

PHANTOM: A smoothed particle hydrodynamics and magnetohydrodynamics code for astrophysics

Daniel J. Price^{1*}, James Wurster^{2,1}, Chris Nixon³, Terrence S. Tricco^{4,1}, Stéven Toupin⁵, Alex Pettitt⁶, Conrad Chan¹, Guillaume Laibe⁷, Simon Glover⁸, Clare Dobbs², Rebecca Nealon¹, David Liptai¹, Hauke Worpel^{9,1}, Clément Bonnerot¹⁰, Giovanni Dipierro¹¹, Enrico Ragusa¹¹, Christoph Federrath¹², Roberto Iaconi¹³, Thomas Reichardt¹³, Duncan Forgan¹⁴, Mark Hutchison¹, Thomas Constantino², Ben Ayliffe^{15,1}, Daniel Mentiplay¹, Kieran Hirsh¹ and Giuseppe Lodato¹¹

¹Monash Centre for Astrophysics (MoCA) and School of Physics and Astronomy, Monash University, Vic. 3800, Australia

²School of Physics, University of Exeter, Stocker Rd., Exeter EX4 4QL, UK

³Theoretical Astrophysics Group, Department of Physics & Astronomy, University of Leicester, Leicester LE1 7RH, UK

⁴Canadian Institute for Theoretical Astrophysics (CITA), University of Toronto, 60 St. George Street, Toronto, ON M5S 3H8, Canada

⁵Institut d’Astronomie et d’Astrophysique (IAA), Université Libre de Bruxelles (ULB), CP226, Boulevard du Triomphe B1050 Brussels, Belgium

⁶Department of Cosmosciences, Hokkaido University, Sapporo 060-0810, Japan

⁷Univ Lyon, ENS de Lyon, CNRS, Centre de Recherche Astrophysique de Lyon UMR5574, F-69230, Saint-Genis-Laval, France

⁸Zentrum für Astronomie der Universität Heidelberg, Institut für Theoretische Astrophysik, Albert-Ueberle-Str 2, D-69120 Heidelberg, Germany

⁹AIP Potsdam, An der Sternwarte 16, 14482 Potsdam, Germany

¹⁰Leiden Observatory, Leiden University, PO Box 9513, NL-2300 RA Leiden, the Netherlands

¹¹Dipartimento di Fisica, Università Degli Studi di Milano, Via Celoria 16, Milano, 20133, Italy

¹²Research School of Astronomy and Astrophysics, Australian National University, Canberra, ACT 2611, Australia

¹³Department of Physics and Astronomy, Macquarie University, Sydney, Australia

¹⁴St Andrews Centre for Exoplanet Science and School of Physics and Astronomy, University of St. Andrews, North Haugh, St. Andrews, Fife KY16 9SS, UK

¹⁵Met Office, FitzRoy Road, Exeter, EX1 3PB, UK

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, H₂ chemistry, dust-gas mixtures, physical viscosity, external forces including numerous galactic potentials as well as implementations of Lense-Thirring precession, Poynting-Robertson drag and stochastic turbulent driving. PHANTOM is hereby made publicly available.

Keywords: hydrodynamics — methods: numerical — magnetohydrodynamics (MHD) — accretion, accretion discs — ISM: general

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 time-scales 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, written in Fortran 90, developed over the last decade. It has been used widely for studies of accretion (Lodato & Price, 2010; Nixon et al., 2012a; Rosotti et al., 2012; Nixon, 2012; Nixon et al., 2012b; Facchini et al., 2013; Nixon et al., 2013; Martin et al., 2014a,b; Nixon & Lubow, 2015; Coughlin & Nixon, 2015; Forgan et al., 2017) and turbulence (Kitsionas et al., 2009; Price & Federrath, 2010; Price et al., 2011; Price, 2012b; Tricco et al., 2016b) as well as for studies of the Galaxy

