (infinite) current sheet.

live chemistry and thus the possibility for a cold phase of the ISM, which theirs does not). Second, they give the setup parameters for the problem in computational units, but the use of ISM cooling requires the physical units of the problem to be specified. Since these are not stated in their paper, one must guess the units by reading descriptions of the same problem in physical units given in Kim, Balsara, & Mac Low (2001), Balsara et al. (2004), and Mac Low et al. (2005).

We set up the problem as follows. Particles are initialised on a close-packed lattice in a periodic box with $x, y, z \in [-0.1, 0.1]$, with $\rho = 1$ and the initial thermal pressure set to $P = 0.3$ (all in code units, as specified in BK04). We infer a length unit of kpc since this is described as a '200 pc^3 box' in their other papers. We employ an adiabatic EOS with $\gamma = 5/3$ and turn on interstellar cooling and chemistry in the code with default initial abundances (i.e. atomic Hydrogen everywhere). We choose the mass unit such that a density of 2.3×10^{-23} g cm^{-3}, as described in Kim et al. (2001), corresponds to $\rho = 1$ in code units as described in BK04. Finally, we set the time unit such that $G = 1$ in code units (a common choice, even though gravity is not involved in the problem). We compute the problem using resolutions of 64 × 74 × 78 particles and 128 × 148 × 156 particles.

Supernovae are injected into the simulation every 0.00125 in code units at the positions listed in Table 1 of BK04. We infer this to correspond to the '12 × Galactic' rate described in Balsara et al. (2004), i.e. 12 times faster than 1 per 1.26 Myr. We inject supernovae following the description in BK04 by setting the pressure to $P = 13649.6$ on particles within a distance of 0.005 code units from the injection site (corresponding to a radius of 5 pc in physical units). Our choice of units means that this corresponds to an energy injection within a few percent of 10^{51} ergs in physical units, corresponding to the description given in their other papers. However, this is only true if the density equals the initial value, since BK04 specify pressure rather than the energy. We follow the description in BK04 (i.e. we set the pressure), even though this gives an energy not equal to 10^{51} ergs if injected in a low or high density part of the computational domain.

The initial magnetic field is uniform in the x-direction with $B_x = 0.056117$, as stated in BK04. We could not reconcile this with the magnetic energy plotted in their paper, which show an initial magnetic energy of 10^{-6}. Nor could we reconcile this with the statement in Balsara et al. (2004) that 'the magnetic energy is 10^{-6} times smaller than the thermal energy'. So any comparison is approximate. Nevertheless, BK04 state that the problems caused by divergence cleaning are not dependent on specific details of the implementation.

Figure 37 shows the evolution of the column density in the lowest resolution calculation using 64 × 74 × 78 particles. The combination of supernovae injection and cooling drives turbulence and significant structure in the density field.

To specifically address the issues found by BK04, Figure 38 shows a cross-section slice of the magnetic pressure at a time similar to the one shown in Figure 3 of BK04 (they do not indicate which slice they plotted, we chose $z = 0.0936$). The magnetic pressure in the interior of the supernovae shells is smooth, and does not display any of the large-scale artefacts of the type found in their paper.

Figure 39 shows the time evolution of the magnetic energy in the low-resolution calculation. The magnetic energy rises monotonically up to $t \approx 0.02$ before the magnetic energy saturates, similar to what was found by BK04 for their staggered mesh/constrained transport scheme (compare with Figure 2 in their paper). There are no large temporal fluctuations in the magnetic energy of the kind they report for their divergence cleaning methods.

In summary, the results we obtain for the magnetic field energy and structure within supernova-driven turbulence in the interstellar medium match closest the constrained transport result of BK04. There is no evidence that our simulations experience the numerical issues encountered by BK04 with their divergence cleaning schemes, suggesting that the problems they reported are primarily code dependent rather than being fundamental. The most probable reason our results are of the same quality as BK04's constrained transport result is our use of constrained divergence cleaning, which guarantees that energy removed from the magnetic field is negative definite. A follow-up study to examine this in more depth

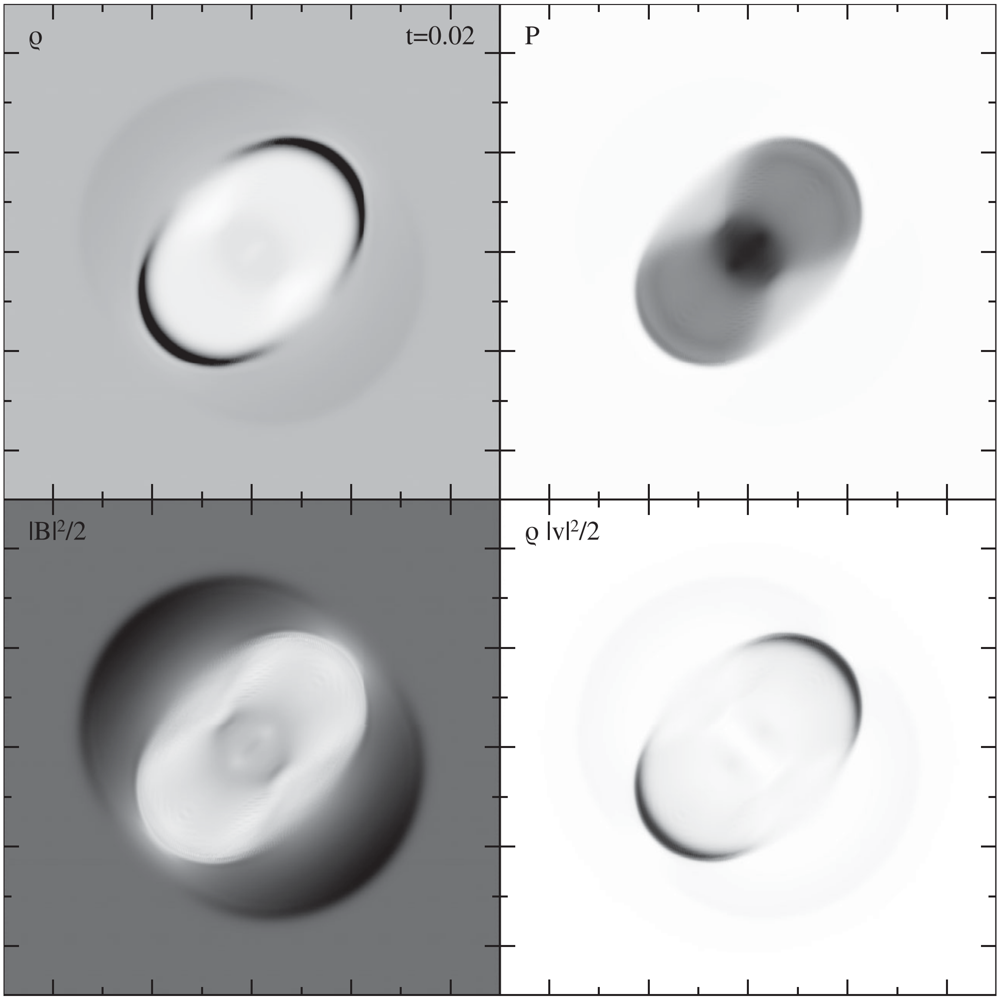

Figure 36. Slices through $y = 0$ for the MHD blast wave problem showing density (top left), gas pressure (top right), magnetic energy density (bottom left), and kinetic energy density (bottom right). The plot limits are $\rho \in [0.19, 2.98]$, $P \in [1, 42.4]$, [25.2, 64.9] for the magnetic energy density and [0, 33.1] for the kinetic energy density. These are directly comparable to Figure 8 in Gardiner & Stone (2008).

would be worthwhile. For our present purposes, we can conclude that our divergence cleaning algorithm is sufficiently robust—even in 'real-world' problems—to be useful in practice (see also Section 6.3).

5.7. Non-ideal MHD

The following tests demonstrate the non-ideal MHD algorithms (Section 2.11). We adopt periodic boundary conditions for both tests, initialising the particles on a close-packed lattice with an isothermal EOS, $P = c_s^2\rho$, and using the C^4 Wendland kernel.

5.7.1. Wave damping test

To test ambipolar diffusion in the strong coupling approximation, we follow the evolution of decaying Alfvén waves, as done in (e.g.) Choi, Kim, & Wiita (2009) and Wurster et al. (2014).

In arbitrary units, the initial conditions are a box of size $L_x \times \frac{\sqrt{3}}{2} L_x \times \frac{\sqrt{6}}{2} L_x$ with $L_x = 1$, a density of $\rho = 1$, magnetic field of $\boldsymbol{B} = B_0\hat{\boldsymbol{x}}$ with $B_0 = 1$, sound speed of $c_s = 1$, and velocity of $\boldsymbol{v} = v_0 \sin(kx)\hat{\boldsymbol{z}}$, where $k = 2\pi/L_x$ is the wave number, and $v_0 = 0.01 v_A$, where v_A is the Alfvén velocity. We adopt an ambipolar diffusion coefficient of $\eta_{AD} = 0.01 v_A^2$. All artificial dissipation terms are turned off. We use $n_x = 128$ particles.

The solution to the dispersion relation for Alfvén waves (Balsara 1996) is

$$\omega^2 + \eta_{AD} k^2 \omega i - v_A^2 k^2 = 0, \tag{323}$$

where $\omega = \omega_R + \omega_I i$ is the complex angular frequency of the wave, giving a damped oscillation in the form

$$h(t) = h_0 \left|\sin(\omega_R t)\right| e^{\omega_I t}. \tag{324}$$

In our test, $h(t)$ corresponds the root mean square of the magnetic field in the z-direction, $\langle B_z^2 \rangle^{1/2}$, and $h_0 = v_0 B_0/(v_A\sqrt{2})$.

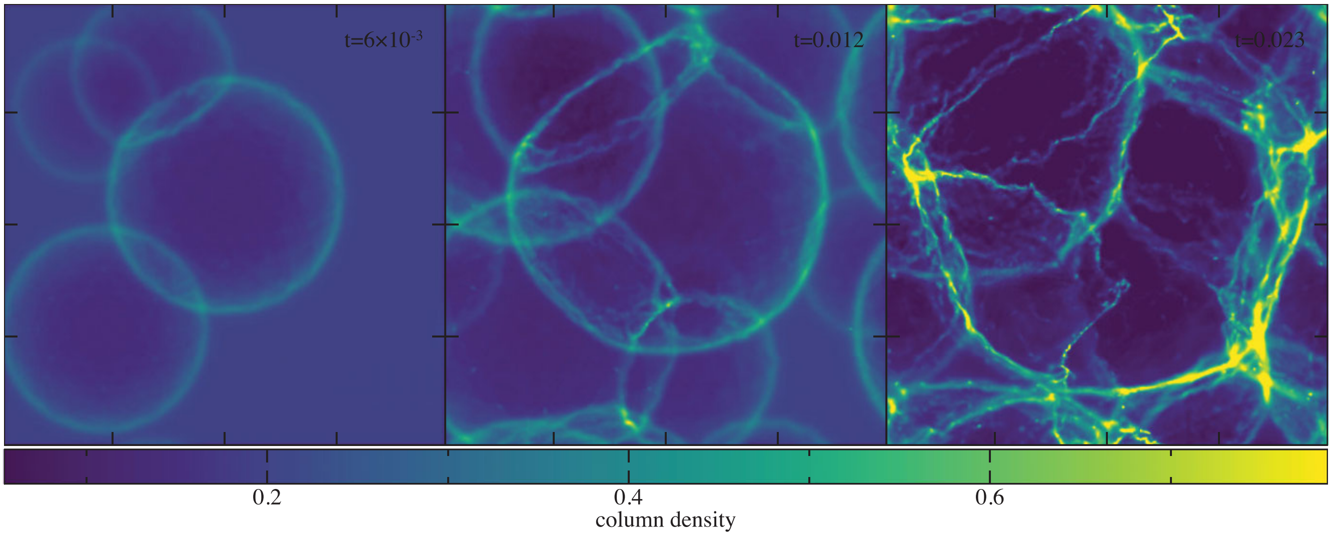

Figure 37. Balsara–Kim supernova-driven turbulence, showing column density at three different times at a resolution of 64 × 74 × 78 particles. Supernovae are injected every 0.00125 in code units, leading to a series of interacting blast waves. Interstellar chemistry and cooling is turned on, producing a dense filaments in a turbulent interstellar medium.

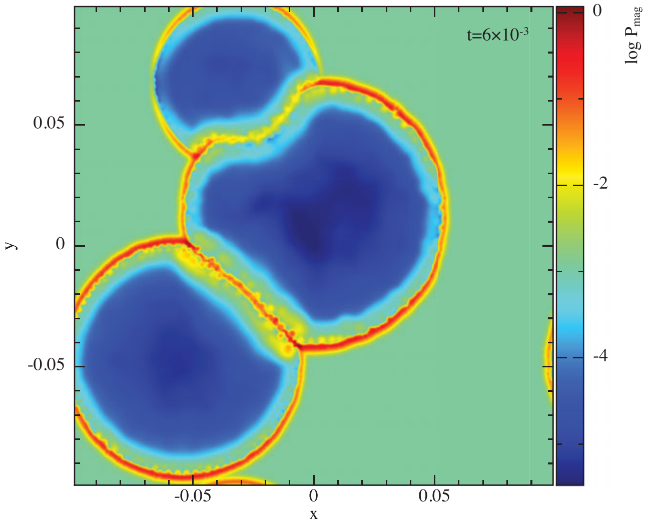

Figure 38. Cross-section slice of magnetic pressure at $t = 0.006$ in the Balsara–Kim supernova-driven turbulence test using 128 × 148 × 156 particles. No large-scale artefacts in magnetic energy are visible, indicating that the simulation is not corrupted by divergence cleaning.

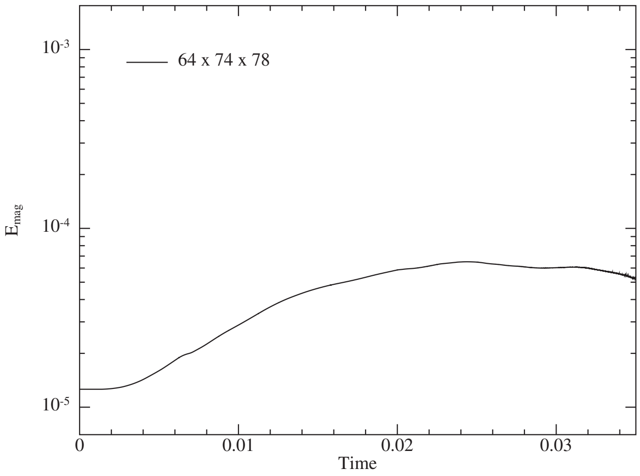

Figure 39. Magnetic energy as a function of time in the Balsara–Kim supernova-driven turbulence problem. The magnetic energy increases monotonically by approximately an order of magnitude before reaching its saturation value at $t \approx 0.02$. There are no spurious spikes in magnetic energy caused by divergence cleaning, in contrast to what was found by Balsara & Kim (2004).

Figure 40 shows the time evolution of $\langle B_z^2 \rangle^{1/2}$ to $t = 5$ for both the numerical results (blue line) and the analytic solution (red line). At the end of the test, the $\mathcal{L}_2$ error is 7.5×10^{-5} (evaluated at intervals of $\mathrm{d}t = 0.01$), demonstrating close agreement between the numerical and analytical results.

Given that, by design, there is motion of the particles and that we have excluded artificial dissipation, the particles tend to 'break' from the initial lattice. For both the M_6 quintic kernel and the C^4 Wendland kernel on a cubic lattice, the particles fall off the lattice at $t \approx 0.75$. Prior to this, however, the $\mathcal{L}_2$ error is smaller than that calculated using the C^4 Wendland kernel and a close-packed lattice, thus there is a trade off between accuracy and long-term stability.

5.7.2. Standing shock

To test the Hall effect, we compare our solutions against the 1D isothermal steady-state equations for the strong Hall effect regime. The numerical solution to this problem is given in Falle (2003) and O'Sullivan & Downes (2006), and also summarised in Appendix C1.2 of Wurster et al. (2016).

The left- and right-hand side of the shock are initialised with $(\rho_0,\ v_{\mathrm{x},0},\ v_{\mathrm{y},0},\ v_{\mathrm{z},0},\ B_{\mathrm{x},0},\ B_{\mathrm{y},0},\ B_{\mathrm{z},0}) = (1.7942,$ $-0.9759, -0.6561, 0.0, 1.0, 1.74885, 0.0)$ and $(1.0, -1.751,$

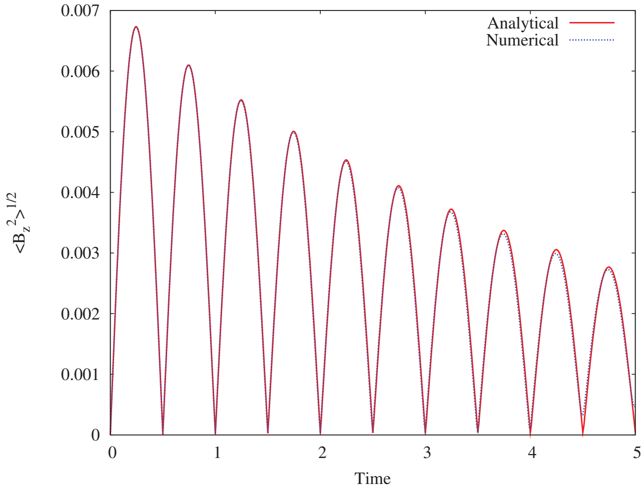

Figure 40. Wave damping test showing the decay of Alfvén waves in the presence of ambipolar diffusion, using the coefficient $\eta_{\rm AD} = 0.01 v_{\rm A}^2$. The $\mathcal{L}_2$ error between the analytic and numerical solution is 7.5×10^{-5}.

0.0, 0.0, 1.0, 0.6, 0.0), respectively, with the discontinuity at $x = 0$. We use boundary particles at the x-boundary, superseding the periodicity in this direction. To replicate inflowing boundary conditions in the x-direction, when required, the initial domain of interest $x_l < x < x_r$ is automatically adjusted to $x'_l < x < x'_r$, where $x'_r = x_r - v_0 t_{\rm max}$, where $t_{\rm max}$ is the end time of the simulation. Note that for inflowing conditions, x_r and v_0 will have opposite signs. Thus, at the end of the simulation, the entire range of interest will still be populated with particles.

The non-ideal MHD coefficients are $\eta_{\rm OR} = 1.12 \times 10^{-12}$, $\eta_{\rm HE} = -3.53 \times 10^{-2} B$, and $\eta_{\rm AD} = 7.83 \times 10^{-3} v_{\rm A}^2$. We include all artificial dissipation terms using their default settings. We initialise $512 \times 14 \times 15$ particles in $x < 0$ and $781 \times 12 \times 12$ particles in $x \geqslant 0$, with the domain extending from $x_l = x'_l = -2$ to $x_r = 2$ with $x'_r = 3.75$.

Figure 41 shows $v_{\rm x}$ and $B_{\rm y}$ for both the numerical and analytical results, which agree to within 3% at any given position.

On the left-hand side of the shock interface, the numerical results are lower than the analytical solution because of the artificial dissipation terms, which are required to properly model a shock. These results do not depend on either the kernel choice or the initial particle lattice configuration.

5.8. Self-gravity

5.8.1. Polytrope

The simplest test of self-gravity is to model a spherical polytrope in hydrostatic equilibrum. Similar tests have been shown for SPH codes dating back to the original papers of Gingold & Monaghan (1977, 1978, 1980). Modern calculations have used these simple models in more complex applications, ranging from common envelope evolution (Iaconi et al. 2017) to tidal disruption events (Coughlin & Nixon 2015; Coughlin et al. 2016b, 2016a; Bonnerot et al. 2016, 2017).

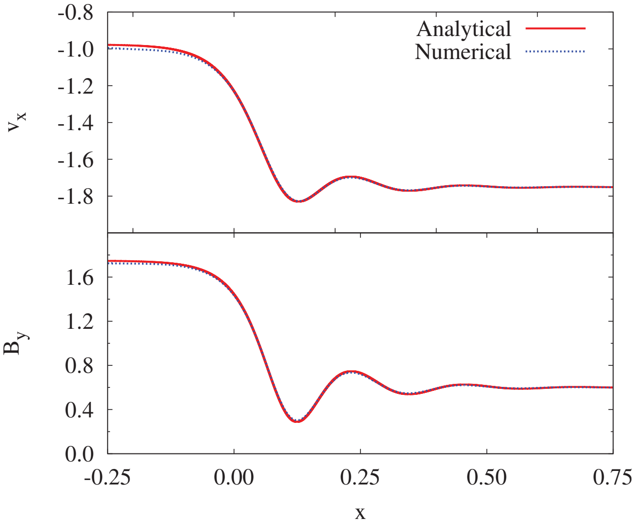

Figure 41. Hall-dominated standing shock using $\eta_{\rm OR} = 1.12 \times 10^{-12}$, $\eta_{\rm HE} = -3.53 \times 10^{-2} B$, and $\eta_{\rm AD} = 7.83 \times 10^{-3} v_{\rm A}^2$. The numerical and analytical results agree to within 3% everywhere.

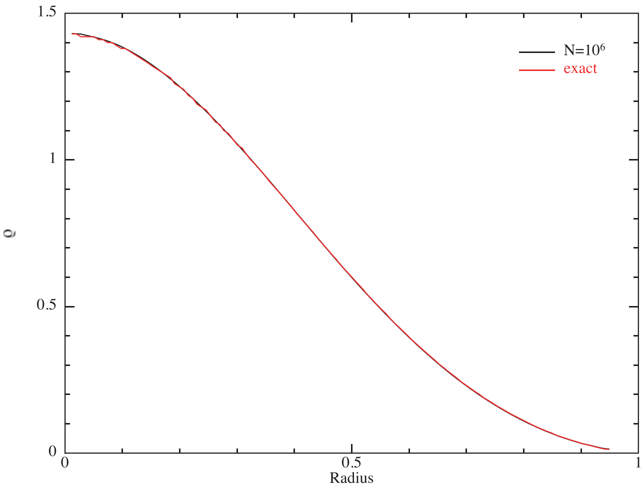

Figure 42. Polytrope static structure using 10^6 particles (black), compared to the exact solution (red), shown at $t = 100$.