*daniel.price@monash.edu

(Pettitt et al., 2014; Dobbs et al., 2016), for simulating dust-gas mixtures (Laibe & Price, 2012a,b; Dipierro et al., 2015, 2016; Ragusa et al., 2017), star cluster formation (Liptai et al., 2017) and non-ideal magnetohydrodynamics (Wurster et al., 2014, 2016, 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 focussed on cosmological applications, as in the highly successful GADGET code (Springel et al., 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 very 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). Many of these have since been implemented in the development version of GADGET (e.g. Iannuzzi & Dolag 2011) but remain unavailable or unused in the public version. Similarly, the SPHNG code (Benz et al., 1990; Bate, 1995; Bate et al., 1995; Whitehouse et al., 2005; Price & Bate, 2007) has been a workhorse for our group for simulations of star formation (Price & Bate, 2007, 2008, 2009; Tricco & Price, 2012; Price et al., 2012; Bate et al., 2014; Lewis et al., 2015) and accretion discs (e.g. Lodato & Rice, 2004; Cossins et al., 2009; Meru & Bate, 2010, 2011), 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. The GASOLINE code (Wadsley et al., 2004) is another that has been widely and successfully used for galaxy formation simulations, but is also not yet publicly available (though plans are afoot). Hubber et al. (2011) have developed SEREN with similar goals to PHANTOM, focussed 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 magnetohydrodynamics, for dust-gas mixtures, and for self-gravity as well as a range of other physics. The paper is also designed to serve as a handbook to guide those using the code in the correct use of the various algorithms. Stable releases of PHANTOM are posted on the web¹, while the development version and wiki documentation are available on the BITBUCKET platform².

2 Numerical method

PHANTOM is based on the Smoothed Particle Hydrodynamics (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 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, meaning that the fluid is discretised onto a set of ‘particles’ of mass m that are moved with the local fluid velocity \mathbf{v} . Hence the two basic equations common to all physics in PHANTOM are

$$\frac{d\mathbf{r}}{dt} = \mathbf{v}, \quad (1)$$

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

namely, the Lagrangian update of the particle position \mathbf{x} and the continuity equation expressing the conservation of mass (where ρ is the density).

¹<https://phantomsph.bitbucket.io/>

²<https://bitbucket.org/danielprice/phantom>

2.1.2 Conservation of mass in SPH

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

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

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

$$\frac{d\rho_a}{dt} = \frac{1}{\Omega_a} \sum_b m_b (\mathbf{v}_a - \mathbf{v}_b) \cdot \nabla_a W_{ab}(h_a), \quad (4)$$

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

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

The factor $1/\Omega_a$ is stored as `gradh` in the code. 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), (4) and (5) all depend on the kernel evaluated on neighbours within R_{kern} times h_a , all three of these summations are computed simultaneously using a single loop over the same set of neighbours. Details of the neighbour finding procedure are given in Section 2.1.8, below.

2.1.3 Setting the smoothing length

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

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

where h_{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.

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

$$\bar{N}_{\text{neigh}} = \frac{4}{3} \pi (R_{\text{kern}} h_{\text{fact}})^3, \quad (7)$$

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_{fact} (referred to as `hfact` in the code input file) is 1.2, corresponding to an average of 57.9 neighbours for a kernel truncated at $2h$ (i.e. for $R_{\text{kern}} = 2$) in three dimensions. Table 2 lists the settings recommended for different choices of kernel. The derivative required in (5) is given by

$$\frac{\partial h_a}{\partial \rho_a} = -\frac{3h_a}{\rho_a}. \quad (8)$$

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) and Price & Monaghan (2007), solving, for each particle, the equation

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

where ρ_{sum} is the density computed from (3) and

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

from (6). Equation (9) is solved with Newton-Raphson iterations,

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

where the derivative is given by

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

The iterations proceed until $|h_{a,\text{new}} - h_{a,0}|/h_{a,0} < \text{tolh}$, where $h_{a,0}$ is the smoothing length of particle a at the start of the iteration procedure and `tolh` is the tolerance. The convergence with Newton-Raphson is very 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. Iterations are further avoided in PHANTOM by predicting the smoothing length from the previous timestep according to

$$h_a^0 = h_a + \Delta t \frac{dh_a}{dt} = h_a + \Delta t \frac{\partial h_a}{\partial \rho_a} \frac{d\rho_a}{dt}, \quad (13)$$

where $d\rho_a/dt$ is evaluated from (4).

Since h and ρ are mutually dependent, we do not need to store both of them, and so in PHANTOM only the smoothing length is stored, from which the density can be obtained at any time via a function call evaluating $\rho(h)$. The default value of `tolh` is therefore 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

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

Table 1 Particle types in PHANTOM. The density and smoothing length of each type is computed only from neighbours of the same type (c.f. Section 2.14.3). Sink particles are handled separately in a different set of arrays.

neighbours which can be (and are in PHANTOM) efficiently cached. However, it is very important that the tolerance is a floating point number that can have arbitrary precision rather than an integer as implemented in 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). Failure to enforce this to sufficient precision may result in a loss of energy conservation.

2.1.5 Particle types

Particles can be assigned with a ‘type’ from the list given in Table 1. 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 2.13.3). 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 2.9).

2.1.6 Kernel functions

We write the kernel function in the form

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

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 $q \equiv |\mathbf{r}_a - \mathbf{r}_b|/h$. 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

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

and the derivative of the kernel with respect to h ,

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

The functions F_{ab} and $\partial W_{ab}(h)/\partial h_a$ are referred to as **grkern** and **dwdh** in the code. 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.

2.1.7 Choice of smoothing kernel

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

$$f(q) = \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3, & 0 \leq q < 1; \\ \frac{1}{4}(2 - q)^3, & 1 \leq q < 2; \\ 0, & q \geq 2, \end{cases} \quad (17)$$

where the normalisation constant $C_{\text{norm}} = 1/\pi$ in 3D and the compact support of the function implies that $R_{\text{kern}} = 2$. 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 factors³.

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)], this also increases the smoothing length itself, resulting in anti-convergence. 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,b; Børve et al., 2004; Price, 2012a; Dehnen & Aly, 2012). This is the motivation of our default choice

³See discussion in Price (2012a); in astrophysics exact preservation of conservation laws is always preferable to consistency of derivatives in SPH, as proposed in numerous ‘alternative’ particle methods.

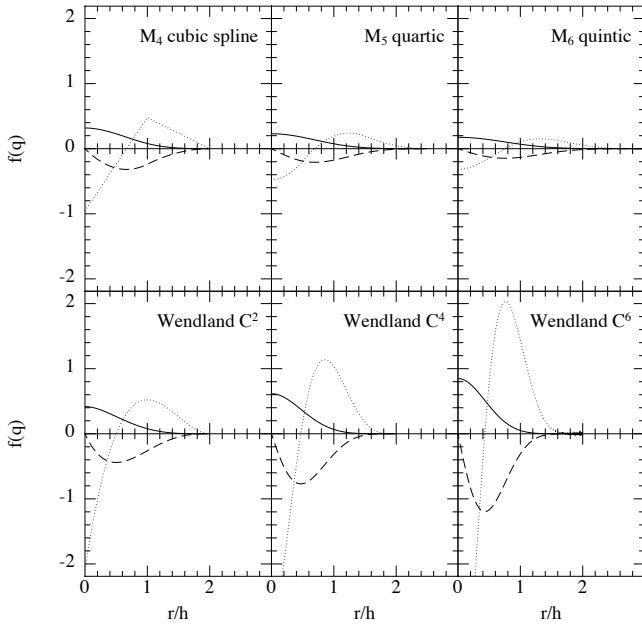


Figure 1. Smoothing kernels available in PHANTOM (solid lines) together with their first (dashed lines) and second (dotted lines) derivatives. Wendland kernels in PHANTOM (bottom row) are given compact support radii of 2, whereas the B-spline kernels (top row) adopt the traditional practice where the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

Kernel	R_{kern}	σ^2/h^2	σ/h	h_{fact}	h_{fact}^d	N_{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	≥ 1.35	1.4	92
C^4	2.0	8/13	0.78	≥ 1.55	1.6	137
C^6	2.0	1/2	0.71	≥ 1.7	2.2	356

Table 2 Compact support radii, variance, standard deviation, recommended ranges of h_{fact} and recommended default h_{fact} settings (h_{fact}^d) for the kernel functions available in PHANTOM

of $h_{\text{fact}} = 1.2$ in PHANTOM, since it is just short of the maximum neighbour number that can be used with the cubic spline while remaining stable to the pairing instability.