The EOS is $P = K\rho^\gamma$, with $\gamma = 1 + 1/n$ where n is the polytropic index. The exact hydrostatic solution is given by

$$\frac{\gamma K}{4\pi G(\gamma - 1)} \frac{{\rm d}^2}{{\rm d}r^2} \left(r\rho^{\gamma-1} \right) + r\rho = 0. \tag{325}$$

Our initial setup uses a solution scaled to a radius $R = 1$, for a polytropic index of $n = 3/2$, corresponding to $\gamma = 5/3$, with $K = 0.4244$. We solve (325) numerically. We place the particles initially on a hexagonal close-packed lattice, truncated to a radius of $R = 1$, which we then stretch map (see Section 3.2) such that the initial radial density profile matches the exact solution.

The relaxation time depends on the initial density profile and on how far the initial particle configuration is from equi-

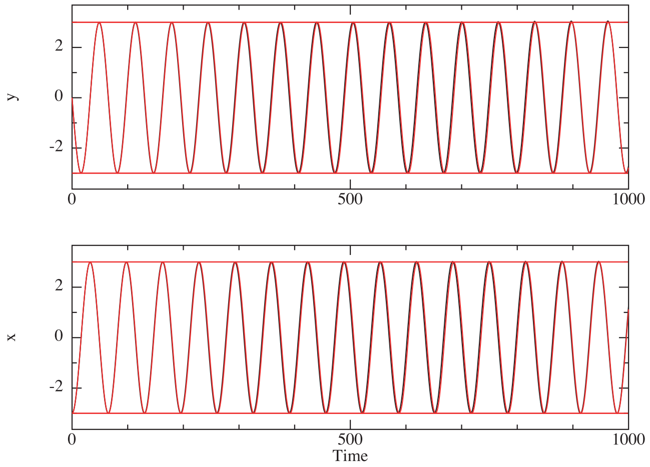

Figure 43. y- and x-positions of the centre of mass of one star of a binary system (top and bottom, respectively). Each star is a polytrope with mass and radius of unity, and $N = 10^4$ particles; the initial separation is 6.0 in code units. After ~15 orbits, the separation remains within 1%, and the period remains constant within the given time resolution. The red line represents the analytical position with respect to time, and the black line represents the numerical position.

librium. Figure 42 shows the solution at $t = 100$ in code units. The polytrope relaxes within a few dynamical times, with only a slight rearrangement of the particles from the stretched lattice. The density profile at all times is equal to within 3% of the exact solution for $r \leqslant 0.7$ and the polytrope remains in hydrostatic equilibrium.

Once the static solution is obtained, we tested the energy conservation by giving the star a radial perturbation. That is, we applied a velocity perturbation of the form $v_r = 0.2r$ to the $N \approx 10^5$ model, and evolved the polytrope for 100 time units. We turned off the artificial viscosity for this test. The total energy—including contributions from thermal, kinetic, and gravitational energy—remained conserved to within 3%.

5.8.2. Binary polytrope

Next, we placed two initially unrelaxed, identical polytropes, each with $N = 10^4$ particles in a circular orbit around each other with a separation of $6R$ and evolved for ~15 orbits (1 000 code units). Figure 43 demonstrates that the separation remains within 1% of the initial separation over 15 orbits, and that the orbital period remains constant. After the initial relaxation, total energy is conserved to within 0.06%.

The stars are far enough apart that any tidal deformation as they orbit is insignificant. The final density profile of each star agrees with the expected profile within 3% for $r \leqslant 0.7$.

5.8.3. Evrard collapse

A more complex test, relevant especially for star formation, is the so-called 'Evrard collapse' (Evrard 1988) modelling the adiabatic collapse of a cold gas sphere. It has been used many times to test SPH codes with self-gravity, e.g. Hernquist & Katz (1989), Steinmetz & Mueller (1993), Thacker et al. (2000), Escala et al. (2004), and many others.

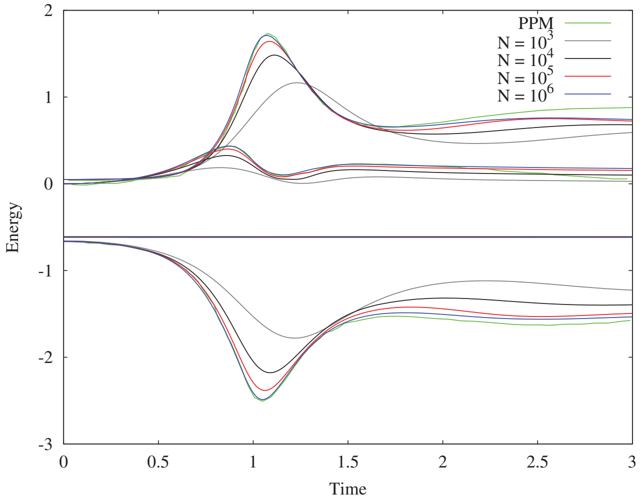

Figure 44. Thermal, kinetic, total, and potential energies as a function of time during the Evrard collapse (Evrard 1988). Green lines are calculated using a 1D PPM code, taken from Figure 6 of Steinmetz & Mueller (1993), while remaining colours show SPH simulations of different resolutions. Both energy and time are given in code units, where $R = M = G = 1$.

Following the initial conditions of Evrard (1988), we setup the particles initially in a sphere of radius $R = 1$ and mass $M = 1$, with density profile

$$\rho(r) = \frac{M(R)}{2\pi R^2}\frac{1}{r}. \tag{326}$$

The density profile is created using the same stretch mapping method as for the polytrope (see Section 5.8.1). The sphere is initially isothermal, with the specific internal energy set to $u = 0.05GM/R$ with an adiabatic index of $\gamma = 5/3$. The sphere initially undergoes gravitational collapse.

In the literature, the results of the Evrard collapse are typically normalised to a characteristic value. Here, we simply show the results in code units (Section 2.2.3), since these units already represent a normalised state. A distance unit of $R = 1$ and mass unit $M(R) = 1$ is adopted, with the time unit set such that $G \equiv 1$, where G is the gravitational constant.

Figure 44 shows the kinetic, thermal, total, and potential energies as a function of time, at four different numerical resolutions. The green line shows the reference solution, computed using a 1D piecewise parabolic method (PPM) code using 350 zones, which we transcribed from Figure 6 of Steinmetz & Mueller (1993).

As the number of particles increases, the energies for $t \lesssim 1.5$ converge to the results obtained from the PPM code. At $t \gtrsim 2$, the SPH results appear to converge to energies that differ slightly from the PPM code. Given that we are not able to perform a comparable convergence study with the PPM code, we are unable to assess whether or not this discrepancy is significant.

Figure 45 shows enclosed mass, density, thermal energy, and radial velocity as a function of radius at $t = 0.77$, where the SPH results may be compared to the PPM results presented by Steinmetz & Mueller (1993) shown with the red

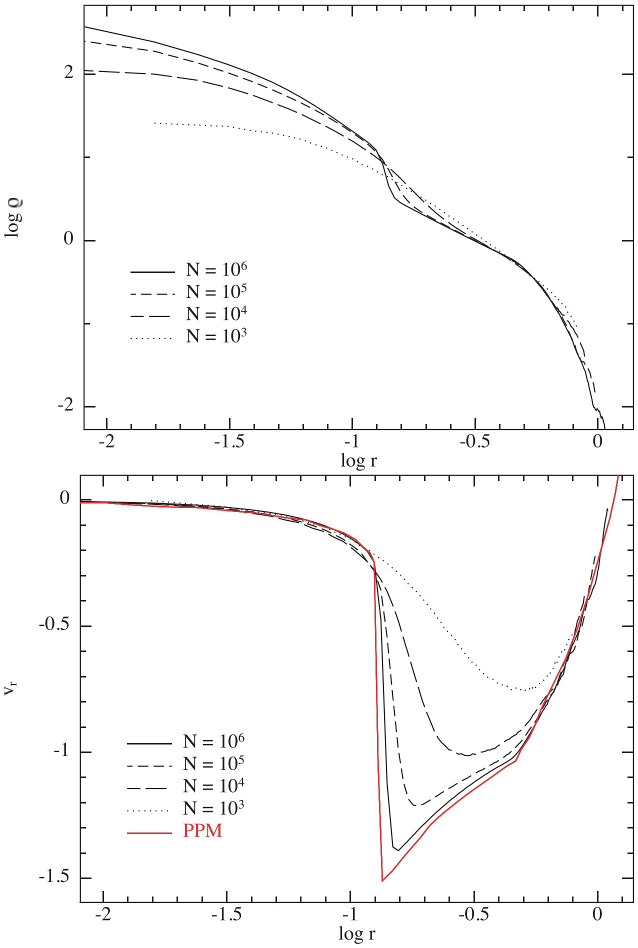

Figure 45. Radial profile of the Evrard collapse (Evrard 1988) at $t = 0.77$, red lines are taken from Figure 7 of Steinmetz & Mueller (1993), with panels showing density (top) and radial velocity (bottom) as a function of (log) radius. All values are given in code units, where $R = M = G = 1$. The outward propagating shock at $r \approx 0.1$ is sharper at high resolution.

line in the figure. At this time, the outward propagating shock is at $r \approx 0.1$, with the shock profile in agreement with the reference solution at high resolution.

5.9. Dust–gas mixtures

The SPH algorithms used in PHANTOM for dust–gas mixtures have been extensively benchmarked in Laibe & Price (2012a, 2012b) (for the two-fluid method) and in Price & Laibe (2015a) (for the one-fluid method; hereafter PL15). Here, we merely demonstrate that the implementation of these algorithms in PHANTOM gives satisfactory results on these tests. For recent applications of PHANTOM to more realistic problems involving dust/gas mixtures, see Dipierro et al. (2015, 2016), Ragusa et al. (2017), and Tricco et al. (2017).

5.9.1. *DUSTYBOX*

Figure 46 shows the results of the DUSTYBOX problem (Monaghan & Kocharyan 1995; Paardekooper & Mellema 2006; Laibe & Price 2011). We setup a uniform, periodic box x, y, $z \in [-0.5, 0.5]$ with 32 × 36 × 39 gas particles set on a close-packed lattice and 32 × 36 × 39 dust particles also set on a close-packed lattice. The gas particles are initially at rest, while the dust is given a uniform velocity $v_x = 1$ in the x-direction. We employ an isothermal EOS with $c_s = 1$, uniform gas and dust densities of $\rho_g = \rho_d = 1$, using the cubic spline kernel for the SPH terms and the double-hump cubic spline kernel (Section 2.13.4) for the drag terms, following Laibe & Price (2012a).

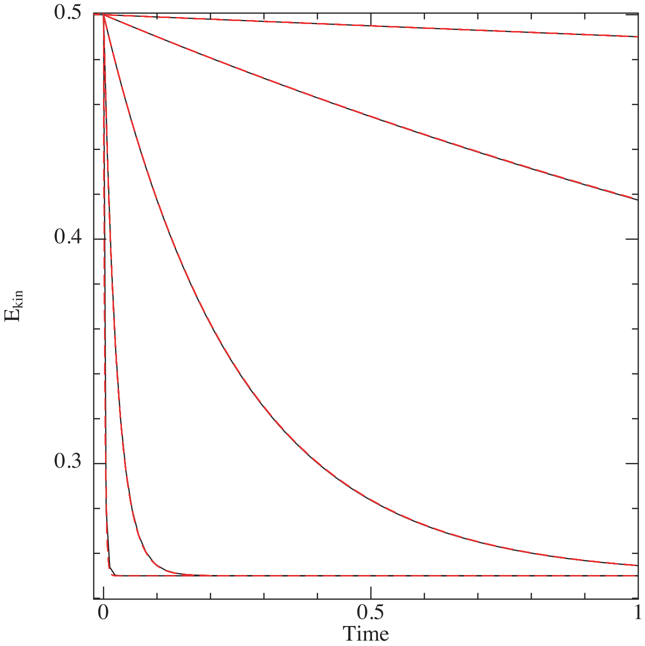

Figure 46. Decay of kinetic energy as a function of time in the DUSTYBOX test, involving a uniformly translating mixture of gas and dust coupled by drag. Solid lines show the PHANTOM results for drag coefficients $K =$ 0.01, 0.1, 1.0, 10, and 100 (top to bottom), which may be compared to the corresponding analytic solutions given by the dashed red lines.

The red dashed lines in Figure 46 show the exact solution for kinetic energy as a function of time. For our chosen parameters, the barycentric velocity is $v_x = 0.5$, giving $v_g(t) = 0.5[1 + \Delta v_x(t)]$, $v_d(t) = 0.5[1 - \Delta v_x(t)]$, where $\Delta v_x(t) = \exp(-2Kt)$ (Laibe & Price 2011) and the red lines show $E_{\rm kin}(t) = \frac{1}{2}[v_g(t)^2 + v_d(t)^2]$. The close match between the numerical and analytic solutions ($\mathcal{L}_2 \sim 3 \times 10^{-4}$) demonstrates that the drag terms are implemented correctly.

The DUSTYBOX test is irrelevant for the one-fluid method (Section 2.13.14) since this method implicitly assumes that the drag is strong enough so that the terminal velocity approximation holds—implying that the relative velocites are simply the barycentric values at the end of the DUSTYBOX test.

5.9.2. *DUSTYWAVE*

Maddison (1998) and Laibe & Price (2011) derived the analytic solution for linear waves in a dust–gas mixture: the

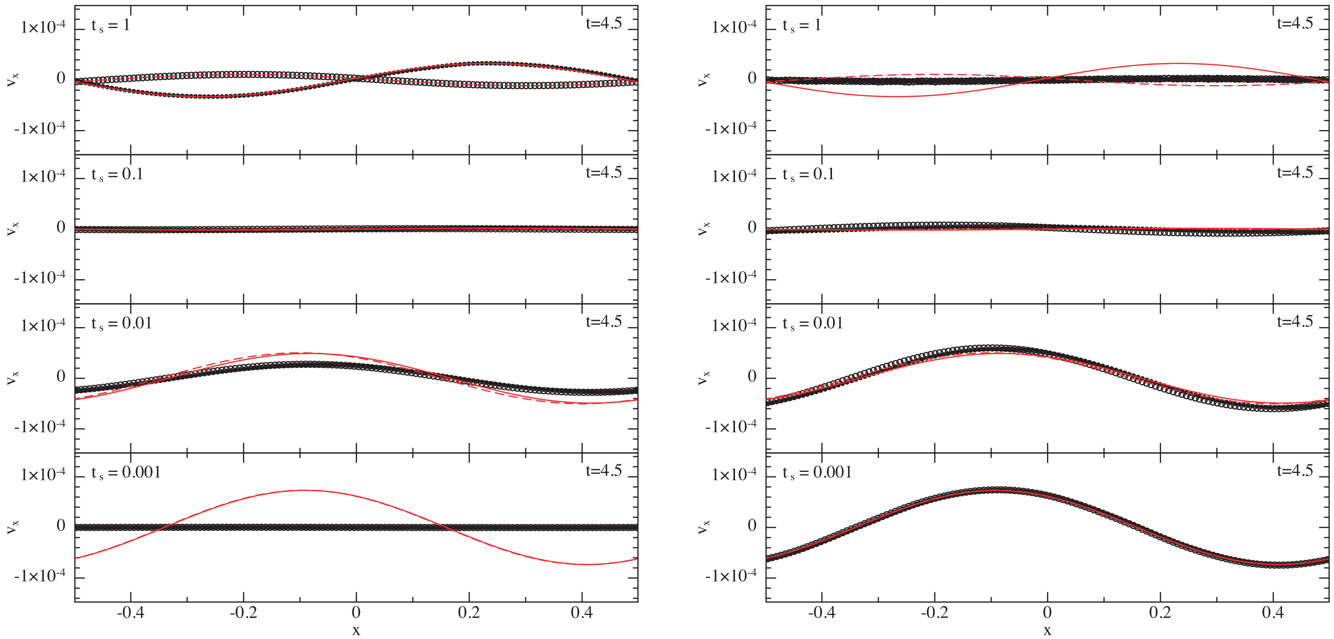

Figure 47. Velocity of gas (solid) and dust (circles) at $t = 4.5$ in the DUSTYWAVE test, using the two-fluid method (left; with 64 × 12 × 12 gas particles, 64 × 12 × 12 dust particles) and the one-fluid method (right; with 64 × 12 × 12 mixture particles) and a dust-to-gas ratio of unity. Panels show results with $K = 0.5$, 5, 50, and 500 (top to bottom), corresponding to the stopping times indicated. The two-fluid method is accurate when the stopping time is long (top three panels in left figure) but requires $h \lesssim c_{\rm s} t_{\rm s}$ to avoid overdamping (bottom two panels on left; Laibe & Price 2012a). The one-fluid method should be used when the stopping time is short (right figure).

'DUSTYWAVE' test. The corresponding dispersion relation is given by (Maddison 1998; Laibe & Price 2011, 2012a)

$$\omega^3 + iK\left(\frac{1}{\rho_{\rm g}} + \frac{1}{\rho_{\rm d}}\right)\omega^2 - c_{\rm s}^2 k^2 \omega - iK\frac{k^2 c_{\rm s}^2}{\rho_{\rm d}} = 0, \qquad (327)$$

which can be more clearly expressed as

$$(\omega^2 - c_{\rm s}^2 k^2) + \frac{i}{\omega t_{\rm s}}(\omega^2 - \tilde{c}_{\rm s}^2 k^2) = 0, \qquad (328)$$

where $\tilde{c}_{\rm s} = c_{\rm s}(1 + \rho_{\rm d}/\rho_{\rm g})^{-1/2}$ is the modified sound speed (e.g. Miura & Glass 1982). This demonstrates the two important limits: (i) $t_{\rm s} \rightarrow \infty$, giving undamped sound waves in the gas and (ii) $t_{\rm s} \rightarrow 0$, giving undamped sound waves in the mixture propagating at the modified sound speed. In between these limits, the mixture is dissipative and waves are damped by the imaginary term. This is seen in the analytic solutions shown in Figure 47.

Two-fluid. We perform this test first with the two-fluid algorithm, using 64 × 12 × 12 gas particles and 64 × 12 × 12 dust particles set up on a uniform, close-packed lattice in a periodic box with $x \in [-0.5, 0.5]$ and the y and z boundaries set to correspond to 12 particle spacings on the chosen lattice. The wave is set to propagate along the x-axis with $v_{\rm g} = v_{\rm d} = A\sin(2\pi x)$, $\rho = \rho_0[1 + A\sin(2\pi x)]$ with $\rho_0 = 1$ and $A = 10^{-4}$. The density perturbation is initialised using stretch mapping (Section 3.2; see also Appendix B in Price & Monaghan 2004b). We perform this test using an adiabatic EOS with $c_{\rm s,0} = 1$. We adopt a simple, constant K drag prescription, choosing $K = 0.5$, 5, 50, and 500 such that the stopping time given by (236) is a multiple of the wave period ($t_{\rm s} = 1$, 0.1, 0.01, and 0.001, respectively).

The left panels in Figure 47 show the results of this test using the two-fluid method, showing velocity in each phase compared to the analytic solution after 4.5 wave periods (the time is chosen to give a phase offset between the phases). For stopping times $t_{\rm s} \gtrsim 0.1$, the numerical solution matches the analytic solution to within 4%. For short stopping times, Laibe & Price (2012a) showed that the resolution criterion $h \lesssim c_{\rm s} t_{\rm s}$ needs to be satisfied to avoid overdamping of the mixture. For the chosen number of particles, the smoothing length is $h = 0.016$, implying in this case that the criterion is violated when $t_{\rm s} \lesssim 0.016$. This is evident from the lower two panels in Figure 47, where the numerical solution is overdamped compared to the analytic solution.

This problem is not unique to SPH codes, but represents a fundamental limitation of two-fluid algorithms in the limit of short stopping times due to the need to resolve the physical separation between the phases (which becomes ever smaller as $t_{\rm s}$ decreases) when they are modelled with separate sets of particles (or with a grid and a physically separate set of dust particles). The need to resolve a physical length scale results in first-order convergence of the algorithm in the limit of short stopping times, as already noticed by Miniati (2010) in the context of grid-based codes. The problem is less severe when the dust fraction is small (Lorén-Aguilar & Bate 2014), but is difficult to ameliorate fully.

One-fluid. The limit of short stopping time (small grains) is the limit in which the mixture is well described by the

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

one-fluid formulation in the terminal velocity approximation (Section 2.13.12). To compare and contrast the two methods for simulating dust in PHANTOM, the right half of Figure 47 shows the results of the DUSTYWAVE test computed with the one-fluid method. To perform this test, we set up a single set of 64 × 12 × 12 'mixture' particles placed on a uniform closepacked lattice, with an initially uniform dust fraction $\varepsilon = 0.5$. The particles are given a mass corresponding to the combined mass of the gas *and* dust, with the density perturbation set as previously.

The one-fluid solution is accurate precisely where the two-fluid method is inaccurate, and vice-versa. For short stopping times ($t_s = 0.001$; bottom row) the numerical solution is within 1.5% of the analytic solution, compared to errors greater than 60% for the two-fluid method (left figure). For long stopping times ($t_s \lesssim 1$; top two rows), the one-fluid method is both inaccurate and slow, but this is precisely the regime in which the two-fluid method (left figure) is explicit and therefore cheap. Thus, the two methods are complementary.

5.9.3. Dust diffusion

A simple test of the one-fluid dust diffusion algorithm is given by PL15. For this test, we set up the particles on a uniform cubic lattice in a 3D periodic box $x, y, z \in [-0.5, 0.5]$ using 32 × 32 × 32 particles with an initial dust fraction set according to

$$\epsilon(r, 0) = \begin{cases} \epsilon_0 \left[1 - \left(\frac{r}{r_c}\right)^2\right], & r < r_c, \\ 0, & \text{elsewhere}, \end{cases} \tag{329}$$

with $\varepsilon_0 = 0.1$ and $r_c = 0.25$. We then evolve the dust diffusion equation, (272), discretised according to (279), while setting the acceleration and thermal energy evolution to zero and assuming $P = \rho$, with the stopping time set to a constant $t_s = 0.1$ and the computational timestep set to $\Delta t = 0.05$. Figure 48 shows the evolution of the dust fraction $\varepsilon \equiv \rho_d/\rho$ as a function of radius at various times, showing the projection of all particles in the box (points) compared to the exact solution (red lines) at $t = 0.0$, 0.1, 0.3, 1.0, and 10.0 (top to bottom). The solution shows a close match to the analytic solution, with agreement to within 0.3% of the analytic solution at all times.

5.9.4. Dust settling

We perform the dust settling test from PL15 in order to directly compare the PHANTOM solutions to those produced in PL15 with the NDSPMHD code. To simplify matters, we do not consider rotation but simply adapt the 2D problem to 3D by using a thin Cartesian box (as for several of the MHD tests in Section 5.6). Our setup follows PL15, considering a slice of a protoplanetary disc at $R_0 = 50$ au in the r–z plane (corresponding to our x and y Cartesian coordinates, respectively)

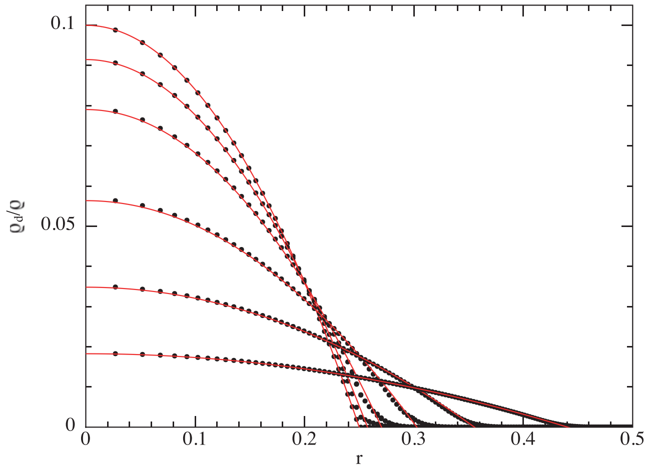

Figure 48. Dust diffusion test from Price & Laibe (2015a), showing the evolution of the dust fraction on the particles (black dots) as a function of radius at six different times (top to bottom), which may be compared to the analytic solution given by the red lines.

with density in the 'vertical' direction (y) given by

$$\rho(y) = \rho_0 \exp\left(\frac{-y^2}{2H^2}\right), \tag{330}$$

where we choose $H/R_0 = 0.05$ giving $H = 2.5$ au. We use an isothermal EOS with sound speed $c_s \equiv H\Omega$, where $\Omega \equiv \sqrt{GM/R_0^3}$, corresponding to an orbital time $t_{\rm orb} \equiv 2\pi/\Omega \approx 353$ yr. We adopt code units with a distance unit of 10 au, mass in solar masses, and time units such that $G = 1$, giving an orbital time of ≈70.2 in code units. We apply an external acceleration in the form

$$\boldsymbol{a}_{\rm ext} = -\frac{GMy}{\sqrt{R_0^2 + y^2}}, \tag{331}$$

where $G = M = 1$ in code units.

The particles are placed initially on a close-packed lattice using 32 × 110 × 24 = 84 480 particles in the domain $[x, y, z] \in [\pm 0.25, \pm 3H, \pm\sqrt{3/128}]$. We then use the stretch mapping routine (Section 3.2) to give the density profile according to (330). We set the mid-plane density to 10^{-3} in code units, or $\approx 6 \times 10^{-13}$g cm^{-3}. The corresponding particle mass in code units is 1.13×10^{-9}. We use periodic boundaries, with the boundary in the y direction set at $\pm 10H$ to avoid periodicity in the vertical direction.

Following the procedure in PL15, we relax the density profile by evolving for 15 orbits with gas only with damping switched on. We then restart the calculation with either (i) a dust fraction added to each particle (one-fluid) or (ii) a corresponding set of dust particles duplicated from the gas particles (two-fluid). For the dust, we assume 1-mm grains with an Epstein drag prescription, such that the stopping time is given by (250). Since Δv is not available when computing t_s with the one-fluid method, we set the factor $f = 1$ in (250) when using this method (this is a valid approximation since by definition Δv is small when the one-fluid method

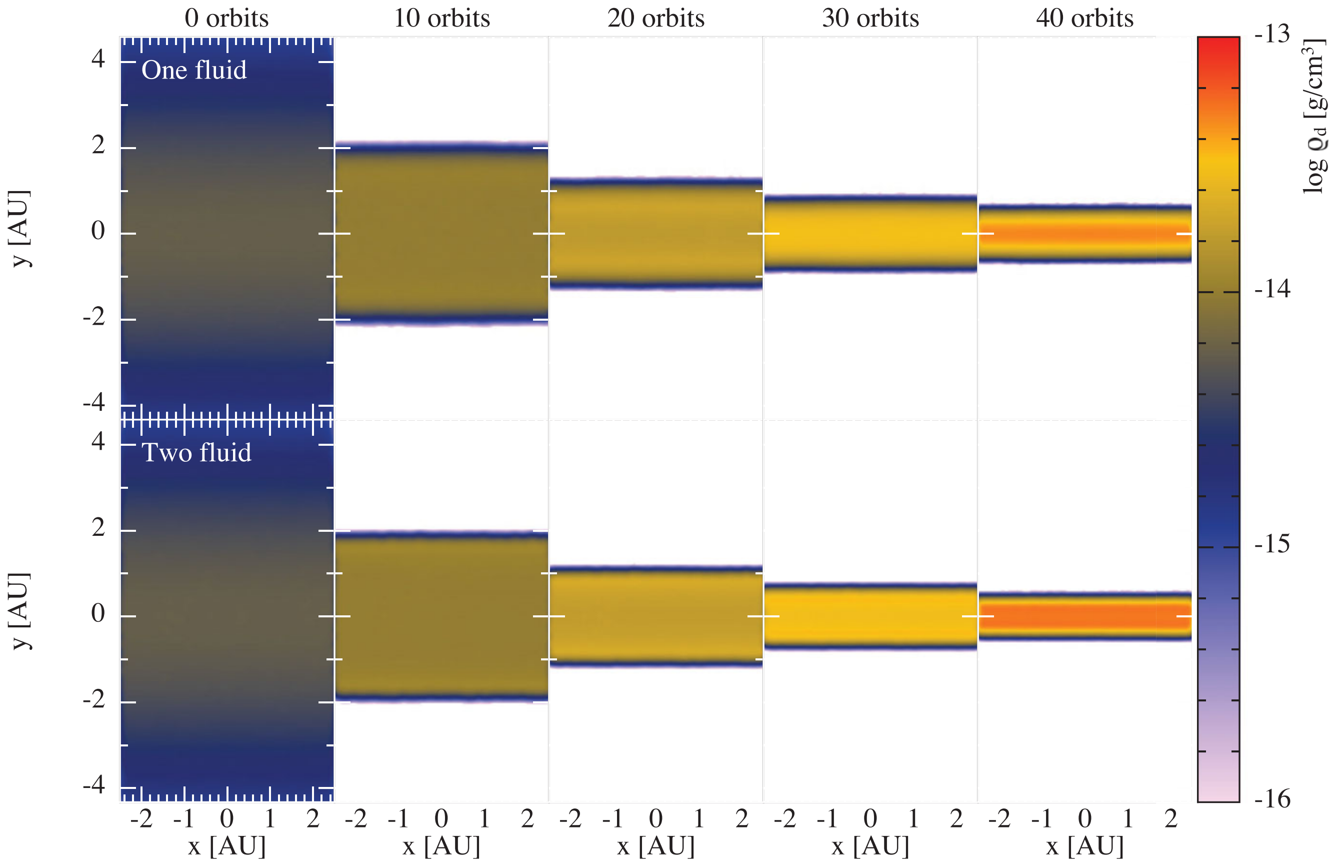

Figure 49. 3D version of the dust settling test from Price & Laibe (2015a), showing the dust density in the 'r–z' plane of a protoplanetary disc. We assume mm-sized grains with a 1% initial dust-to-gas ratio in a stratified disc atmosphere with $H/R_0 = 0.05$ with $R_0 = 50$ au in (331).

is applicable). The dimensionless stopping time $t_s\Omega = 8.46 \times 10^{-3}$ initially at the disc midplane. After adding dust, we continued the simulation for a further 50 orbits.

Figure 49 shows the dust density at intervals of 10 $t_{\rm orb}$, showing the cross-section slice through the $z = 0$ plane of the 3D box which may be directly compared to the 2D solutions shown in PL15. Settling of the dust layer proceeds as expected, with close agreement between the one-fluid (top row) and two-fluid (bottom row) methods, though the two-fluid method is much slower for this test because of the timestep constraint imposed by the stopping time, c.f. (261). The dust resolution is higher in the two-fluid calculation because the set of dust particles follow the dust mass rather than the total mass (for the one-fluid method).

5.10. ISM cooling and chemistry

Figure 50 shows the behaviour of the various cooling and chemistry modules used when modelling the ISM. These plots were made from the data from a simulation of gas rings embedded in a static background potential giving a flat rotation curve (Binney & Tremaine 1987). Gas is setup in a ring of constant surface density from 5 to 10 kpc in radius, initially at 10 000 K, with a total gas mass of 2×10^9 M$_\odot$. The top panels show a temperature and pressure profile of all gas particles. The temperature plateaus around 10 000 K and forms a two-phase medium visible in the 'knee' in the pressure profile, as expected for ISM thermal models (Wolfire et al. 1995). Much lower temperatures can be reached if the gas is given a higher surface density or if self-gravity is active. In the case of the latter, some energy delivery scheme, or the inclusion of a large number of sink particles, is needed to break apart the cold knots.

The bottom two panels of Figure 50 show the chemical abundances of H_2 and CO. The exact form of the molecular abundance profiles are a function of many variables that are set at run time, with the data in the figure made from the default values. The molecular abundances are strong functions of total gas density, with the CO being a strong function of H_2 abundance. If a higher gas mass (e.g. × 10 the value used here) or self-gravity is included then abundances reach a maximum of either 0.5 for H_2 or the primordial carbon abundance for CO. See Dobbs et al. (2008) for a detailed discussion of the features of these abundance curves.

6 EXAMPLE APPLICATIONS

PHANTOM is already a mature code, in the sense that we have always developed the code with specific applications in mind. In this final section, we demonstrate five example

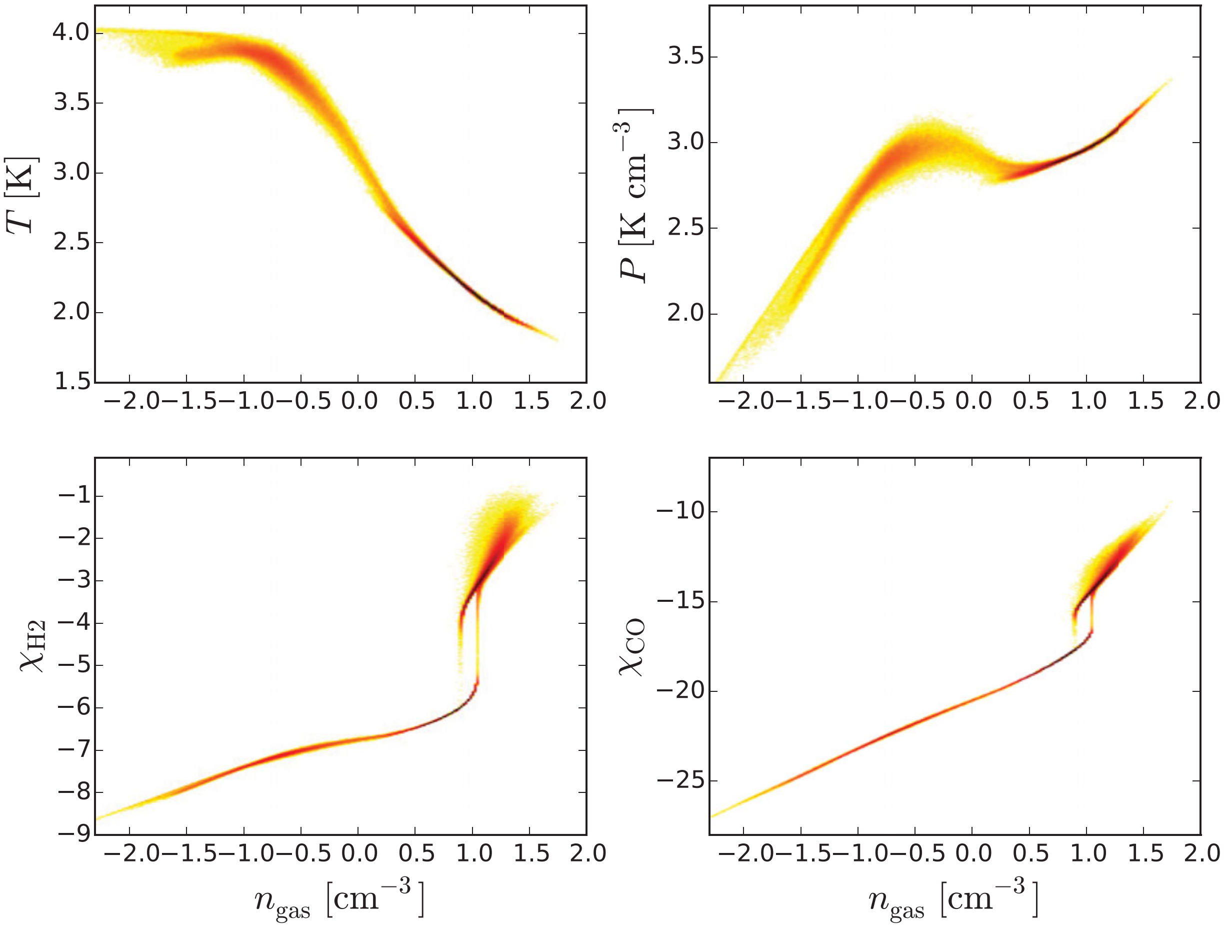

Figure 50. Temperature and pressure profiles resulting from the ISM heating and cooling functions (top) and the abundances of H_2 and CO as a function of gas density (bottom). The behaviour of the H_2 and CO fractions close to $n = 10\,\mathrm{cm}^{-3}$ is a consequence of H_2 self-shielding: In gas which was initially molecular, this remains effective down to lower densities than is the case in gas which was initially atomic. This behaviour is discussed in much greater detail in Dobbs et al. (2008).

applications for which the code is well suited. The setup for each of these applications are provided in the wiki documentation so they can be easily reproduced by the novice user. We also plan to incorporate these examples into an ‘optimisation suite’ to benchmark performance improvements to the code.

6.1. Supersonic turbulence

Our first example application employs the turbulence forcing module described in Section 2.5. Figure 51 shows the gas column density in simulations of isothermal supersonic turbulence driven to an rms Mach number of $\mathcal{M} \approx 10$, identical to those performed by Price & Federrath (2010). The calculations use 256^3 particles and were evolved for 10 turbulent crossing times, $t_c = L/(2\mathcal{M})$. This yields a crossing time of $t_c = 0.05$ in code units. The gas is isothermal with sound speed $c_s = 1$ in code units. The initial density is uniform $\rho_0 = 1$.

The gas column density plots in Figure 51 may be directly compared to the panels in Figure 3 of Price & Federrath (2010). Figure 52 shows the time-averaged probability distribution function (PDF) of $s \equiv \ln(\rho/\rho_0)$. This demonstrates the characteristic signature of isothermal supersonic turbulence, namely the appearance of a log-normal PDF in s (e.g. Vazquez-Semadeni 1994; Nordlund & Padoan 1999; Ostriker, Gammie, & Stone 1999). Indeed, PHANTOM was used in the study by Price, Federrath, & Brunt (2011) to confirm the relationship between the standard deviation and the Mach number in the form

$$\sigma_s^2 = \ln\left(1 + b^2\mathcal{M}^2\right), \qquad (332)$$

where $b = 1/3$ for solenoidally driven turbulence, as earlier suggested by Federrath et al. (2008, 2010a).

6.2. Star cluster formation

SPH has been used to study star formation since the earliest studies by Gingold & Monaghan (1982a, 1983), Phillips (1982, 1986a, 1986b), and Monaghan & Lattanzio (1986, 1991), even motivating the original development of MHD

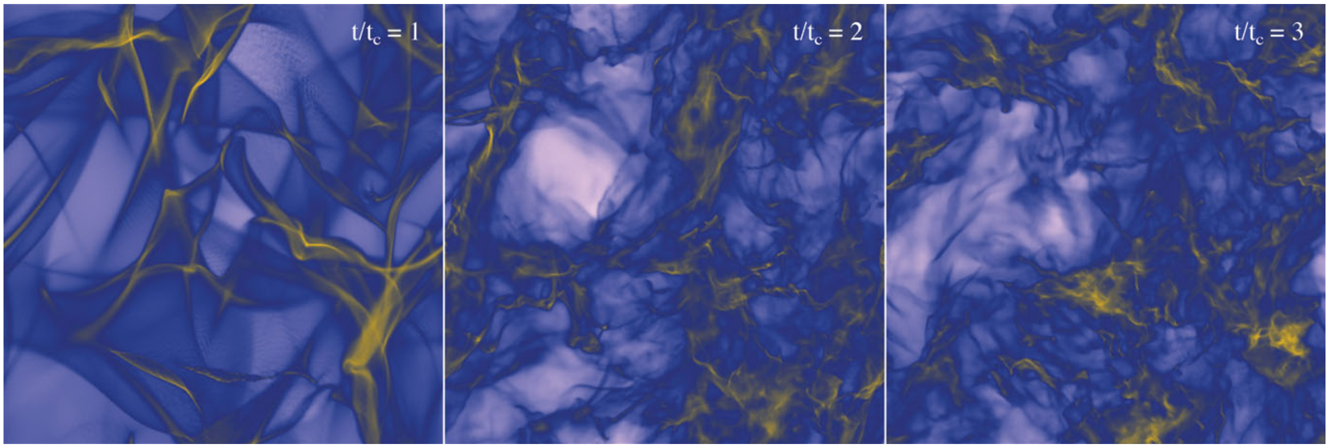

Figure 51. Gas column density in PHANTOM simulations of driven, isothermal, supersonic turbulence at Mach 10, similar to the calculations performed by Price & Federrath (2010). We show the numerical solutions at $t = 1$, 2, and 3 crossing times (left to right, respectively). The colour scale is logarithmic between 10^{-1} and 10 in code units.

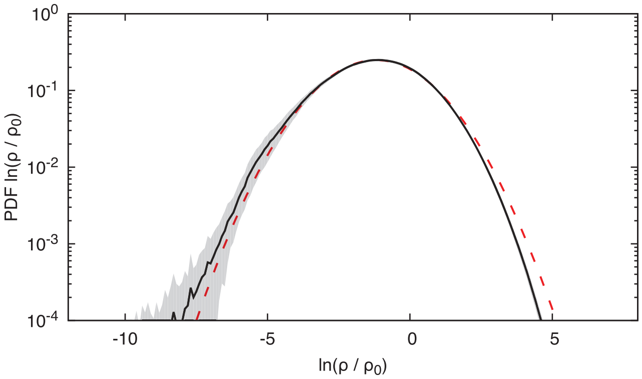

Figure 52. Time-averaged PDF of $s = \ln(\rho/\rho_0)$ for supersonic Mach 10 turbulence, with the shaded region representing the standard deviation of the averaging. The PDF is close to a log-normal distribution, shown by the dashed red line.

in SPH by Phillips & Monaghan (1985). The study by Bate et al. (2003) represented the first simulation of 'large-scale' star cluster formation, resolved to the opacity limit for fragmentation (Rees 1976; Low & Lynden-Bell 1976). This was enabled by the earlier development of sink particles by Bate et al. (1995), allowing star formation simulations to continue beyond the initial collapse (Bonnell et al. 1997; Bate & Bonnell 1997). This heritage is present in PHANTOM which inherits many of the ideas and algorithms implemented in the SPHNG code.

Figure 53 shows a series of snapshots taken from a recent application of PHANTOM to star cluster formation by Liptai et al. (2017). The initial setup follows Bate et al. (2003)—a uniform density sphere of 0.375 pc in diameter with a mass of 50 M$_\odot$. The initial velocity field is purely solenoidal, generated on a 64^3 uniform grid in Fourier space to give a power spectrum $P(k) \propto k^{-4}$ consistent with the Larson (1981) scaling relations, and then linearly interpolated from the grid to the particles. The initial kinetic energy is set to match the gravitational potential energy, ($3/5GM^2/R$), giving a root mean square Mach number ≈6.4. We set up 3.5×10^6 particles in the initial sphere placed in a uniform random distribution. We evolve the simulation using a barotropic EOS $P = K\rho^\gamma$ in the form

$$\frac{P}{\rho} = \begin{cases} c_{\rm s,0}^2, & \rho < \rho_1, \\ c_{\rm s,0}^2 \left(\dfrac{\rho}{\rho_1}\right)^{(\gamma_1-1)}, & \rho_1 \le \rho < \rho_2, \\ c_{\rm s,0}^2 \left(\dfrac{\rho_2}{\rho_1}\right)^{(\gamma_1-1)} \left(\dfrac{\rho}{\rho_1}\right)^{(\gamma_2-1)}, & \rho_2 \le \rho < \rho_3, \\ c_{\rm s,0}^2 \left(\dfrac{\rho_2}{\rho_1}\right)^{(\gamma_1-1)} \left(\dfrac{\rho_3}{\rho_2}\right)^{(\gamma_2-1)} \left(\dfrac{\rho}{\rho_3}\right)^{(\gamma_3-1)}, & \rho \ge \rho_3, \end{cases} \tag{333}$$

where we set the initial sound speed $c_{\rm s,\,0} = 2 \times 10^4$ cm s^{-1} and set $[\rho_1, \rho_2, \rho_3] = [10^{-13}, 10^{-10}, 10^{-3}]$ g cm^{-3} and $[\gamma_1, \gamma_2, \gamma_3] = [1.4, 1.1, 5/3]$, as in Bate et al. (2003). We turn on automatic sink particle creation with a threshold density of 10^{-10} g cm^{-3}, with sink particle accretion radii set to 5 au and particles accreted without checks at 4 au. No sink particles are allowed to be created within 10 au of another existing sink. The calculations satisfy the Bate & Burkert (1997) criterion of resolving the minimum Jeans mass in the calculation (known as the opacity limit for fragmentation; Rees 1976; Low & Lynden-Bell 1976) by at least the number of particles contained within one smoothing sphere.