A better approach to reducing kernel bias is to keep the same resolution length⁴ but to use a kernel that

⁴This 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 the kernel function. Hence the Gaussian has the same resolution length as the M_6 quintic with compact support radius of $3h$ with $h_{\text{fact}} = 1.2$. Setting the number of neighbours, though related, is not a good way of specifying the resolution length.

has a larger compact support radius. The traditional approach (e.g. Morris, 1996a,b; Børve et al., 2004; Price, 2012a) has been to use the higher kernels in the B-spline series, i.e. the M_5 quartic which extends to $2.5h$

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

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

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

where $C_{\text{norm}} = 1/(120\pi)$ 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 kernels (Wendland, 1995), 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 three dimensional Wendland kernels scaled to a radius of $2h$ are given by C^2 ,

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

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

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

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

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

where $C_{\text{norm}} = 1365/(512\pi)$. Figure 1 shows the functional form 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. $h_{\text{fact}} = 2.2$ with $R_{\text{kern}} = 2.0$) 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 it is not always clear that this additional cost is worthwhile, and whether or not it depends on the application. A more comprehensive analysis would be very valuable here, as the ‘best’ choice of kernel remains an open question (see also the kernels proposed by Cabezón et al. 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 2.14; 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).

A simple and practical approach to demonstrating convergence 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 useful for preventing the pairing instability, particularly if one desires to employ a very large number of neighbours.

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 above 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`.

2.1.8 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 *kd*-tree. This was because PHANTOM was originally built 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 *kd*-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.

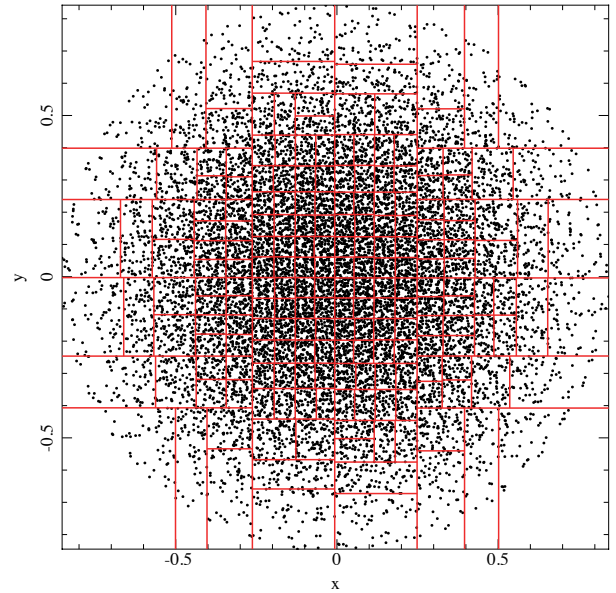


Figure 2. Example of the *kd*-tree build. For illustrative purposes only we have constructed a two dimensional version of the tree on the projected particle distribution in the *x-y* plane of the particle distribution from a polytrope test with 13,115 particles. Each level of the tree recursively splits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

```

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 3. 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.

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 into memory order and used to check for neighbours for individual particles in the group. The implementation of the *kd*-tree closely follows the description in Gafton & Rosswog (2011), splitting the particles recursively based on the centre of mass and bisecting the longest axis at each level (Fig-

ure 2). The tree build is refined until a cell contains less than N_{\min} particles, which is then referred to as a ‘leaf node’. $N_{\min} = 10$ by default. The neighbour search is then performed once for each leaf node.

The procedure for the tree build is given in Figure 3. We use a stack rather than recursive subroutines to aid parallelisation (the parallelisation strategy for the tree build is discussed further in Section 2.2.2). 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 (radius of a sphere containing all the particles centred on the centre of mass), the maximum smoothing length for all the particles 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 2.13). The `construct_node` subroutine also decides whether or not the node should be split (i.e. if the number of particles $> N_{\min}$) 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 8 quantities without self gravity, requiring 52 bytes per particle (`x,y,z,size,hmax`: 5 x 8-byte double precision; `leftchild, rightchild, parent`: 3 x 4-byte integer). With self-gravity we store 15 quantities (`mass` and `quads(6)`, i.e. 7 additional 8-byte doubles) requiring 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