The snapshots shown in Figure 53 show a similar evolution to the original calculation of Bate et al. (2003). The evolution is not identical since we used a different realisation of the initial turbulent velocity field. A more quantitative comparison can be found in Liptai et al. (2017) where we performed seven different realisations of the collapse in order to measure a statistically meaningful initial mass function (IMF) from the calculations, finding an IMF in agreement with the one found by Bate (2009a) in a much larger (500 M$_\odot$) calculation. The IMF produced with a barotropic EOS does not match the observed local IMF in the Milky Way (e.g. Chabrier 2005), tending to over-produce low mass stars and brown dwarfs.

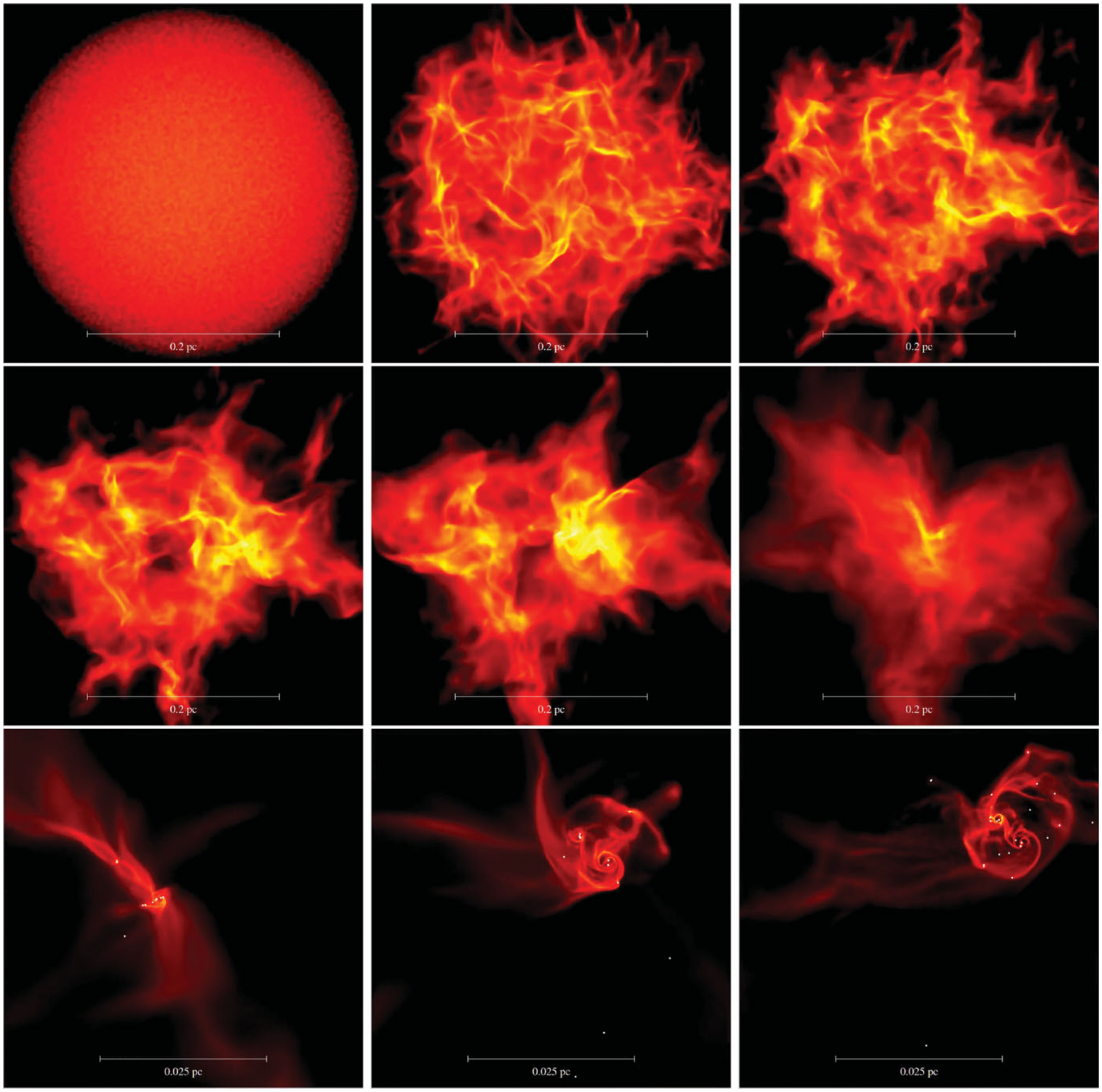

Figure 53. Star cluster formation with PHANTOM, showing snapshots of gas column density during the gravitational collapse of a 50 $M_\odot$ molecular cloud core, following Bate, Bonnell, & Bromm (2003). Snapshots are shown every 0.2 t_{ff} (left to right, top to bottom), with the panels after $t > t_{ff}$ zoomed in to show the details of the star formation sequence. As in Bate et al. (2003), we resolve the fragmentation to the opacity limit using a barotropic equation of state.

This is a known artefact of the barotropic EOS (e.g. Matzner & Levin 2005; Krumholz 2006; Bate 2009b), since material around the stars remains cold rather than being heated. It can be fixed by implementing radiative feedback, for example, by implementing radiation hydrodynamics in the flux-limited diffusion approximation (Whitehouse & Bate 2004; Whitehouse, Bate, & Monaghan 2005; Whitehouse & Bate 2006). This is not yet implemented in PHANTOM but it is high on the agenda.

6.3. Magnetic outflows during star formation

PHANTOM may also be used to model the formation of individual protostars. We present an example following the initial setup and evolution of Price et al. (2012). A molecular cloud core with initial density $\rho_0 = 7.4 \times 10^{-18}$ g cm^{-3} is embedded in pressure equilibrium with ambient medium of density 2.5×10^{-19} g cm^{-3}. The barotropic EOS given by (333) is used, setting $c_s = 2.2 \times 10^4$ cm s^{-1}. The radius of the core

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

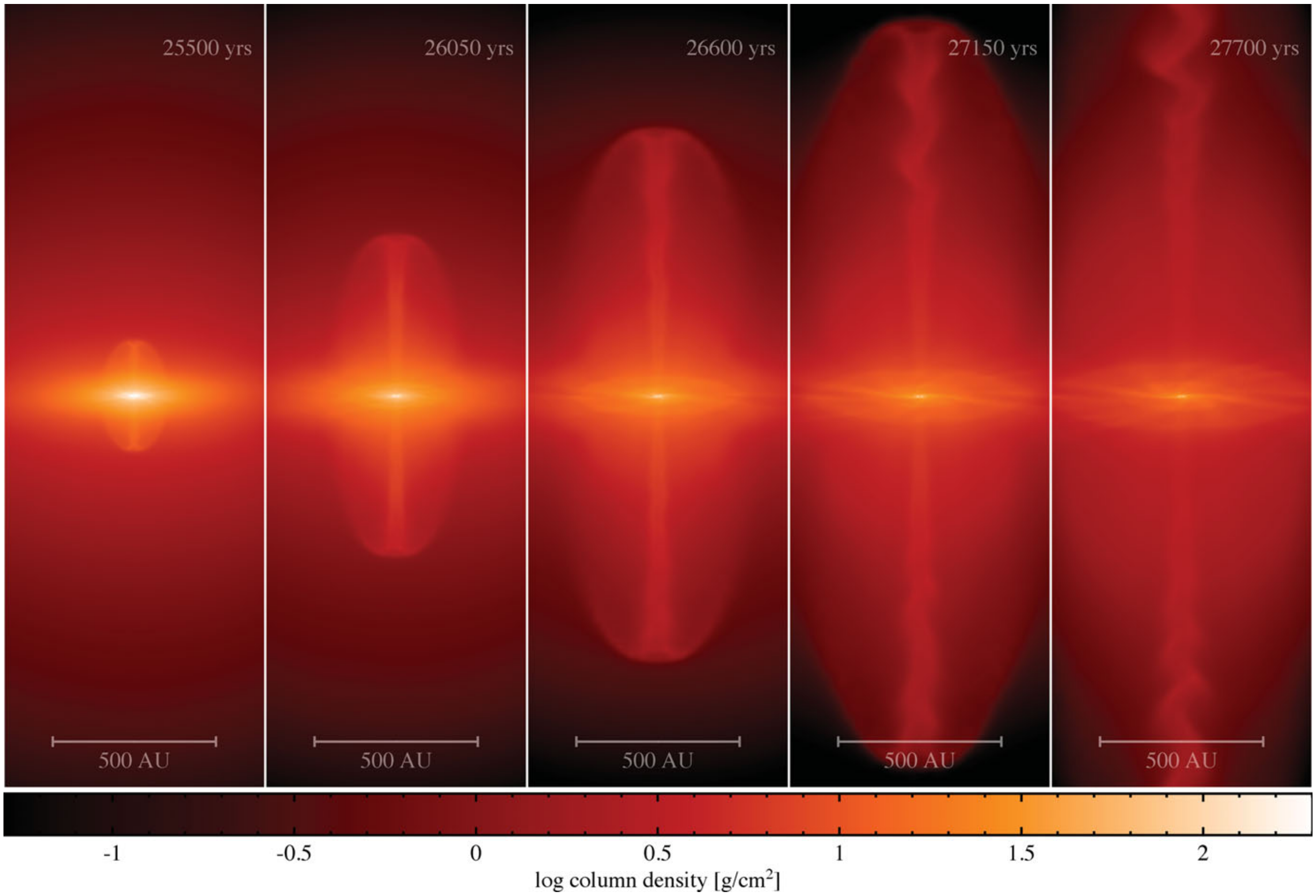

Figure 54. Magnetically propelled jet of material bursting out of the first hydrostatic core phase of star formation.

is 4×10^{16} cm ($\approx$2700 au), with the length of the cubic domain spanning $[x, y, z] = \pm 8 \times 10^{16}$ cm. The core is in solid body rotation with angular speed $\Omega = 1.77 \times 10^{-13}$ rad s^{-1}. The magnetic field is uniform and aligned with the rotation axis with a mass-to-flux ratio $\mu = 5$, corresponding to $B_0 \approx$ 163 μG. A sink particle is inserted once the gas reaches a density of 10^{-10} g cm^{-3}, with an accretion radius of 5 au. Thus, this calculation models only the evolution of the first hydrostatic core phase of star formation. The core is composed of 1 004 255 particles, with 480 033 particles in the surrounding medium.

Figure 54 shows the evolution of a magnetised, collimated bipolar jet of material, similar to that given by Price et al. (2012). Infalling material is ejected due to the wind up of the toroidal magnetic field. The sink particle is inserted at $t \approx 25$ 400 yrs, shortly before the jet begins. The jet continues to be driven, while material continues to infall, lasting for several thousand years.

6.4. Galaxy merger

To provide a realistic test of the collisionless N-body and SPH implementations, we performed a comparison study where we modelled a galaxy merger, comparing the PHANTOM results with the HYDRA N-body/SPH code (Couchman, Thomas, & Pearce 1995; Thacker & Couchman 2006). This test requires gravity along with multiple particle types—gas, stars, and dark matter. Gas interacts hydrodynamically only with itself, and all three particle types interact with each other via gravity (c.f. Table A1).

To create a Milky Way-like galaxy, we used GALACTICS (Kuijken & Dubinski 1995b; Widrow & Dubinski 2005; Widrow, Pym, & Dubinski 2008) to first create a galaxy consisting of a stellar bulge, stellar disc, and a dark matter halo. To create the gas disc, the stellar disc was then duplicated and reflected in the $x = y$ plane to avoid coincidence with the star particles. Ten percent of the total stellar mass was then removed and given to the gas disc. Although the gas disc initially has a scale height larger than physically motivated, this will quickly relax into a disc that physically resembles the Milky Way. Next, we added a hot gas halo embedded within the dark matter halo. The hot gas halo has an observationally motivated β-profile (e.g. Cavaliere & Fusco-Femiano 1976) and a temperature profile given by Kaufmann et al. (2007); the mass of the hot gas halo is removed from the dark matter particles to conserve total halo mass. The mass of each component, as well as particle numbers and particle masses are given in Table 5. To model the major merger, the

Table 5. Component breakdown for each galaxy. For each component, the total mass is M, the particle mass is m, and the number of particles is N.

	$M/\mathrm{M}_\odot$ (10^{10})	$m/\mathrm{M}_\odot$ (10^{5})	N
Dark matter halo	89.92	89.92	100 000
Hot gas halo	0.60	2.77	21 619
Stellar bulge	1.34	18.10	7 407
Stellar disc	3.56	18.10	19 662
Gas disc	0.54	2.77	19 662

galaxy is duplicated and the two galaxies are placed 70 kpc apart on a parabolic trajectory. These initial conditions are identical to those used in Wurster & Thacker (2013b, 2013a). To simplify the comparison, there is no star formation recipe, no black holes, and no feedback from active galactic nuclei (there are currently no plans to implement cosmological recipes in PHANTOM). Thus, only the SPH and gravity algorithms are being compared.

Figure 55 shows the gas column density evolution from $t = 100$ Myr to $t = 1.4$ Gyr, comparing PHANTOM (left) to HYDRA (right), and Figure 56 shows the evolution of the separation of the galaxies and the mean and maximum gas density in each model. The evolution of the two galaxy mergers agree qualitatively with one another, with slight differences in the trajectories, evolution times, and gas densities between the two codes. Using the centre of mass of the star particles that were assigned to each galaxy as a proxy for the galaxy's centre, the maximum separation at $t \approx 450$ Myr is 59 and 61 kpc for HYDRA and PHANTOM, respectively. Second, periapsis occurs at 875 and 905 Myr for HYDRA and PHANTOM, respectively, which is a difference of 3.4% since the beginning of the simulation. The maximum gas density is approximately two times higher in PHANTOM prior to the merger, and about 1.2 times higher after the merger; the average gas densities typically differ by less than a factor of 1.2 both before and after the merger.

There are several differences in the algorithms used in HYDRA compared to PHANTOM. The first is the gravity solver. The long-range gravity in HYDRA uses an adaptive particle-mesh algorithm (Couchman 1991), while PHANTOM uses a kd-tree (c.f. Section 2.12.4). For the short-range gravity, HYDRA uses a fixed S2 softening length for all particles, where the S2 softening is scaled to an equivalent Plummer softening such that $\epsilon_{\mathrm{S2}} = 2.34\epsilon_{\mathrm{Plummer}}$; for this simulation, $\epsilon_i \equiv \epsilon_{\mathrm{Plummer}} = 300$ pc. In PHANTOM, $\epsilon_i = h_i$ for each particle, where h_i is calculated using only the particles of the same type as particle i.

A second difference is the treatment of the smoothing length in high density regions. In HYDRA, as is common in most galactic and cosmological codes, the smoothing length is limited such that $h_i = \max(h_i, h_{\min})$, where $h_{\min} = \epsilon_{\mathrm{Plummer}}/8\,(= 37.5$ pc). In PHANTOM, h_i is always calculated self-consistently and thus has no imposed lower limit.

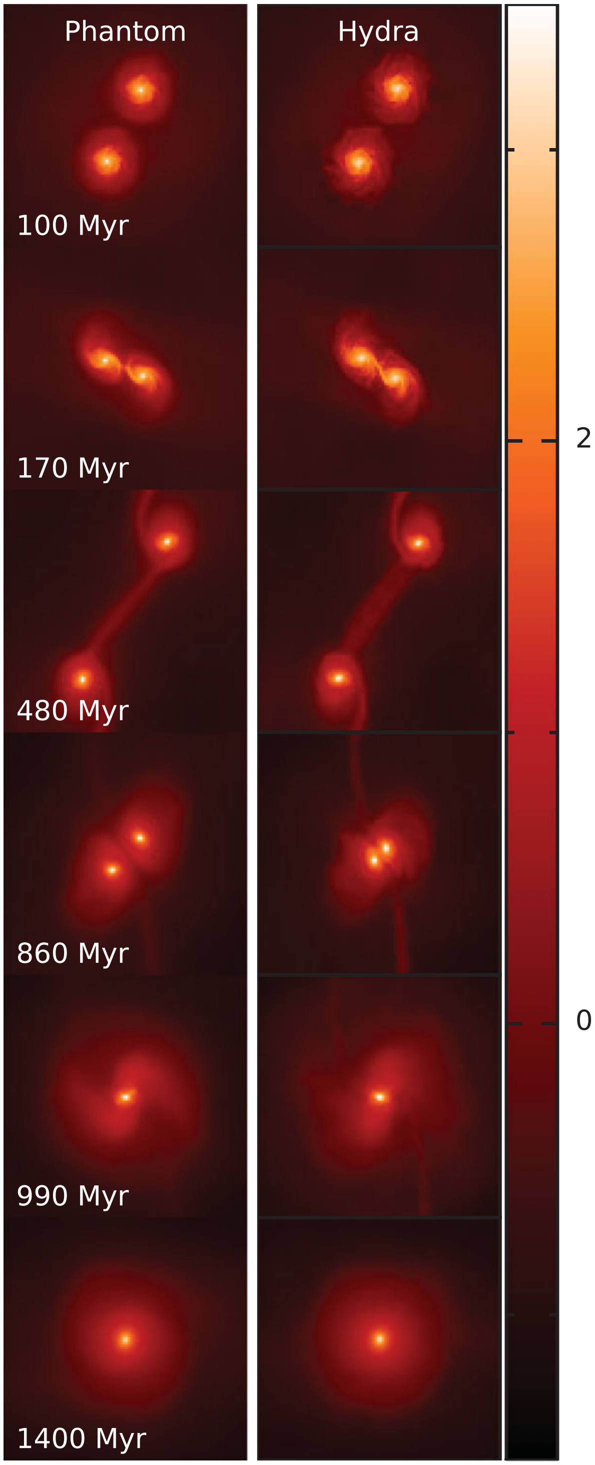

Figure 55. Evolution of the gas column density in a major merger of two Milky Way-sized galaxies, comparing PHANTOM to the HYDRA code. Times shown are from the onset of the simulation, with each frame $(100\ \mathrm{kpc})^2$. The colour bar is log [column density/($\mathrm{M}_\odot\ \mathrm{pc}^{-1}$)].

Finally, PHANTOM contains an artificial conductivity term (Section 2.2.8) that acts to ensure continuous pressure fields across contact discontinuities (Price 2008).

In Wurster & Thacker (2013b), the HYDRA major merger model was compared to a simulation run using the publicly

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

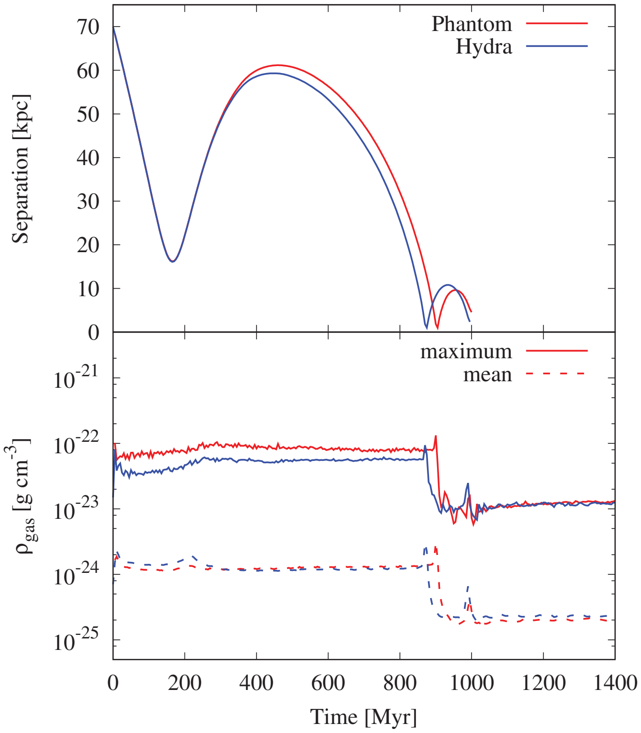

Figure 56. *Top*: The evolution of the separation of the galaxies, using centre of mass of the star particles that were assigned to each galaxy as a proxy for the galaxy's centre; the stars are sufficiently mixed after 1000 Myr, thus a meaningful separation cannot be calculated. *Bottom*: The evolution of the maximum (solid) and mean (dashed) gas densities for each model.

available version of GADGET2 (Springel et al. 2001; Springel & Hernquist 2002). As here, the comparison was simplified such that only the gravity and SPH solvers were being compared. They found that the galaxies in each simulation followed similar trajectories and both models reached second periapsis within 0.2% of one another, as measured from the beginning of the simulation. Note that both HYDRA and GADGET2 were both written primarily to solve galactic and cosmological models.

The quantitive difference in results may be attributed to the improved SPH algorithms in PHANTOM compared to HYDRA. The higher density in HYDRA is consistent with the results in Richardson et al. (2016), who found higher densities in HYDRA compared to the adaptive mesh refinement code RAMSES (Teyssier 2002). It was determined that this was a result of a combination of the artificial viscosity, $h_{\rm min}$ and the suppression of 'mixing' (which occurs when no thermal conductivity is applied).