```
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 4. Pseudo-code for the neighbour search (referred to as the `get_neigh` routine in Figure 5).

for problems with high dynamic range, such as star formation, where a small fraction of the particles collapse to very high density. Hence, we use this indexing scheme only up to a maximum level (`maxlevel_indexed`) which is set such that $2^{\text{maxlevel_indexed}}$ 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).

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 4. We loop over the nodes on the stack, checking the criterion

$$r_{nm}^2 < (s_n + s_m + R_{\text{kern}} h_{\text{max}}^n)^2 \quad (23)$$

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

```

!$omp parallel do
do node = 1, number of nodes
  if (node is leaf node)
    call get_neigh(node,listneigh,nneigh)
    i = firstinnode(node)
    do while (i > 0)
      do while not converged
        if (h > hmax(node)) call get_neigh
        do k=1,nneigh
          j = listneigh(k)
          if (n <= cache size)
            get j position from cache
          else
            get j position from memory
          endif
          if (actual neighbour)
            evaluate kernel and dwdh
            add to density sum
            add to gradh sum
            add to div v sum
          endif
        enddo
        update h
        check convergence
      enddo
      i = next(i)
    enddo
  endif
enddo
!$omp end parallel do

```

Figure 5. Pseudo-code for the density evaluation in PHANTOM, showing how Eqs. 3, 4 and 5 are computed. The force evaluation (evaluating Eqs. 35 and 37) 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 8).

length `nneigh` and a cache containing the trial neighbour positions (`xyzcache`) up to some maximum size (12000 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.

Once the neighbour list has been obtained for a given leaf node, we then 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_{\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 2.3.5). Figure 5 summarises the resulting procedure for calculating the density.

2.2 OpenMP parallelisation

2.2.1 Density and force calculation

Shared memory parallelisation of the density and force calculation is described in pseudo-code in Figure 5. 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 2.3.5) 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).

2.2.2 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 3, 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 2.1.8) where the levels of the tree are stored contiguously in memory. However, to avoid excessive memory consumption we only use this scheme while $2^{n_{\text{level}}} < n_{\text{part}}$. 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.

2.3 Hydrodynamics

2.3.1 Compressible hydrodynamics

The simplest ‘complete’ set of equations that can be used in PHANTOM are those describing compressible hydrodynamics. Here the velocity and internal energy are evolved according to

$$\frac{d\mathbf{v}}{dt} = -\frac{\nabla P}{\rho} + \Pi_{\text{shock}} + \mathbf{a}_{\text{ext}}(\mathbf{r}, t) + \mathbf{a}_{\text{sink-gas}} + \mathbf{a}_{\text{selfgrav}}, \quad (24)$$

$$\frac{du}{dt} = -\frac{P}{\rho} (\nabla \cdot \mathbf{v}) + \Lambda_{\text{shock}} - \Lambda_{\text{cool}}, \quad (25)$$

where P is the pressure, u is the specific internal energy, \mathbf{a}_{ext} , $\mathbf{a}_{\text{sink-gas}}$ and $\mathbf{a}_{\text{selfgrav}}$ refer to (optional) accelerations from ‘external’ or ‘body’ forces (Section 2.5), sink particles (Section 2.9) and self-gravity (Section 2.13), respectively. Π_{shock} and Λ_{shock} are dissipation terms required to give the correct entropy increase at a shock front, and Λ_{cool} is a cooling term.

2.3.2 Equation of state

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

$$P = (\gamma - 1)\rho u, \quad (26)$$

where γ is the adiabatic index and the sound speed c_s is given by

$$c_s = \sqrt{\frac{\gamma P}{\rho}}. \quad (27)$$

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

$$P = \frac{\rho k_B T}{\mu m_H}, \quad (28)$$

giving

$$T = \frac{\mu m_H}{k_B} (\gamma - 1) u, \quad (29)$$

where k_B is Boltzmann’s constant, μ is the mean molecular weight and m_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 (26) with $\gamma = 5/3$ is the default equation of state in PHANTOM, but the precise equation of state can be specified as a runtime option in the input file, while the implementation of different equations of state is straightforward as this is done in a self-contained module.