6.5. Gap opening in dusty discs

Our final example is taken from Dipierro et al. (2016) and builds on our recent studies of dust dynamics in protoplanetary discs with PHANTOM (e.g. Dipierro et al. 2015; Ragusa et al. 2017). We perform calculations using the two-fluid approach, setting up a disc with 500 000 gas particles and 100 000 dust particles with $\Sigma \propto r^{-0.1}$ between 1 and 120 au with a total disc mass of 2×10^{-4} M$_\odot$. The disc mass is chosen to place the mm dust particles in a regime where the Stokes number is greater than unity. The initial dust-to-gas ratio is 0.01 and we assume a locally isothermal EOS with $c_{\rm s} \propto r^{-0.35}$, normalised such that $H/R = 0.05$ at 1 au. We use the minimum disc viscosity possible, setting $\alpha_{\rm AV} = 0.1$.

Figure 57 shows the results of two calculations employing planets of mass 0.1 $M_{\rm Jupiter}$ (top row) and 1.0 $M_{\rm Jupiter}$ (bottom) embedded in a disc around a 1.3 M$_\odot$ star. Left and right panels show gas and dust surface densities, respectively. While the theory of gap opening in gaseous discs is relatively well understood as a competition between the gravitational torque from the planet trying to open a gap and the viscous torques trying to close it (e.g. Goldreich & Tremaine 1979, 1980), gap opening in dusty discs is less well understood (see e.g. Paardekooper & Mellema 2004, 2006). In Dipierro et al. (2016), we identified two regimes for gap opening in dusty discs where gap opening in the dust disc is either *resisted* or *assisted* by the gas–dust drag. The top row of Figure 57 demonstrates that low mass planets can carve a gap which is visible only in the *dust* disc, while for high mass planets (bottom row), there is a gap opened in both gas and dust but it is deeper in the dust. Moreover, the gap opening mechanism by low mass planets has been further investigated in Dipierro & Laibe (2017). They derived a grain size-dependent criterion for dust gap opening in discs, an estimate of the location of the outer edge of the dust gap and an estimate of the minimum Stokes number above which low-mass planets are able to carve gaps which appear only in the dust disc. These predictions has been tested against PHANTOM simulations of planet–disc interaction in a broad range of dusty protoplanetary discs, finding a remarkable agreement between the theoretical model and the numerical experiments.

Interestingly, our prediction of dust gaps that are not coincident with gas gaps for low mass planets appears to be observed in recent observations of the TW Hya protoplanetary disc, by comparing VLT-SPHERE imaging of the scattered light emission from small dust grains (van Boekel et al. 2017; tracing the gas) to ALMA images of the mm dust emission (Andrews et al. 2016).

7 SUMMARY

We have outlined the algorithms and physics currently implemented in the PHANTOM SPH and MHD code in the hope that this will prove useful to both users and developers of the code. We have also demonstrated the performance of the code as it currently stands on a series of standard test problems, most with known or analytic solutions. While no code is ever 'finished' nor bug free, it is our hope that the code as it stands will prove useful to the scientific community. Works in progress for future code releases include radiation

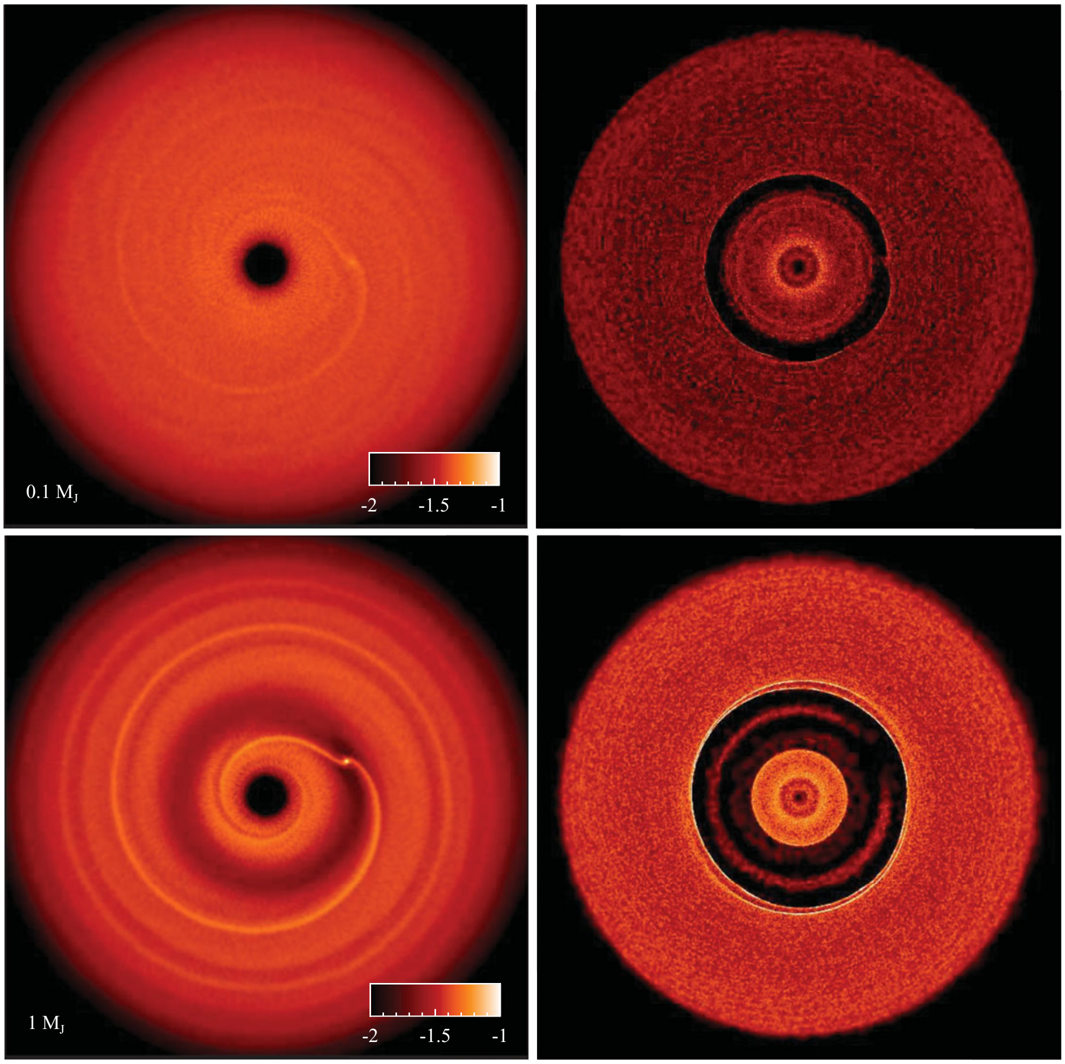

Figure 57. Gap opening in dusty protoplanetary discs with PHANTOM (from Dipierro et al. 2016), showing surface density in gas (left) and mm dust grains (right) in two simulations of planet–disc interaction with planet masses of 0.1 $M_{\rm Jupiter}$ (top) and 1 $M_{\rm Jupiter}$ (bottom) in orbit around a 1.3 $M_{\odot}$ star. In the top case, a gap is opened only in the dust disc, while in the bottom row, the gap is opened in both gas and dust. The colour bar is logarithmic surface density in cgs units.

hydrodynamics, continuing development of the dust algorithms, and an implementation of relativistic hydrodynamics.

ACKNOWLEDGEMENTS

PHANTOM is the result of interactions over the years with numerous talented and interesting people. Particular mentions go to Joe Monaghan, Matthew Bate, and Stephan Rosswog from whom I (DP) learnt and discussed a great deal about SPH over the years. We also thank Walter Dehnen, James Wadsley, Evghenii Gaburov, Matthieu Viallet, and Pedro Gonnet for stimulating interactions. This work and the public release of PHANTOM was made possible by the award of a four-year Future Fellowship (FT130100034) to DJP from the Australian Research Council (ARC), as well as funding via Discovery Projects DP130102078 (funding James Wurster and Mark Hutchison) and DP1094585 (which funded Guillaume Laibe and partially funded Terrence Tricco). CN is supported by the Science and Technology Facilities Council (grant number ST/M005917/1). CF gratefully acknowledges funding provided by ARC Discovery Projects (grants DP150104329 and DP170100603). SCOG acknowledges support from the Deutsche Forschungsgemeinschaft via SFB 881 (sub-projects B1, B2, B8) and from the European Research Council via the ERC Advanced Grant 'STARLIGHT' (project number 339177). This work was supported by resources awarded under Astronomy Australia Ltd's merit allocation scheme on the gSTAR and swinSTAR national facilities at Swinburne University of Technology and the Pawsey National Supercomputing Centre. gSTAR and swinSTAR are funded by Swinburne and the Australian Government's Education Investment Fund. We thank Charlene Yang from the Pawsey Supercomputing Centre for particular help and assistance as part of a Pawsey uptake project. We used SPLASH for many of the figures (Price 2007). We thank Max Tegmark for his excellent icosahedron module used in various analysis routines. We thank the three referees of this paper for extensive comments on the manuscript.

References

Abel, T. 2011, MNRAS, 413, 271

Abel, T., Anninos, P., Zhang, Y., & Norman, M. L. 1997, NewA, 2, 181

Agertz, O., et al. 2007, MNRAS, 380, 963

Alexiades, V., Amiez, G., & Gremaud, P.-A. 1996, CNME, 12, 31

Allen, C., & Santillan, A. 1991, RMxAA, 22, 255

Andrews, S. M., et al. 2016, ApJ, 820, L40

Artymowicz, P., & Lubow, S. H. 1994, ApJ, 421, 651

Artymowicz, P., & Lubow, S. H. 1996, ApJ, 467, L77

Athanassoula, E., Fady, E., Lambert, J. C., & Bosma, A. 2000, MNRAS, 314, 475

Bagla, J. S., & Khandai, N. 2009, MNRAS, 396, 2211

Ballabio, G., Dipierro, G., Veronesi, B., Lodato, G., Hutchison, M., Laibe, G., & Price, D. J. 2018, MNRAS, 477, 2766

Balsara, D. S. 1995, JCoPh, 121, 357

Balsara, D. S. 1996, ApJ, 465, 775

Balsara, D. S. 1998, ApJS, 116, 133

Balsara, D. S., & Kim, J. 2004, ApJ, 602, 1079

Balsara, D. S., & Spicer, D. 1999, JCoPh, 148, 133

Balsara, D. S., Kim, J., Mac Low, M.-M., & Mathews, G. J. 2004, ApJ, 617, 339

Barmin, A., Kulikovskiy, A. G., & Pogorelov, N. V. 1996, JCoPh, 126, 77

Barnes, D. J., Kawata, D., & Wu, K. 2012, MNRAS, 420, 3195

Bartosch, L. 2001, IJMPC, 12, 851

Bate, M. R. 1995, PhD thesis, University of Cambridge, Cambridge, UK

Bate, M. R. 2009a, MNRAS, 392, 590

Bate, M. R. 2009b, MNRAS, 392, 1363

Bate, M. R., & Bonnell, I. A. 1997, MNRAS, 285, 33

Bate, M. R., & Burkert, A. 1997, MNRAS, 288, 1060

Bate, M. R., Bonnell, I. A., & Price, N. M. 1995, MNRAS, 277, 362

Bate, M. R., Bonnell, I. A., & Bromm, V. 2003, MNRAS, 339, 577

Bauer, A., & Springel, V. 2012, MNRAS, 423, 2558

Beck, A. M., et al. 2016, MNRAS, 455, 2110

Begeman, K. G., Broeils, A. H., & Sanders, R. H. 1991, MNRAS, 249, 523

Benz, W. 1990, in Proceedings of the NATO Advanced Research Workshop on The Numerical Modelling of Nonlinear Stellar Pulsations Problems and Prospects, Held in Les Arcs, France, March 20–24, 1986, ed. J. Robert Buchler (Dordrecht and Boston: Kluwer Academic Publishers)

Benz, W., Cameron, A. G. W., Press, W. H., & Bowers, R. L. 1990, ApJ, 348, 647

Bergin, E. A., Hartmann, L. W., Raymond, J. C., & Ballesteros-Paredes J,. 2004, ApJ, 612, 921

Biermann, L. 1950, ZNatA, 5, 65

Binney, J., & Tremaine, S. 1987, Galactic Dynamics (Princeton, NJ: Princeton University Press)

Bocquet, S., Saro, A., Dolag, K., & Mohr, J. J. 2016, MNRAS, 456, 2361

Boily, C. M., Kroupa, P., & Peñarrubia-Garrido, J. 2001, NewA, 6, 27

Bonafede, A., Dolag, K., Stasyszyn, F., Murante, G., & Borgani, S. 2011, MNRAS, 418, 2234

Bonet, J., & Lok, T-S. L. 1999, CMAME, 180, 97

Bonnell, I. A., Bate, M. R., Clarke, C. J., & Pringle, J. E. 1997, MNRAS, 285, 201

Bonnerot, C., Rossi, E. M., Lodato, G., & Price, D. J. 2016, MNRAS, 455, 2253

Bonnerot, C., Price, D. J., Lodato, G., & Rossi, E. M. 2017, MNRAS, 469, 4879

Børve, S., Omang, M., & Trulsen, J. 2001, ApJ, 561, 82

Børve, S., Omang, M., & Trulsen, J. 2004, ApJS, 153, 447

Brio, M., & Wu, C. C. 1988, JCoPh, 75, 400

Brookshaw, L. 1985, PASA, 6, 207

Brookshaw, L. 1994, MmSAI, 65, 1033

Bürzle, F., Clark, P. C., Stasyszyn, F., Greif, T., Dolag, K., Klessen, R. S., & Nielaba, P. 2011a, MNRAS, 412, 171

Bürzle, F., Clark, P. C., Stasyszyn, F., Dolag, K., & Klessen R. S,. 2011b, MNRAS, 417, L61

Cabezón, R. M., García-Senz, D., & Relaño, A. 2008, JCoPh, 227, 8523

Caldwell, J. A. R., & Ostriker, J. P. 1981, ApJ, 251, 61

Cartwright, A., Stamatellos, D., & Whitworth, A. P. 2009, MNRAS, 395, 2373

Cavaliere, A., & Fusco-Femiano, R. 1976, A&A, 49, 137

Cerioli, A., Lodato, G., & Price, D. J. 2016, MNRAS, 457, 939

Cha, S.-H., & Whitworth, A. P. 2003, MNRAS, 340, 73

Chabrier, G. 2005, in Astrophysics and Space Science Library, Vol. 327, Proceedings of The Initial Mass Function 50 Years Later, ed. E. Corbelli, F. Palle, & H. Zinnecker (Dordrecht: Springer), 41

Chapman, S., & Cowling, T. G. 1970, The Mathematical Theory of Non-Uniform Gases. An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases (Cambridge: Cambridge University Press)

Chin, S. A., & Chen, C. R. 2005, CeMDA, 91, 301

Choi, E., Kim, J., & Wiita, P. J. 2009, ApJS, 181, 413

Chow, E., & Monaghan, J. J. 1997, JCoPh, 134, 296

Cossins, P., Lodato, G., & Clarke, C. J. 2009, MNRAS, 393, 1157

Couchman, H. M. P. 1991, ApJ, 368, L23

Couchman, H. M. P., Thomas, P. A., & Pearce, F. R. 1995, ApJ, 452, 797

Coughlin, E. R., & Nixon, C. 2015, ApJ, 808, L11

Coughlin, E. R., Armitage, P. J., Nixon, C., & Begelman, M. C. 2017, MNRAS 465, 3840

Coughlin, E. R., Nixon, C., Begelman, M. C., & Armitage, P. J. 2016b, MNRAS, 459, 3089

Courant, R., Friedrichs, K., & Lewy, H. 1928, MatAn, 100, 32

Cox, D. P., & Gómez, G. C. 2002, ApJS, 142, 261

Cullen, L., & Dehnen, W. 2010, MNRAS, 408, 669

Cummins, S. J., & Rudman, M. 1999, JCoPh, 152, 584

Dahlburg, R. B., & Picone, J. M. 1989, PhFlB, 1, 2153

Dai, W., & Woodward, P. R. 1994a, JCoPh, 111, 354

Dai, W., & Woodward, P. R. 1994b, JCoPh, 115, 485

Dai, W., & Woodward, P. R. 1998, ApJ, 494, 317

de Val-Borro, M., et al. 2006, MNRAS, 370, 529

Dedner, A., Kemm, F., Kröner, D., Munz, C.-D., Schnitzer, T., & Wesenberg, M. 2002, JCoPh, 175, 645

Dehnen, W. 2000a, AJ, 119, 800

Dehnen, W. 2000b, ApJ, 536, L39

Dehnen, W. 2001, MNRAS, 324, 273

Dehnen, W., & Aly, H. 2012, MNRAS, 425, 1068

Dehnen, W., & Read, J. I. 2011, EPJP, 126, 55

Diehl, S., Rockefeller, G., Fryer, C. L., Riethmiller, D., & Statler T. S,. 2015, PASA, 32, 48

Dipierro, G., & Laibe, G. 2017, MNRAS, 469, 1932

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

Dipierro, G., Price, D., Laibe, G., Hirsh, K., Cerioli, A., & Lodato G,. 2015, MNRAS, 453, L73

Dipierro, G., Laibe, G., Price, D. J., & Lodato, G. 2016, MNRAS, 459, L1

Dobbs, C. L., Glover, S. C. O., Clark, P. C., & Klessen, R. S. 2008, MNRAS, 389, 1097

Dobbs, C. L., Price, D. J., Pettitt, A. R., Bate, M. R., & Tricco T. S,. 2016, MNRAS, 461, 4482

Dolag, K., & Stasyszyn, F. 2009, MNRAS, 398, 1678

Dolag, K., Vazza, F., Brunetti, G., & Tormen, G. 2005, MNRAS, 364, 753

Dopcke, G., Glover, S. C. O., Clark, P. C., & Klessen, R. S. 2011, ApJ, 729, L3

Doğan, S., Nixon, C., King, A., & Price, D. J. 2015, MNRAS, 449, 1251

Draine, B. T., & Bertoldi, F. 1996, ApJ, 468, 269

Epstein, P. S. 1924, PhRv, 23, 710

Escala, A., Larson, R. B., Coppi, P. S., & Mardones, D. 2004, ApJ, 607, 765

Español, P., & Revenga, M. 2003, PhRvE, 67, 026705

Eswaran, V., & Pope, S. B. 1988, CF, 16, 257

Evrard, A. E. 1988, MNRAS, 235, 911

Facchini, S., Lodato, G., & Price, D. J. 2013, MNRAS, 433, 2142

Falle, S. A. E. G. 2003, MNRAS, 344, 1210

Falle, S. A. E. G., & Komissarov, S. S. 2001, JPlPh, 65, 29

Fan, L.-S., & Zhu, C. 1998, Principles of Gas-Solid Flows (Cambridge: Cambridge University Press)

Farr, W. M., & Bertschinger, E. 2007, ApJ, 663, 1420

Fassio, F., & Probstein, R. F. 1970, AIAAJ, 8, 772

Federrath, C., Klessen, R. S., & Schmidt, W. 2008, ApJ, 688, L79

Federrath, C., Roman-Duval, J., Klessen, R. S., Schmidt, W., & Mac Low M.-M,. 2010a, A&A, 512, A81

Federrath, C., Banerjee, R., Clark, P. C., & Klessen, R. S. 2010b, ApJ, 713, 269

Feng, H. Q., Lin, C. C., Chao, J. K., Wu, D. J., Lyu, L. H., & Lee L.-C,. 2007, JGRA, 112, 10104

Ferland, G. J., Peterson, B. M., Horne, K., Welsh, W. F., & Nahar S. N,. 1992, ApJ, 387, 95

Flebbe, O., Muenzel, S., Herold, H., Riffert, H., & Ruder, H. 1994, ApJ, 431, 754

Forgan, D., Price, D. J., & Bonnell, I. 2017, MNRAS, 466, 3406

Fryer, C. L., Hungerford, A. L., & Rockefeller, G. 2007, IJMPAD, 16, 941

Fulk, D. A., & Quinn, D. W. 1996, JCoPh, 126, 165

Furth, H. P., Killeen, J., & Rosenbluth, M. N. 1963, PhFl, 6, 459

Gafton, E., & Rosswog, S. 2011, MNRAS, 418, 770

Gammie, C. F. 2001, ApJ, 553, 174

García-Senz, D., Cabezón, R. M., Escartín, J. A., Ebinger, K. 2014, A&A, 570, A14

Gardiner, T. A., & Stone, J. M. 2005, JCoPh, 205, 509

Gardiner, T. A., & Stone, J. M. 2008, JCoPh, 227, 4123

Gingold, R. A., & Monaghan, J. J. 1977, MNRAS, 181, 375

Gingold, R. A., & Monaghan, J. J. 1978, MNRAS, 184, 481

Gingold, R. A., & Monaghan, J. J. 1980, MNRAS, 191, 897

Gingold, R. A., & Monaghan, J. J. 1982a, PASA, 4, 378

Gingold, R. A., & Monaghan, J. J. 1982b, JCoPh, 46, 429

Gingold, R. A., & Monaghan, J. J. 1983, MNRAS, 204, 715

Glover, S. C. O., & Clark, P. C. 2012, MNRAS, 421, 116

Glover, S. C. O., & Jappsen, A.-K. 2007, ApJ, 666, 1

Glover, S. C. O., & Mac Low, M.-M. 2007, ApJS, 169, 239

Glover, S. C. O., Federrath, C., Mac Low, M.-M., & Klessen, R. S. 2010, MNRAS, 404, 2

Goffrey, et al. 2017, A&A 600, A7

Goldreich, P., & Tremaine, S. 1979, ApJ, 233, 857

Goldreich, P., & Tremaine, S. 1980, ApJ, 241, 425

Goldsmith, P. F., & Langer, W. D. 1978, ApJ, 222, 881

Gray, J., Monaghan, J. J., & Swift, R. P. 2001, CMAME, 190, 6641

Graziani, C., Tzeferacos, P., Lee, D., Lamb, D. Q., Weide, K., Fatenejad, M., & Miller, J. 2015, ApJ, 802, 43

Hairer, E., Lubich, C., & Wanner, G. 2003, AcNum, 12, 399

Haworth, T. J., et al. 2016, PASA, 33, e053

Herant, M. 1994, MmSAI, 65, 1013

Hernquist, L. 1990, ApJ, 356, 359

Hernquist, L., & Barnes, J. E. 1990, ApJ, 349, 562

Hernquist, L., & Katz, N. 1989, ApJS, 70, 419

Hollenbach, D., & McKee, C. F. 1989, ApJ, 342, 306

Hopkins, P. F., & Raives, M. J. 2016, MNRAS, 455, 51

Hosking, J. G., & Whitworth, A. P. 2004, MNRAS, 347, 994

Hu, X. Y., & Adams, N. A. 2006, PhFl, 18, 101702

Hu, C.-Y., Naab, T., Walch, S., Moster, B. P., & Oser, L. 2014, MNRAS, 443, 1173

Hubber, D. A., Batty, C. P., McLeod, A., & Whitworth, A. P. 2011, A&A, 529, A27

Hubber, D. A., Allison, R. J., Smith, R., & Goodwin, S. P. 2013a, MNRAS, 430, 1599

Hubber, D. A., Walch, S., & Whitworth, A. P. 2013b, MNRAS, 430, 3261

Hubber, D. A., Falle, S. A. E. G., & Goodwin, S. P. 2013c, MNRAS, 432, 711

Hubble, E. P. 1930, ApJ, 71, 231

Hut, P., Makino, J., & McMillan, S. 1995, ApJ, 443, L93

Hutchison, M. A., Price, D. J., Laibe, G., & Maddison, S. T. 2016, MNRAS, 461, 742

Iaconi, R., Reichardt, T., Staff, J., De Marco, O., Passy, J.-C., Price, D., Wurster, J., & Herwig, F. 2017, MNRAS, 464, 4028

Iannuzzi, F., & Dolag, K. 2011, MNRAS, 417, 2846

Inutsuka, S.-I. 2002, JCoPh, 179, 238

Iwasaki, K. 2015, JCoPh, 302, 359

Jubelgas, M., Springel, V., & Dolag, K. 2004, MNRAS, 351, 423

Kaufmann, T., Mayer, L., Wadsley, J., Stadel, J., & Moore, B. 2007, MNRAS, 375, 53

Khomenko, E., Vitas, N., Collados, M., & de Vicente, A. 2017, A&A, 604, A66

Khoperskov, S. A., Vasiliev, E. O., Sobolev, A. M., & Khoperskov, A. V. 2013, MNRAS, 428, 2311

Kim, J., Balsara, D., & Mac Low, M.-M. 2001, JKAS, 34, 333

Kitsionas, S., et al. 2009, A&A, 508, 541

Krumholz, M. R. 2006, ApJ, 641, L45

Kuijken, K., & Dubinski, J. 1995a, MNRAS, 277, 1341

Kuijken, K., & Dubinski, J. 1995b, MNRAS, 277, 1341

Kwok, S. 1975, ApJ, 198, 583

Laibe, G., & Price, D. J. 2011, MNRAS, 418, 1491

Laibe, G., & Price, D. J. 2012a, MNRAS, 420, 2345

Laibe, G., & Price, D. J. 2012b, MNRAS, 420, 2365

Laibe, G., & Price, D. J. 2014a, MNRAS, 440, 2136

Laibe, G., & Price, D. J. 2014b, MNRAS, 440, 2147

Laibe, G., & Price, D. J. 2014c, MNRAS, 444, 1940

Larson, R. B. 1981, MNRAS, 194, 809

Lattanzio, J., Monaghan, J., Pongracic, H., & Schwarz, M. 1986, SIAM JSSC, 7, 591

Le Bourlot, J., Pineau des Forêts, G., & Flower, D. R. 1999, MNRAS, 305, 802

Lecoanet, D., et al. 2016, MNRAS, 455, 4274

Lense, J., & Thirring, H. 1918, PhyZ, 19, 156

Lewis, B. T., Bate, M. R., & Price, D. J. 2015, MNRAS, 451, 288

Liptai, D., Price, D. J., Wurster, J., & Bate, M. R. 2017, MNRAS, 465, 105

Lodato, G., & Price, D. J. 2010, MNRAS, 405, 1212

Lodato, G., & Pringle, J. E. 2007, MNRAS, 381, 1287

Lodato, G., & Rice, W. K. M. 2004, MNRAS, 351, 630

Lodato, G., Nayakshin, S., King, A. R., & Pringle, J. E. 2009, MNRAS, 398, 1392

Lombardi, J. C., Sills, A., Rasio, F. A., & Shapiro, S. L. 1999, JCoPh, 152, 687

Londrillo, P., & Del Zanna, L. 2000, ApJ, 530, 508

Long, K., & Murali, C. 1992, ApJ, 397, 44

Lorén-Aguilar, P., & Bate, M. R. 2014, MNRAS, 443, 927

Low, C., & Lynden-Bell, D. 1976, MNRAS, 176, 367

Lubow, S. H., & Ogilvie, G. I. 2000, ApJ, 538, 326

Lubow, S. H., Ogilvie, G. I., & Pringle, J. E. 2002, MNRAS, 337, 706

Lucy, L. B. 1977, AJ, 82, 1013

Lynden-Bell, D., & Pringle, J. E. 1974, MNRAS, 168, 603

Mac Low, M.-M., Balsara, D. S., Kim, J., & de Avillez, M. A. 2005, ApJ, 626, 864

Maddison, S. T. 1998, PhD thesis, Department of Mathematics, Monash University

Maddison, S. T., Murray, J. R., & Monaghan, J. J. 1996, PASA, 13, 66

Makino, J., & Aarseth, S. J. 1992, PASJ, 44, 141

Makino, J., Hut, P., Kaplan, M., & Saygın, H. 2006, NewA, 12, 124

Martin, R. G., Nixon, C., Armitage, P. J., Lubow, S. H., & Price D. J,. 2014a, ApJ, 790, L34

Martin, R. G., Nixon, C., Lubow, S. H., Armitage, P. J., Price, D. J., Doğan, S., & King, A. 2014b, ApJ, 792, L33

Martos, M., Hernandez, X., Yáñez, M., Moreno, E., & Pichardo B,. 2004, MNRAS, 350, L47

Matzner, C. D., & Levin, Y. 2005, ApJ, 628, 817

McMillan, P. J., & Dehnen, W. 2007, MNRAS, 378, 541

McNally, C. P., Lyra, W., & Passy, J.-C. 2012, ApJS, 201, 18

Merritt, D. 1996, AJ, 111, 2462

Meru, F., & Bate, M. R. 2011, MNRAS, 411, L1

Meru, F., & Bate, M. R. 2012, MNRAS, 427, 2022

Miller, M. C., & Lamb, F. K. 1993, ApJ, 413, L43

Miniati, F. 2010, JCoPh, 229, 3916

Miura, H., & Glass, I. I. 1982, RSPSA, 382, 373

Miyamoto, M., & Nagai, R. 1975, PASJ, 27, 533

Monaghan, J. J. 1989, JCoPh, 82, 1

Monaghan, J. J. 1992, ARA&A, 30, 543

Monaghan, J. J. 1997, JCoPh, 136, 298

Monaghan, J. J. 2000, JCoPh, 159, 290

Monaghan, J. J. 2002, MNRAS, 335, 843

Monaghan, J. J. 2005, RPPh, 68, 1703

Monaghan, J. J. 2012, AnRFM, 44, 323

Monaghan, J. J., & Kocharyan, A. 1995, CoPhC, 87, 225

Monaghan, J. J., & Lattanzio, J. C. 1985, A&A, 149, 135

Monaghan, J. J., & Lattanzio, J. C. 1986, A&A, 158, 207

Monaghan, J. J., & Lattanzio, J. C. 1991, ApJ, 375, 177

Monaghan, J. J., & Price, D. J. 2001, MNRAS, 328, 381

Monaghan, J. J., & Price, D. J. 2006, MNRAS, 365, 991

Morris, J. P. 1996a, PhD thesis, Monash University, Melbourne, Australia

Morris, J. P. 1996b, PASA, 13, 97

Morris, J. P., & Monaghan, J. J. 1997, JCoPh, 136, 41

Müller, K., Fedosov, D. A., & Gompper, G. 2015, JCoPh, 281, 301

Murray, J. R. 1996, MNRAS, 279, 402

Nagasawa, M., Nakamura, T., & Miyama, S. M. 1988, PASJ, 40, 691

Navarro, J. F., Frenk, C. S., & White, S. D. M. 1996, ApJ, 462, 563

Nealon, R., Price, D. J., & Nixon, C. J. 2015, MNRAS, 448, 1526

Nealon, R., Nixon, C., Price, D. J., & King, A. 2016, MNRAS, 455, L62

Nelson, R. P., & Langer, W. D. 1997, ApJ, 482, 796

Nelson, R. P., & Papaloizou, J. C. B. 2000, MNRAS, 315, 570

Neufeld, D. A., & Kaufman, M. J. 1993, ApJ, 418, 263

Neufeld, D. A., Lepp, S., & Melnick, G. J. 1995, ApJS, 100, 132

Nixon, C. J., King, A. R., & Price, D. J. 2012a, MNRAS, 422, 2547

Nixon, C., King, A., Price, D., & Frank, J. 2012b, ApJ, 757, L24

Nixon, C., King, A., & Price, D. 2013, MNRAS, 434, 1946

Nordlund, Å. K., & Padoan, P. 1999, in Interstellar Turbulence, eds. J. Franco, & A. Carraminana (Cambridge: Cambridge University Press), 218 (`arXiv:astro-ph/9810074`)

O'Sullivan, S., & Downes, T. P. 2006, MNRAS, 366, 1329

Ogilvie, G. I. 1999, MNRAS, 304, 557

Ogilvie, G. I., & Lubow, S. H. 2002, MNRAS, 330, 950

Omelyan, I. P., Mryglod, I. M., & Folk, R. 2002, CoPhC, 146, 188

Orszag, S. A., & Tang, C.-M. 1979, JFM, 90, 129

Ostriker, E. C., Gammie, C. F., & Stone, J. M. 1999, ApJ, 513, 259

Paardekooper, S.-J., & Mellema, G. 2004, A&A, 425, L9

Paardekooper, S.-J., & Mellema, G. 2006, A&A, 453, 1129

Pandey, B. P., & Wardle, M. 2008, MNRAS, 385, 2269

Papaloizou, J. C. B., & Pringle, J. E. 1983, MNRAS, 202, 1181

Passy, J.-C., et al. 2012, ApJ, 744, 52

Paxton, B., Bildsten, L., Dotter, A., Herwig, F., Lesaffre, P., & Timmes, F. 2011, ApJS, 192, 3

Pettitt, A. R., Dobbs, C. L., Acreman, D. M., & Price, D. J. 2014, MNRAS, 444, 919

Pettitt, A. R., Dobbs, C. L., Acreman, D. M., & Bate, M. R. 2015, MNRAS, 449, 3911

Phillips, G. J. 1982, PASA, 4, 371

Phillips, G. J. 1986a, MNRAS, 221, 571

Phillips, G. J. 1986b, MNRAS, 222, 111

Phillips, G. J., & Monaghan, J. J. 1985, MNRAS, 216, 883

Pichardo, B., Martos, M., Moreno, E., & Espresate, J. 2003, ApJ, 582, 230

Picone, J. M., & Dahlburg, R. B. 1991, PhFlB, 3, 29

Plummer, H. C. 1911, MNRAS, 71, 460

Politano, H., Pouquet, A., & Sulem, P. L. 1989, PhFlB, 1, 2330

Potekhin, A. Y., & Chabrier, G. 2010, CoPP, 50, 82

Powell, K. G., Roe, P. L., Linde, T. J., Gombosi, T. I., & de Zeeuw D. L,. 1999, JCoPh, 154, 284

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, Numerical Recipes in FORTRAN. The Art of Scientific Computing (Cambridge: Cambridge University Press)

Price, D. J. 2004, PhD thesis, University of Cambridge, Cambridge, UK. astro-ph/0507472

Price, D. J. 2007, PASA, 24, 159

Price, D. J. 2008, JCoPh, 227, 10040

Price, D. J. 2010, MNRAS, 401, 1475

Price, D. J. 2012a, JCoPh, 231, 759

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press

Price, D. J. 2012b, MNRAS, 420, L33
Price, D. J., & Bate, M. R. 2007, MNRAS, 377, 77
Price, D. J., & Bate, M. R. 2009, MNRAS, 398, 33
Price, D. J., & Federrath, C. 2010, MNRAS, 406, 1659
Price, D. J., & Laibe, G. 2015a, MNRAS, 451, 5332
Price, D. J., & Laibe, G. 2015b, MNRAS, 454, 2320
Price, D. J., & Monaghan, J. J. 2004a, MNRAS, 348, 123
Price, D. J., & Monaghan, J. J. 2004b, MNRAS, 348, 139
Price, D. J., & Monaghan, J. J. 2005, MNRAS, 364, 384
Price, D. J., & Monaghan, J. J. 2007, MNRAS, 374, 1347
Price, D. J., Federrath, C., & Brunt, C. M. 2011, ApJ, 727, L21
Price, D. J., Tricco, T. S., & Bate, M. R. 2012, MNRAS, 423, L45
Priest, E. R. 1985, RPPh, 48, 955
Pringle, J. E. 1981, ARA&A, 19, 137
Pringle, J. E. 1992, MNRAS, 258, 811
Puri, K., & Ramachandran, P. 2014, JCoPh, 256, 308
Quinn, T., Katz, N., Stadel, J., & Lake, G. 1997, ArXiv Astrophysics e-prints
Quinn, T., Perrine, R. P., Richardson, D. C., & Barnes, R. 2010, AJ, 139, 803
Ragusa, E., Lodato, G., & Price, D. J. 2016, MNRAS, 460, 1243
Ragusa, E., Dipierro, G., Lodato, G., Laibe, G., & Price, D. J. 2017, MNRAS, 464, 1449
Read, J. I., & Hayfield, T. 2012, MNRAS, 422, 3037
Rees, M. J. 1976, MNRAS, 176, 483
Rein, H., & Tremaine, S. 2011, MNRAS, 415, 3168
Richardson, M. L. A., Scannapieco, E., Devriendt, J., Slyz, A., Thacker, R. J., Dubois, Y., Wurster, J., & Silk, J. 2016, ApJ, 825, 83
Robertson, B. E., Kravtsov, A. V., Gnedin, N. Y., Abel, T., & Rudd D. H,. 2010, MNRAS, 401, 2463
Rogers, F. J., & Nayfonov, A. 2002, ApJ, 576, 1064
Rosotti, G. P., Lodato, G., & Price, D. J. 2012, MNRAS, 425, 1958
Rosswog, S. 2009, NewAR, 53, 78
Rosswog, S. 2015, MNRAS, 448, 3628
Rosswog, S., & Price, D. 2007, MNRAS, 379, 915
Rosswog, S., Ramirez-Ruiz, E., & Hix, W. R. 2009, ApJ, 695, 404
Ryu, D., & Jones, T. W. 1995, ApJ, 442, 228
Ryu, D., Jones, T. W., & Frank, A. 1995, ApJ, 452, 785
Saitoh, T. R., & Makino, J. 2009, ApJ, 697, L99
Saitoh, T. R., & Makino, J. 2010, PASJ, 62, 301
Saumon, D., Chabrier, G., & van Horn, H. M. 1995, ApJS, 99, 713
Schmidt, W., Hillebrandt, W., & Niemeyer, J. C. 2006, CF, 35, 353
Schoenberg, I. J. 1946, QApMa, 4, 45
Schwarzschild, K. 1916, Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin), 189
Sedov, L. I. 1959, Similarity and Dimensional Methods in Mechanics (New York: Academic Press)
Sembach, K. R., Howk, J. C., Ryans, R. S. I., & Keenan, F. P. 2000, ApJ, 528, 310
Sembolini, F., et al. 2016, MNRAS, 457, 4063
Shakura, N. I., & Sunyaev, R. A. 1973, A&A, 24, 337
Shu, F. H., Galli, D., Lizano, S., & Cai, M. 2006, ApJ, 647, 382
Sijacki, D., & Springel, V. 2006, MNRAS, 371, 1025
Sod, G. A. 1978, JCoPh, 27, 1
Sofue, Y. 2012, PASJ, 64, 75
Springel, V. 2005, MNRAS, 364, 1105
Springel, V. 2010, ARA&A, 48, 391
Springel, V., & Hernquist, L. 2002, MNRAS, 333, 649
Springel, V., Yoshida, N., & White, S. D. M. 2001, NewA, 6, 79
Springel, V., Di Matteo, T., & Hernquist, L. 2005, MNRAS, 361, 776
Steinmetz, M., & Mueller, E. 1993, A&A, 268, 391
Stepinski, T. F., & Valageas, P. 1996, A&A, 309, 301
Stone, J. M., Hawley, J. F., Evans, C. R., & Norman, M. L. 1992, ApJ, 388, 415
Stone, J. M., Gardiner, T. A., Teuben, P., Hawley, J. F., & Simon J. B,. 2008, ApJS, 178, 137
Sutherland, R. S., & Dopita, M. A. 1993, ApJS, 88, 253
Suttner, G., Smith, M. D., Yorke, H. W., & Zinnecker, H. 1997, A&A, 318, 595
Syrovatskii, S. I. 1981, ARA&A, 19, 163
Takahashi, K., Yamada, S., & Yamada 2013, JPlPh, 79, 335
Tasker, E. J., Brunino, R., Mitchell, N. L., Michielsen, D., Hopton S., Pearce, F. R., Bryan, G. L., & Theuns, T. 2008, MNRAS, 390, 1267
Taylor, G. 1950a, RSEPS, 201, 159
Taylor, G. 1950b, RSEPS, 201, 175
Taylor, G. I., & Green, A. E. 1937, RSEPS, 158, 499
Tejeda, E., & Rosswog, S. 2013, MNRAS, 433, 1930
Teyssier, R. 2002, A&A, 385, 337
Thacker, R. J., & Couchman, H. M. P. 2006, CoPhC, 174, 540
Thacker, R. J., Tittley, E. R., Pearce, F. R., Couchman, H. M. P., & Thomas, P. A. 2000, MNRAS, 319, 619
Thomas, P. A., & Couchman, H. M. P. 1992, MNRAS, 257, 11
Timmes, F. X., & Arnett, D. 1999, ApJS, 125, 277
Timmes, F. X., & Swesty, F. D. 2000, ApJS, 126, 501
Tóth, G. 2000, JCoPh, 161, 605
Toupin, S., Braun, K., Siess, L., Jorissen, A., Gail, H.-P., & Price, D. 2015a, in EAS Publications Series, Vol. 71–72, The Physics of Evolved Stars: A Conference Dedicated to the Memory of Olivier Chesneau, 173, doi:10.1051/eas/1571038
Toupin, S., Braun, K., Siess, L., Jorissen, A., & Price, D. 2015b, in ASP Conf. Ser., Vol. 497, Why Galaxies Care About AGB Stars III: A Closer Look in Space and Time, eds. F. Kerschbaum, R. F. Wing, & J. Hron (San Francisco: Astronomical Society of the Pacific), 225
Townsend, R. H. D. 2009, ApJS, 181, 391
Tricco, T. S., & Price, D. J. 2012, JCoPh, 231, 7214
Tricco, T. S., & Price, D. J. 2013, MNRAS, 436, 2810
Tricco, T. S., Price, D. J., & Bate, M. R. 2016a, JCoPh, 322, 326
Tricco, T. S., Price, D. J., & Federrath, C. 2016b, MNRAS, 461, 1260
Tricco, T. S., Price, D. J., & Laibe, G. 2017, MNRAS, 471, L52
Tsukamoto, Y., Iwasaki, K., & Inutsuka, S.-i. 2013, MNRAS, 434, 2593
Tsukamoto, Y., Iwasaki, K., Okuzumi, S., Machida, M. N., & Inutsuka S,. 2015, ApJ, 810, L26
Tuckerman, M., Berne, B. J., & Martyna, G. J. 1992, JChPh, 97, 1990
Uhlenbeck, G. E., & Ornstein, L. S. 1930, PhRv, 36, 823
Valdarnini, R. 2011, A&A, 526, A158
Valdarnini, R. 2012, A&A, 546, A45
Valdarnini, R. 2016, ApJ, 831, 103
van Boekel, R., et al. 2017, ApJ, 837, 132
Vanaverbeke, S., Keppens, R., Poedts, S., & Boffin, H. 2009, CoPhC, 180, 1164
Vanaverbeke, S., Keppens, R., & Poedts, S. 2014, CoPhC, 185, 1053
Vazquez-Semadeni, E. 1994, ApJ, 423, 681
Verlet, L. 1967, PhRv, 159, 98
Violeau, D., & Leroy, A. 2014, JCoPh, 256, 388

Vogt, D., & Letelier, P. S. 2011, MNRAS, 411, 2371

Von Neumann, J., & Richtmyer, R. D. 1950, JAP, 21, 232

Wada, K., & Koda, J. 2001, PASJ, 53, 1163

Wadsley, J. W., Stadel, J., & Quinn, T. 2004, NewA, 9, 137

Wadsley, J. W., Veeravalli, G., & Couchman, H. M. P. 2008, MNRAS, 387, 427

Wadsley, J. W., Keller, B. W., & Quinn, T. R. 2017, MNRAS, 471, 2357

Wang, H.-H., Klessen, R. S., Dullemond, C. P., van den Bosch, F. C., & Fuchs, B. 2010, MNRAS, 407, 705

Wang, Y., Zhao, H., Mao, S., & Rich, R. M. 2012, MNRAS, 427, 1429

Wardle, M., & Ng, C. 1999, MNRAS, 303, 239

Watkins, S. J., Bhattal, A. S., Francis, N., Turner, J. A., & Whitworth, A. P. 1996, A&AS, 119, 177

Weidenschilling, S. J. 1977, MNRAS, 180, 57

Weingartner, J. C., & Draine, B. T. 2001, ApJ, 563, 842

Wendland H. 1995, Adv. Comput. Math., 4, 389

Whipple, F. L. 1972, in Proceedings of From Plasma to Planet, Proceedings of the Twenty-First Nobel Symposium Held 6–10 September, 1971 at Saltsjöbaden, near Stockholm, Sweden, ed. A. Elvius (New York: Wiley Interscience Division), 211

Whitehouse, S. C., & Bate, M. R. 2004, MNRAS, 353, 1078

Whitehouse, S. C., & Bate, M. R. 2006, MNRAS, 367, 32

Whitehouse, S. C., Bate, M. R., & Monaghan, J. J. 2005, MNRAS, 364, 1367

Widrow, L. M., & Dubinski, J. 2005, ApJ, 631, 838

Widrow, L. M., Pym, B., & Dubinski, J. 2008, ApJ, 679, 1239

Wilde, O. 1917, Lady Windermere's Fan: A Play about a Good Woman (17th edn; London: Methuen & Co. Ltd.)

Woitke, P., & Helling, C. 2003, A&A, 399, 297

Wolfire, M. G., Hollenbach, D., McKee, C. F., Tielens, A. G. G. M., & Bakes, E. L. O. 1995, ApJ, 443, 152

Wolfire, M. G., McKee, C. F., Hollenbach, D., & Tielens, A. G. G. M. 2003, ApJ, 587, 278

Worpel, H. 2015, PhD thesis, Monash University

Wu, C. C. 1988, J. Geophys. Res., 93, 987

Wurster, J. 2016, PASA, 33, e041

Wurster, J., & Thacker, R. J. 2013a, MNRAS, 431, 539

Wurster, J., & Thacker, R. J. 2013b, MNRAS, 431, 2513

Wurster, J., Price, D., & Ayliffe, B. 2014, MNRAS, 444, 1104

Wurster, J., Price, D. J., & Bate, M. R. 2016, MNRAS, 457, 1037

Wurster, J., Price, D. J., & Bate, M. R. 2017, MNRAS, 466, 1788

Wurster, J., Bate, M. R., & Price, D. J. 2018, MNRAS, 475, 1859

Youdin, A. N., & Goodman, J. 2005, ApJ, 620, 459

Yurin, D., & Springel, V. 2014, MNRAS, 444, 62

Ziegler, U., Yorke, H. W., & Kaisig, M. 1996, A&A, 305, 114

Zurek, W. H., & Benz, W. 1986, ApJ, 308, 123

A DETAILS OF CODE IMPLEMENTATION

Figure A1 shows the basic structure of the code. The core of the code is the timestepping loop, while the most time consuming part are the repeated calls to evaluate density and acceleration on the particles via sums over neighbouring particles. Further details of these steps are given in Section A.3.

A.1 Smoothing kernels

A PYTHON script distributed with PHANTOM can be used to generate the code module for alternative smoothing kernels, including symbolic manipulation of piecewise functions using SYMPY to obtain the relevant functions needed for gravitational force softening (see below). Pre-output modules for the six kernels described in Section 2.1.6 are included in the source code, and the code can be recompiled with any of these replacing the default cubic spline on the command line, e.g. `make KERNEL=quintic`.

Table A1. Particle types in PHANTOM. The density and smoothing length of each type is computed only from neighbours of the same type (c.f. Section 2.13.3). Sink particles are handled separately in a different set of arrays.

ID	Type	Description
1	Gas	Default type, all forces applied
2	Dust	Drag, external & gravitational forces
3	Boundary	Velocity and gas properties fixed
4	Star	External and gravitational forces
5	Dark matter	Same as star, but different mass
6	Bulge	Same as star, but different mass
0	Unknown	Usually dead particles

The double hump kernel functions used in the two-fluid dust algorithm (Section 2.13.4) can be generated automatically from the corresponding density kernel by the `kernels.py` script distributed with PHANTOM, as described in Section 2.12.2. The pre-generated modules implementing each of the kernels described in Section 2.1.6 hence automatically contain the corresponding double hump kernel function, which is used to compute the drag terms.

A.2 Particle types

Particles can be assigned with a 'type' from the list (see Table A1 in the appendix). The main use of this is to be able to apply different sets of forces to certain particle types (see description for each type). Densities and smoothing lengths are self-consistently computed for *all* of these types except for 'dead' particles which are excluded from the tree build and boundary particles whose properties are fixed. However, the kernel interpolations used for these involve only neighbours of the *same type*. Particle masses in PHANTOM are fixed to be equal for all particles of the same type, to avoid problems with unequal mass particles (e.g. Monaghan & Price 2006). We use adaptive gravitational force softening for all particle types, both SPH and *N*-body (see Section 2.12.3). Sink particles are handled separately to these types, being stored in a separate set of arrays, carry only a fixed softening length which is set to zero by default and compute their mutual gravitational force without approximation (see Section 2.8).

A.3 Evaluating density and acceleration

A.3.1 Tree build

PHANTOM uses a *kd*-tree for nearest neighbour finding and to compute long range gravitational accelerations. The procedure for the tree build is given in Figure A2. We use a stack, rather than recursive subroutines, for efficiency and to aid parallelisation (the parallelisation strategy for the tree build is discussed further in Section A.4.2). The stack initially contains only the highest level node for each thread (in serial, this would be the root node). We loop over all nodes on the stack and call a subroutine to compute the node properties, namely the centre of mass position, the node size, s, which is the radius of a sphere containing all the particles centred on the centre of mass, the maximum smoothing length for all the particles

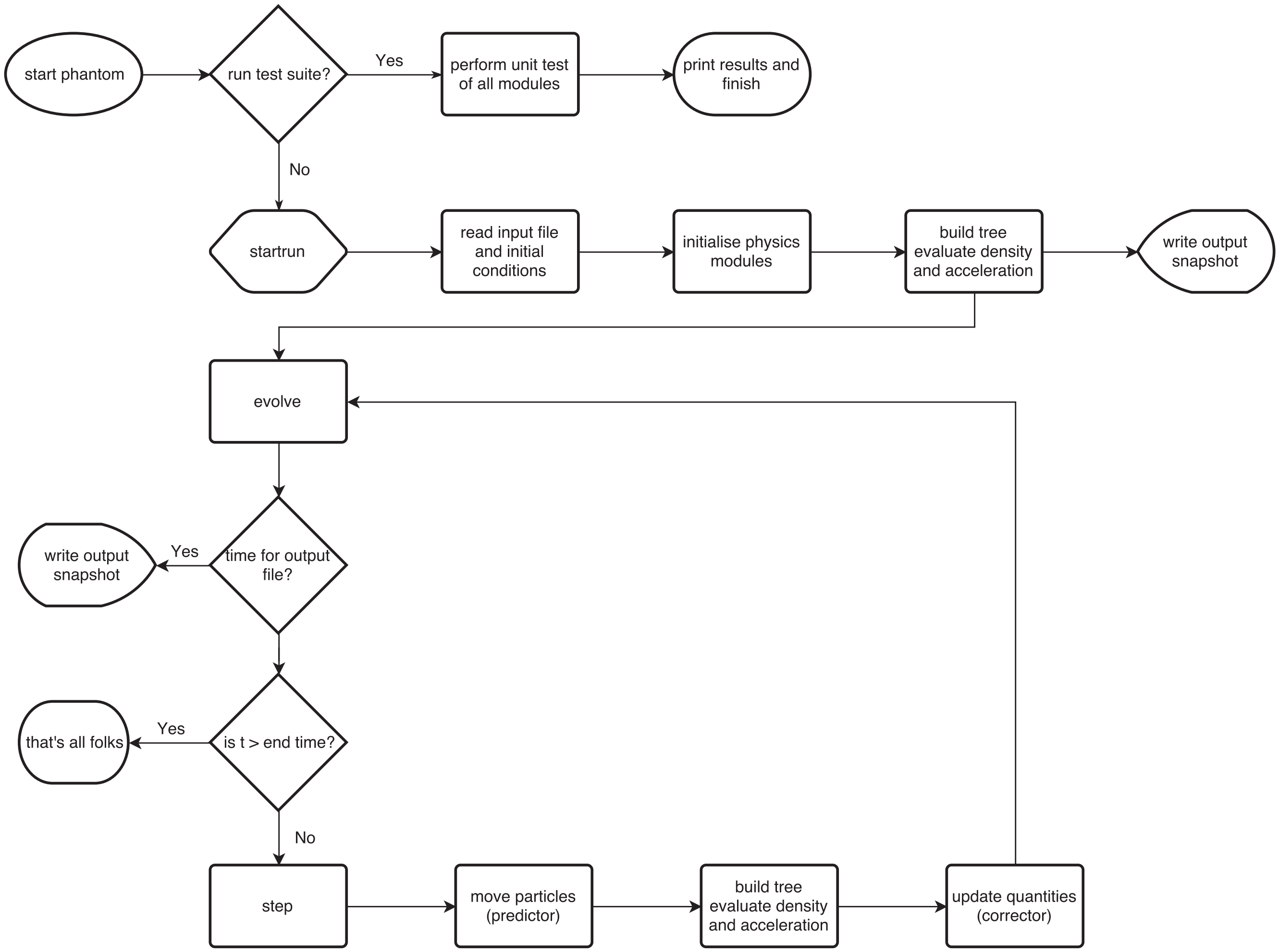
start phantom
run test suite?
Yes
perform unit test of all modules
print results and finish
No
startrun
read input file and initial conditions
initialise physics modules
build tree evaluate density and acceleration
write output snapshot
evolve
write output snapshot
Yes
time for output file?
that's all folks
Yes
is t > end time?
No
step
move particles (predictor)
build tree evaluate density and acceleration
update quantities (corrector)

Figure A1. Flowchart showing the basic structure of PHANTOM. To the user what appears is a sequence of output files written at discrete time intervals. The core of the code is the timestepping loop, while most of the computational cost is spent building the tree and evaluating density and acceleration by summing over neighbours.

```
construct_root_node(rootnode,bounds)
push_onto_stack(rootnode,bounds)
number on stack = 1
do while (number on stack > 0)
   pop_off_stack(node,bounds)
   construct_node(node,bounds,boundsl,boundsr)
   if (node was split)
      push_onto_stack (leftchild,boundsl)
      push_onto_stack (rightchild,boundsr)
      number on stack += 2
   endif
enddo
```

Figure A2. Pseudo-code for the tree build. The *construct_node* procedure computes, for a given node, the centre of mass, size, maximum smoothing length, quadrupole moments, and the child and parent pointers and the boundaries of the child nodes.

contained within the node, pointers to the two node children and the parent node, and, if self-gravity is used, the total mass in the node as well as the quadrupole moment (see Section 2.12). The construct_node subroutine also decides whether or not the node should be split (i.e. if the number of particles $>N_{\rm min}$) and returns the indices and boundaries of the resultant child nodes.

We access the particles by storing an integer array containing the index of the first particle in each node (firstinnode), and using a linked list where each particle stores the index of the next particle in the node (next), with an index of zero indicating the end of the list. During the tree build, we start with only the root node, so firstinnode is simply the first particle that is not dead or accreted and the next array is filled to contain the next non-dead-or-accreted particle using a simple loop. In the construct_node routine, we convert this to a simple list of the particles in each node and use this temporary list to update the linked list when creating the child nodes (i.e., by setting firstinnode to zero for the parent nodes, and filling firstinnode and next for the child nodes based on whether the particle position is to the 'left' or the 'right' of the bisected parent node).

The tree structure itself stores eight quantities without self-gravity, requiring 52 bytes per particle (x,y,z,size,hmax: 5 × 8-byte double precision; leftchild, rightchild, parent: 3 × 4-byte integer). With self-gravity, we store 15 quantities (mass and quads(6), i.e. seven additional 8-byte doubles) requiring

```
rcut = size(node) + radkern*hmax(node)
add_to_stack(root node)
number on stack = 1
do while (number on stack > 0)
   nodem = stack(nstack)
   distance = node - nodem
   if (distance < rcut + size(nodem))
      if (node is leaf node)
         ipart = firstinnode(nodem)
         do while(ipart > 0)
            add particle to neighbour list
            nneigh = nneigh + 1
            if (nneigh <= cache size)
               cache positions
            endif
            ipart = next(ipart)
         enddo
      else
         add_to_stack(leftchild)
         add_to_stack(rightchild)
         number on stack += 2
      endif
   endif
enddo
```

Figure A3. Pseudo-code for the neighbour search (referred to as the *get_neigh* routine in Figure A4).

108 bytes per particle. We implement the node indexing scheme outlined by Gafton & Rosswog (2011) where the tree nodes on each level l are stored starting from 2^{l-1}, where level 1 is the 'root node' containing all the particles, to avoid the need for thread locking when different sections of the tree are built in parallel. However, the requirement of allocating storage for all leaf nodes on all levels regardless of whether or not they contain particles either limits the maximum extent of the tree or can lead to prohibitive memory requirements, particularly for problems with high dynamic range, such as star formation, where a small fraction of the particles collapse to high density. Hence, we use this indexing scheme only up to a maximum level (`maxlevel_indexed`) which is set such that $2^{\texttt{maxlevel\_indexed}}$ is less than the maximum number of particles (`maxp`). We do, however, allow the tree build to proceed beyond this level, whereupon the `leftchild`, `rightchild` and `parent` indices are used and additional nodes are added in the order that they are created (requiring limited thread locking).

A.3.2 Neighbour search

The neighbour search for a given 'leaf node', n, proceeds from the top down. As with the tree build, this is implemented using a stack, which initially contains only the root node. The procedure is summarised in Figure A3. We loop over the nodes on the stack, checking the criterion

$$r_{nm}^2 < (s_n + s_m + R_{\rm kern} h^n_{\rm max})^2, \tag{A1}$$

where $r_{nm}^2 \equiv (x_n - x_m)^2 + (y_n - y_m)^2 + (z_n - z_m)^2$ is the square of the separation between the node centres and s is the node size. Any node m satisfying this criteria, that is not a leaf node, has its children added to the stack and the search continues. If node m is a leaf node, then the list of particles it contains are added to the trial neighbour list and the positions cached. The end product is a list of trial neighbours (`listneigh`), its length `nneigh`, and a cache containing the trial neighbour positions (`xyzcache`) up to some maximum size (12 000 by default, the exact size not being important except that it is negligible compared to the number of particles). Trial neighbours exceeding this number are retrieved directly from memory during the density calculation rather than from the cache. This occurs rarely, but the overflow mechanism allows for the possibility of a few particles with a large number of neighbours, as happens under certain circumstances.

A.3.3 Density and force calculation

Once the neighbour list has been obtained for a given leaf node, we proceed to perform density iterations for each member particle. The neighbours only have to be re-cached if the smoothing length of a given particle exceeds $h_{\rm max}$ for the node, which is sufficiently rare so as not to influence the code performance significantly. In the original version of PHANTOM (on the `nogravity` branch), this neighbour cache was re-used immediately for the force calculation but this is no longer the case on the master branch (see Section A.3.4). Figure A4 summarises the resulting procedure for calculating the density.

The corresponding force calculation is given in Figure A5.

A.3.4 SPH in a single loop

A key difference in the force calculation compared to the density calculation (Section 2.1.2) is that computation of the acceleration [Equation (34)] involves searching for neighbours using both h_a and h_b. One may avoid this requirement, and the need to store various intermediate quantities, by noticing that the h_b term can be computed by 'giving back' a contribution to one's neighbours. In this way, the whole SPH algorithm can be performed in a single giant outer loop, but with multiple loops over the same set of neighbours, following the outline in Figure A4. This also greatly simplifies the neighbour search, since one can simply search for neighbours within a known search radius ($2h_a$) without needing to search for neighbours that 'might' contribute if their h_b is large. Hence, a simple fixed grid can be used to find neighbours, as already discussed in Section 2.1.7, and the same neighbour positions can be efficiently cached and re-used (one or more times for the density iterations, and once for the force calculation). This is the original reason we decided to average the dissipation terms as above, since at the end of the density loop one can immediately compute quantities that depend on ρ_a (i.e. P_a and q^a_{ab}) and use these to 'give back' the b contribution to ones neighbours. This means that the density and force calculations can be done in a single subroutine with effectively only one neighbour call, in principle, saving a factor of two in computational cost.

The two disadvantages to this approach are (i) that particles may receive simultaneous updates in a parallel calculation, requiring locking which hurts the overall scalability of the code and (ii) that when using individual timesteps only a few particles are updated at any given timestep, but with the simple neighbour search algorithms one is required to loop over all the inactive particles to see if they *might* contribute to an active particle. Hence, although we employed this approach for a number of years, we have now abandoned it for a more traditional approach, where the density and force calculations

```
!$omp parallel do
do node = 1, number of nodes
   if (node is leaf node)
      call get_neigh(node,listneigh,nneigh)
      i = firstinnode(node)
      do while (i > 0)
         do while not converged
            if (h > hmax(node)) call get_neigh
            do k=1,nneigh
               j = listneigh(k)
               if (n <= cache size)
                  get j position from cache
               else
                  get j position from memory
               endif
               if (actual neighbour)
                  evaluate kernel and dwdh
                  add to density sum
                  add to gradh sum
                  add to div v sum
               endif
            enddo
            update h
            check convergence
         enddo
         i = next(i)
      enddo
   endif
enddo
!$omp end parallel do
```

Figure A4. Pseudo-code for the density evaluation in PHANTOM, showing how Equations (3)–(5) are computed. The force evaluation [evaluating Equations (34) and (35)] is similar except that *get_neigh* returns neighbours within the kernel radius computed with both h_i and h_j and there is no need to update/iterate h (see Figure A5).

are done in separate subroutines and the *k*d-tree is used to search for neighbours checking both h_a and h_b for the force calculation.

A.4 OpenMP parallelisation

A.4.1 Density and force calculation

Shared memory parallelisation of the density and force calculation is described in pseudo-code in Figure A4. The parallelisation is done over the 'leaf nodes', each containing around 10 particles. Since the leaf nodes can be done in any order, this can be parallelised with a simple `$omp parallel do` statement. The neighbour search is performed once for each leaf node, so each thread must store a private cache of the neighbour list. This is not an onerous requirement, but care is required to ensure that sufficient per-thread memory is available. This usually requires setting the `OMP_STACKSIZE` environment variable at runtime. No thread locking is required during the density or force evaluations (unless the single loop algorithm is employed; see Section A.3.4) and the threads can be scheduled at runtime to give the best performance using either dynamic, guided or static scheduling (the default is dynamic). Static scheduling is faster when there are few density contrasts and the work per node is similar, e.g. for subsonic turbulence in a periodic box (c.f. Price 2012b).

```
!$omp parallel do
do node = 1, number of nodes
   if (node is leaf node)
      call get_neigh(node,listneigh,nneigh,fnode)
      i = firstinnode(node)
      do while (i > 0)
         do n=1,nneigh
            j = listneigh(n)
            if (n <= cache size)
               get j position from cache
            else
               get j position from memory
            endif
            if (dr < Rkern*hi or dr < Rkern*hj)
               evaluate kernel and softening
               add to force sum
            else
               add 1/r^2 forces to force sum
            endif
         enddo
         get i distance from node centre
         expand fnode at position of i
         add long range terms to force sum
         i = next(i)
      enddo
   endif
enddo
!$omp end parallel do
```

Figure A5. Pseudo-code for the force calculation, showing how the short and long-range accelerations caused by self-gravity are computed. The quantity *fnode* refers to the long-range gravitational force on the node computed from interaction with distant nodes not satisfying the tree-opening criterion.

A.4.2 Parallel tree build

We use a domain decomposition to parallelise the tree build, similar to Gafton & Rosswog (2011). That is, we first build the tree nodes as in Figure A2, starting from the root-level node and proceeding to its children and so forth, putting each node into a queue until the number of nodes in the queue is equal to the number of OPENMP threads. Since the queue itself is executed in serial, we parallelise the loop over the particles inside the `construct_node` routine during the construction of each node. Once the queue is built, each thread proceeds to build its own sub-tree independently.

By default, we place each new node into a stack, so the only locking required during the build of each sub-tree is to increment the stack counter. We avoid this by adopting the indexing scheme proposed by Gafton & Rosswog (2011) (discussed in Section 2.1.7) where the levels of the tree are stored contiguously in memory. However, to avoid excessive memory consumption, we only use this scheme while $2^{n_{\rm level}} < n_{\rm part}$. For levels deeper than this, we revert to using a stack which therefore requires a (small) critical section around the increment of the stack counter.

A.4.3 Performance

Figure A6 shows strong scaling results for the pure OPENMP code. For the scaling tests, we wanted to employ a more representative problem than the idealised tests shown in Section 5. With this in mind, we tested the scaling using a problem involving the collapse of a molecular cloud core to form a star, as described in Section 6.3.

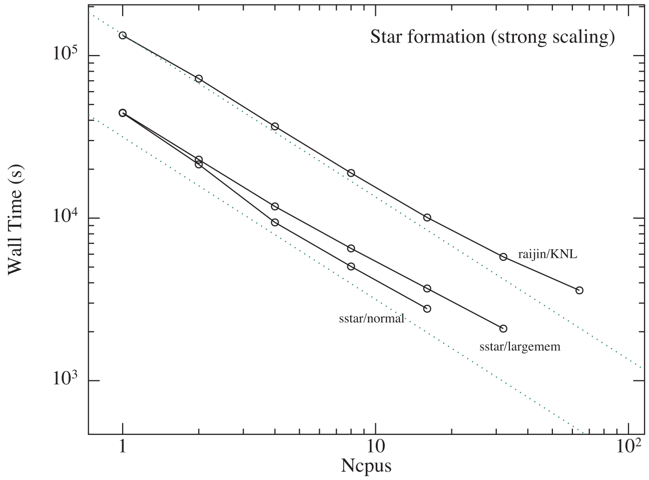

Figure A6. Strong scaling results for the pure OpenMP code for the magnetised star formation problem, showing wall time as a function of the number of OpenMP threads.

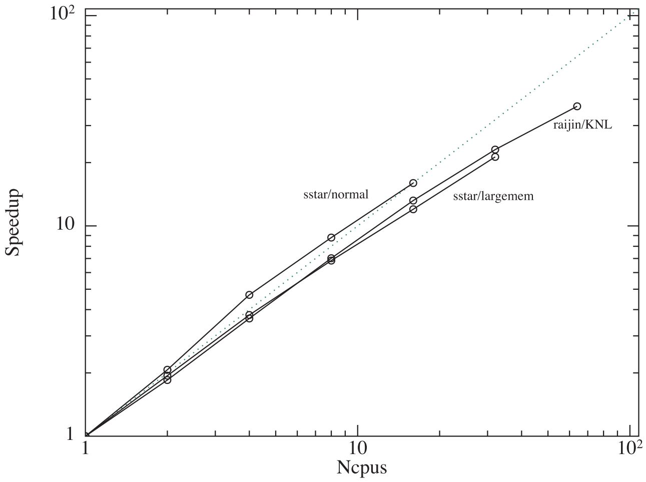

Figure A7. Strong scaling results for the pure OpenMP code for the magnetised star formation problem, showing speedup as a function of the number of OpenMP threads.

We used 10^6 particles in the initial sphere, corresponding to 1.44 million particles in total. We recorded the wall time of each simulation evolved for one free-fall time of the collapsing sphere, corresponding to $t = 0.88$ in code units.

To show scaling of the OpenMP code to a reasonable number of cpus, we performed the test on the Knights Landing™ (KNL) nodes of the Raijin supercomputer (the main supercomputer of the National Computational Infrastructure in Australia). Each CPU of this machine is an Intel® Xeon Phi™ CPU 7230 with a clock speed of 1.30 GHz and a 1 024 kB cache size. The results shown in Figure A6 demonstrate strong scaling to 64 CPUs on this architecture. We used the Intel® Fortran Compiler to compile the code. Timings are also shown for the same calculation performed on two different nodes of the Swinstar supercomputer, namely the ‘largemem’ queue, consisting of up to 32 CPUs using Intel® Xeon™ E7-8837 chips running at 2.66 GHz with cache size 24 576 kB, and the ‘normal’ queue, consisting of up to 16 Intel® Xeon™ E5-2660 chips running at 2.20 GHz with cache size 20 480 kB. Our shortest wall time is achieved on the largemem queue using 32 CPUs. Figure A7 shows the corresponding speedup (wall time on single cpu divided by wall time on multiple cpus) for each architecture. We find the best scaling on the Swinstar ‘normal’ nodes, which show super-linear scaling and 100% parallel efficiency on 16 cpus. The parallel efficiency on KNL is 60% on 64 cpus.

A.5 MPI parallelisation

The code has been recently parallelised for distributed memory architecture using the Message Passing Interface (MPI), using a hybrid MPI-OpenMP implementation. However, Phantom does not yet compete with GADGET variants in terms of the ability to use large numbers of particles [e.g. the most recent calculations by Bocquet et al. (2016) employed more than 10^{11} particles], since almost all of our published simulations to date have been performed with the OpenMP code. Nevertheless, we describe the implementation below.

We decompose the domain using the *k*d-tree in a similar manner to the OpenMP parallelisation strategy described above. That is, we build a global tree from the top down until the number of tree nodes exceeds the number of MPI threads. Storing a global tree across all MPI threads is not too memory intensive since there are only as many nodes in the global tree as there are MPI threads. Each thread then proceeds to build its own independent subtree. During the neighbour search (performed for each leaf node of the tree), we then flag if a node hits parts of the tree that require remote contributions. If this is the case, we send the information for all active particles contained within the leaf node to the remote processor. Once all nodes have computed density (or force) on their local particles, they proceed to compute the contributions of local particles to the density sums of particles they have received. The results of these partial summations are then passed back to their host processors.

Particles are strictly assigned to a thread by their location in the tree. During the tree build, we exchange particles between threads to ensure that all particles are hosted by the thread allocated to their subdomain.

Within each MPI domain, the OpenMP parallelisation then operates as usual. That is, the OpenMP threads decompose the tree further into sub-trees for parallelisation.

A.6 Timestepping algorithm

When employing individual particle timesteps, we assign particles to bins numbered from zero, where zero would indicate a particle with $\Delta t = \Delta t_{\rm max}$, such that the bin identifier is

$$i_{{\rm bin},a} = \max\left\{{\rm int}\left[\log_2\left(\frac{2\Delta t_{\rm max}}{\Delta t_a}\right) - \epsilon_{\rm tiny}\right], 0\right\}, \qquad (A2)$$

where $\varepsilon_{\rm tiny}$ is a small number to prevent round-off error (equal to the `epsilon` function in Fortran). The timestep on which the particles move is simply

$$\Delta t = \frac{\Delta t_{\rm max}}{2^{n_{\rm max}}}, \qquad (A3)$$

where $n_{\rm max}$ is the maximum bin identifier over all the particles. Each timestep increments a counter, $i_{\rm stepfrac}$, which if the timestep hierarchy remains fixed simply counts up to $2^{n_{\rm max}}$. If $n_{\rm max}$ changes after each step, then $i_{\rm stepfrac}$ is adjusted accordingly ($i_{\rm stepfrac} = i_{\rm stepfrac}/2^{(n_{\rm max,old} - n_{\rm max,new})}$). A particle is active if

$$\mathrm{mod}\left[i_{\rm stepfrac}, 2^{(n_{\rm max} - i_{{\rm bin},a})}\right] = 0. \qquad (A4)$$

Active particles are moved onto a smaller timestep at any time (meaning any time that they are active and hence have their timesteps re-evaluated), but can only move onto a larger timestep if it is synchronised with the next-largest bin, determined by the condition

$$\mathrm{mod}\left[i_{\mathrm{stepfrac}}, 2^{(n_{\max}-i_{\mathrm{bin},a}+1)}\right] = 0. \qquad \text{(A5)}$$

In keeping with the above, particles are only allowed to move to a larger timestep by one bin at any given time.

We interpolate the positions of inactive particles by keeping all particles synchronised in time at the beginning and end of the timestep routine. This is achieved by storing an additional variable per particle, t_{was}. All particles begin the calculation with t_{was} set to half of their initial timestep, that is

$$t_{\mathrm{was},a} = \frac{1}{2}\left(\frac{\Delta t_{\max}}{i_{\mathrm{bin},a}}\right). \qquad \text{(A6)}$$

To be consistent with the RESPA algorithm (Section 2.3.3), we first update all particles to their half timestep. These velocities are then used to move the particle positions according to the inner loop of the RESPA algorithm [Equations (80)–(83)]. We then interpolate all velocities to the current time in order to evaluate the SPH derivative, finishing with the Leapfrog corrector step. Figure A8 shows the resulting pseudo-code for the entire timestepping procedure.

In particular, the first and last steps in the above (involving `twas`) interpolate the velocity to the current time. All other variables defined on the particles, including the thermal energy, magnetic field, and dust fraction, are timestepped following the velocity field.

To prevent scenarios were active particles quickly flow into an inactive region (e.g. the Sedov blast wave; see Section 5.1.3), inactive particles can be woken at any time. On each step, particles who should be woken up will be identified by comparing the `ibin(i)` of the active particles to `ibin(j)` of all i's neighbours, both active and inactive. If `ibin(i)+1 > ibin(j)`, then `j` will be woken up to ensure that its timestep is within a factor of two of its neighbours. At the end of the step, these particles will have their `ibin(j)` adjusted as required, and their `twas(j)` will be reset to the value of a particle with i_{bin} that has perpetually been evolved on that timestep. Finally, the predicted timestep will be replaced with `dt_av=dt_evolved(j)+0.5dt(j)`, where `dt_evolved(j)` is the time between the current time and the time the particle was last active, and `dt(j)` is the particle's new timestep.

A.7 Initial conditions: Monte Carlo particle placement

For setting up the surface density profile in discs, we use a Monte Carlo particle placement (Section 3.3.1). This is implemented as follows: We first choose the azimuthal angle as a uniform random deviate $u_1 \in [\phi_{\min}, \phi_{\max}]$ ($0 \to 2\pi$ by default). We then construct a power-law surface density profile $\Sigma \propto R^{-p}$ using the rejection method, choosing a sequence of random numbers $u_2 \in [0, f_{\max}]$ and iterating until we find a random number that satisfies

$$u_2 < f, \qquad \text{(A7)}$$

where $f \equiv R\Sigma = R^{1-p}$ and $f_{\max} = R_{\mathrm{in}}^{1-p}$ (or $f_{\max} = R_{\mathrm{out}}^{1-p}$ if $p \leqslant 1$). Finally, the z position is chosen with a third random number $u_3 \in [-z_{\max}, z_{\max}]$ such that $u_3 < g$, where $g = \exp[-z^2/(2H^2)]$ and $z_{\max} = \sqrt{6}H$.

```
init_step:
 t = 0
 do i=1,n
    twas(i) = 0.5*dt(i)
 enddo

step:
 dt_long = min(dt(1:n))
 do i=1,n
    v(i) = v(i) + (twas(i)-t)*asph(i)
 enddo
 t1 = t + dt_long
 do while (t < t1)
    t = t + dt_short
    do i=1,n
       v(i) = v(i) + 0.5*dt_short*aext(i)
       x(i) = x(i) + dt_short*v(i)
    enddo
    get_external_force(x,aext,dtshort_new)
    get_vdependent_external_force(x,v,aext)
    do i=1,n
       v(i) = v(i) + 0.5*dt_short*aext(i)
    enddo
    dt_short = min(dtshort_new, t1-t)
 enddo
 do i=1,n
    vstar(i) = v(i) + (t-twas(i))*asph(i)
    dtold(i) = dt(i)
 enddo
 get_sph_force(x,vstar,asph,dt)
 do i=1,n
    if (active)
       dt_av = 0.5*(dtold(i)+dt(i))
       v(i) = v(i) + dt_av*asph(i)
       twas(i) = twas(i) + dt_av
    endif
    v(i) = v(i) + (t - twas(i))*asph(i)
    wake_inactive_particles(ibin(i),twas(i),dt_av)
 enddo
```

Figure A8. Pseudo-code for the timestepping routine, showing how the interaction between individual timestepping and the RESPA algorithm is implemented. External forces and sink–gas interactions are computed on the fastest timescale $\Delta t_{\mathrm{short}} \equiv \Delta t_{\mathrm{ext}}$. Additional quantities defined on the particles follow the velocity terms. The variable t_{was} stores the last time the particle was active and is used to interpolate and synchronise the velocities at the beginning and end of each timestep.

A.8 Runtime parameters in PHANTOM in relation to this paper

Table A2 lists a dictionary of compile time and run time parameters used in the code in relation to the notation used in this paper.

A.9 The PHANTOM testsuite

Most numerical codes in astrophysics are tested entirely by their performance on physical problems with known solutions, with solutions that can be compared with other codes and by maintenance of various conservation properties at run time (see Section 5). We also *unit test* code modules. This allows issues to be identified at a much earlier stage in development. The tests are wrapped into the nightly testsuite. When a bug that escapes the testsuite has been

Table A2. Various runtime parameters in the code and their relation to this paper.

Quantity	Code variable	Description	Reference
h_{fact}	`hfact`	Ratio of smoothing length to particle spacing	2.1.2
$1/\Omega_a$	`gradh`	Smoothing length gradient correction term	2.1.2
ε_h	`tolh`	Tolerance in smoothing length–density iterations	2.1.4
F_{ab}	`grkern`	Scalar part of kernel gradient	2.1.5
$\partial W_{ab}(h)/\partial h_a$	`dwdh`	Derivative of kernel with respect to smoothing length	2.1.5
K	`polyk`	Polytropic constant used for barotropic equations of state	2.2.2
u_{time}	`utime`	Code time unit (cgs)	2.2.3
u_{mass}	`umass`	Code mass unit (cgs)	2.2.3
u_{dist}	`udist`	Code distance unit (cgs)	2.2.3
Equation (24)	`ipdv_heating`	Option to turn on/off PdV work term in energy equation	2.2.5
Λ_{shock}	`ishock_heating`	Option to turn on/off shock heating in energy equation	2.2.5
σ_{decay}	`avdecayconst`	Decay constant in artificial viscosity switch	2.2.9
ε_{v}	`tolv`	Tolerance on velocity error during timestepping	2.3.1
Δt_{max}	`dtmax`	Maximum time between output files	2.3.4
R_{acc}	`accradius1`	Accretion radius for central potential	2.4.1
M	`binarymassr`	Binary mass ratio for fixed binary potential	2.4.2
$R_{\mathrm{acc,\,2}}$	`accradius2`	Accretion radius for secondary in fixed binary potential	2.4.2
k_0	`RadiationPressure`	Radiation pressure in Poynting–Robertson drag	2.4.7
k_1	`Redshift`	Gravitational redshift in Poynting–Robertson drag	2.4.7
k_2	`TransverseDrag`	Transverse component of Poynting–Robertson drag	2.4.7
E_{m}	`st_energy`	Energy in turbulent stirring pattern	2.5
t_{decay}	`st_decay`	Decay time in turbulent stirring pattern	2.5
w	`st_solweight`	Solenoidal fraction in turbulent stirring pattern	2.5
k_{min}	`st_stirmin`	Minimum wavenumber in turbulent stirring pattern	2.5
k_{max}	`st_stirmax`	Minimum wavenumber in turbulent stirring pattern	2.5
f_{sol}	`st_solweightnorm`	Solenoidal weighting in turbulent stirring pattern	2.5
ε	`h_soft_sinksink`	Fixed gravitational softening length between sink particles	2.8.1
σ_{c}	`psidecayfac`	Dimensionless ratio of parabolic to hyperbolic $\nabla\cdot\boldsymbol{B}$ cleaning	2.10.8
f_{clean}	`overcleanfac`	Multiplier on maximum speed in divergence cleaning	2.10.9
ρ_{grain}	`graindenscgs`	Intrinsic dust density in cgs units	2.13.6
f_{d}	`damp`	Damping parameter for relaxing initial particle distributions	3.6

discovered, we have endeavoured to create a unit test to prevent a future recurrence.

A.9.1 Unit tests of derivative evaluations

The unit test of the density and force calculations checks that various derivatives evaluate to within some tolerance of the expected value. To achieve this, the test sets up 100 × 100 × 100 SPH particles in a periodic, unit cube, and specifies the input variables in terms of known functions. For example, the evaluation of the pressure gradient is checked by setting the thermal energy of each particle according to

$$u_a(x, y, z) = \frac{1}{2\pi}\left[3 + \sin\left(\frac{2\pi x}{L_x}\right) + \cos\left(\frac{2\pi y}{L_y}\right) + \sin\left(\frac{2\pi z}{L_z}\right)\right], \qquad \text{(A8)}$$

where L_x, L_y, and L_z are the length of the domain in each direction and the positions are relative to the edge of the box. We then compute the acceleration according to (34) with the artificial viscosity and other terms switched off, assuming an adiabatic EOS [Equation (25)] with $\rho =$ constant. We then test that numerical acceleration on all 10^6 particles is within some tolerance of the analytic pressure gradient expected from (341), namely

$$-\frac{\nabla P_a}{\rho_a}(x, y, z) = -(\gamma - 1)\nabla u_a(x, y, z), \qquad \text{(A9)}$$

where typically we use a tolerance of 10^{-3} in the relative error $E(x) = |x - x_{\mathrm{exact}}|/x_{\mathrm{exact}}$.

This procedure is repeated for the various derivatives of velocity, including the velocity divergence, curl, and all components of the strain tensor. We also test the artificial viscosity terms this way—checking that they translate correctly according to (119)—as well as the magnetic field derivatives and time derivatives, magnetic forces, artificial resistivity terms, physical viscosity terms, time derivatives of the dust fraction, and the time derivative of the velocity divergence required in the viscosity switch. We perform each of these tests also for the case where derivatives are evaluated on only a subset of the particles, as occurs when individual particle timesteps are employed.

We additionally check that various conservation properties are maintained. For example, energy conservation for hydrodynamics requires (37) to be satisfied. Hence, we include a test that checks that this summation is zero to machine precision. Similar tests are performed for magnetic fields, and for subsets of the forces that balance subsets of the thermal energy derivatives (e.g. the artificial viscosity terms).

A.9.2 Unit tests of sink particles

Unit tests for sink particles include (i) integrating a sink particle binary for 1 000 orbits and checking that this conserves total energy to a relative error of 10^{-6} and linear and angular momentum to machine precision; (ii) setting up a circumbinary disc of gas particles evolved for a few orbits to check that linear and angular momentum and energy are conserved; (iii) checking that circular orbits are correct in the presence of sink-sink softening; (iv) checking that accreting a gas particle onto a sink particle conserves linear and angular momentum, and that the resulting centre of mass position, velocity, and acceleration are set correctly (c.f. Section 2.8.2); and (v) checking that sink particle creation from a uniform sphere of gas particles (Section 2.8.4) succeeds and that the procedure conserves linear and angular momentum.

A.9.3 Unit tests of external forces

For external forces, we implement general tests that can be applied to any implemented external potential: (i) we check that the acceleration is the gradient of the potential by comparing a finite difference derivative of the potential, Φ, in each direction to the acceleration returned by the external force routine; and (ii) we check that the routines to solve matrix equations for velocity-dependent forces (e.g. Sections 2.4.5–2.4.7) agree with an iterative solution to the Leapfrog corrector step [Equation (69)].

A.9.4 Unit tests of neighbour finding routines

In order to unit test the neighbour finding modules (Section 2.1.7), we set up particles in a uniform random distribution with randomly assigned smoothing lengths. We then check that the neighbour list computed with the treecode agrees with a brute-force evaluation of actual neighbours. We also perform several sanity checks (i) that no dead or accreted particles appear in the neighbour list; (ii) that all particles can be reached by traversing the tree or link list structure; (iii) that nodes tagged as active contain at least one active particle and conversely that (iv) inactive cells contain only inactive particles; (v) that there is no double counting of neighbours in the neighbour lists; and (vi) that the cached and uncached neighbour lists are identical. We further check that particle neighbours are found correctly in pathological configurations, e.g. when all particles lie in a 1D line along each of the coordinate axes.

A.9.5 Unit tests of timestepping and periodic boundaries

As a simple unit test of both the timestepping and periodic boundaries, we set up 50 × 50 × 50 particles in a uniform periodic box with a constant velocity ($v_x = v_y = v_z = 1$) along the box diagonal. We then evolve this for 10 timesteps and check that the density on each particle remains constant and that the acceleration and other time derivatives remain zero to machine precision.

A.9.6 Unit tests of file read/write

We check that variables written to the header of the (binary) output files are successfully recovered by the corresponding read routine, and similarly for the particle arrays written to the body of the file. This quickly and easily picks up mistakes made in reading/writing variables from/to the output file.

A.9.7 Unit tests of kernel module

We ensure that calls to different kernel routines return the same answer, and check that gradients of the kernel and kernel softening functions returned by the routines are within some small tolerance of a finite difference evaluation of these gradients.

A.9.8 Unit tests of self-gravity routines

In order to unit test the treecode self-gravity computation (Section 2.12), we (i) check that the Taylor series expansion of the force on each leaf node matches the exact force for a particle placed close to the node centre; (ii) that the Taylor series expansion of the force and potential around the distant node are within a small ($\sim 10^{-5}$) tolerance of the exact values; (iii) that the combined expansion about both the local and distant nodes produces a force within a small tolerance of the exact value; and finally (iv) that the gravitational force computed on the tree for a uniform sphere of particles is within a small tolerance ($\sim 10^{-3}$) of the force computed by direct summation.

A.9.9 Unit tests of dust physics

We unit test the dust modules by first performing sanity checks of the dust–gas drag routine—namely that the transition between Stokes and Epstein drag is continuous and that the initialisation routine completes without error. We then perform a low-resolution DUSTYBOX test (Section 5.9.1), checking the solution matches the analytic solution as in Figure 46. For one-fluid dust, we perform a low-resolution version of the dust diffusion test (Section 5.9.3), checking against the solution at every timestep is within a small tolerance of the analytic solution. Dust mass, gas mass, and energy conservation in the one-fluid dust derivatives are also checked automatically.

A.9.10 Unit tests of non-ideal MHD

We perform three unit tests on non-ideal MHD. The first is the wave damping test (see Section 5.7.1). This test also uses the super-timestepping algorithm to verify the diffusive [Equation (202) but considering only Ohmic resistivity and ambipolar diffusion] and minimum stable timesteps [Equation (72)], but then evolves the system on the smallest of the two timesteps. The second test is the standing shock test (see Section 5.7.2). This test is also designed as a secondary check on boundary particles. Both tests used fixed coefficients for the non-ideal terms, and for speed, both tests are performed at much lower resolutions than presented in the paper. The third test self-consistently calculates the non-ideal coefficients for given gas properties.

A.9.11 Other unit tests

Various other unit tests are employed, including sanity checks of individual timestepping utility routines, checking that the barotropic EOS is continuous, of the Coriolis and centrifugal force routines, checks of conservation in the generalised Newtonian potential (Section 2.4.6), of the routine to revise the tree, and of the fast inverse square root routines.

A.9.12 Sedov unit test

As a final ‘real’ unit test, the code performs a low-resolution (16^3) version of the Sedov blast wave test (Section 5.1.3). We check that energy and momentum are conserved at a precision appropriate to the timestepping algorithm (for global timesteps, this means momentum conservation to machine precision and energy conservation to $\Delta E/E_0 < 5 \times 10^{-4}$; for individual timesteps, we ensure linear momentum conservation to 2×10^{-4} and energy conservation to 2×10^{-2}).

https://doi.org/10.1017/pasa.2018.25 Published online by Cambridge University Press