In the case where shocks are assumed to radiate away all of the heat generated at the shock front (i.e. $\Lambda_{\text{shock}} = 0$) and there is no cooling ($\Lambda_{\text{cool}} = 0$), Equation (25)

becomes simply, using (2)

$$\frac{du}{dt} = \frac{P}{\rho^2} \frac{d\rho}{dt}, \quad (30)$$

which, using (26) can be integrated to give

$$P = K\rho^\gamma, \quad (31)$$

where K is the polytropic constant (referred to as `polyk` in the code). Even more simply, in the case where the temperature is assumed constant, or prescribed as a function of position, the equation of state is simply

$$P = c_s^2 \rho. \quad (32)$$

In both of these cases (31) and (32) the internal energy does not need to be stored, and is not in PHANTOM when the compile-time option `ISOTHERMAL=yes`. In this case the temperature is effectively set by the value of `polyk` (and the density if $\gamma \neq 1$). Specifically,

$$T = \frac{\mu m_H}{k_B} K \rho^{\gamma-1}. \quad (33)$$

2.3.3 Code units

In the code we store the units of mass, length and time (`umass`, `udist`, and `utime`). For pure hydrodynamics these numbers are irrelevant to the numerical results since Equations (1)–(2) and (24)–(25) are scale free to all but the Mach number. Hence setting physical units is only useful when comparing to a physical system, when physical heating or cooling rates are applied via (25), or when one wishes to interpret the results in terms of temperature using (29) or (33).

In the case where gravitational forces are applied, either using an external force (Section 2.5) or using self-gravity (Section 2.13) we enforce the standard constraint on the units such that $G = 1$ in code units, i.e.

$$u_{\text{time}} = \sqrt{\frac{u_{\text{dist}}^3}{G u_{\text{mass}}}}. \quad (34)$$

Additional constraints apply when using relativistic terms (Section 2.5.5) or magnetic fields (Section 2.11.2). Units are specified during the particle setup.

2.3.4 Equations of motion in SPH

The SPH algorithm implemented in PHANTOM follows the variable smoothing length formulation described by Price (2012a), Price & Federrath (2010) and Lodato & Price (2010). We discretise (24) using

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \left[\frac{P_a + q_{ab}^a}{\rho_a^2 \Omega_a} \nabla_a W_{ab}(h_a) + \frac{P_b + q_{ab}^b}{\rho_b^2 \Omega_b} \nabla_a W_{ab}(h_b) \right] + \mathbf{a}_{\text{ext}}(\mathbf{x}_a, t) + \mathbf{a}_{\text{sink-gas}}^a + \mathbf{a}_{\text{selfgrav}}^a, \quad (35)$$

where the dissipation term Π_{shock} is encapsulated in the q_{ab}^a and q_{ab}^b terms, i.e.

$$\Pi_{\text{shock}}^a \equiv - \sum_b m_b \left[\frac{q_{ab}^a}{\rho_a^2 \Omega_a} \nabla_a W_{ab}(h_a) + \frac{q_{ab}^b}{\rho_b^2 \Omega_b} \nabla_a W_{ab}(h_b) \right]. \quad (36)$$

The shock capturing terms are discussed in detail in Section 2.3.7, below.

2.3.5 SPH in a single loop

A key difference to the density calculation (Section 2.1.2) is that computation of the acceleration (Equation 35) involves searching for neighbours within both $R_{\text{kern}} h_a$ and $R_{\text{kern}} h_b$. In the initial version of PHANTOM we avoided 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 ones 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 5. 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 2.1.8, 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 that averaging in the dissipation terms (above) is performed differently in PHANTOM, since at the end of the density loop one can immediately compute quantities that depend on ρ_a (i.e. P_a and q_{ab}^a) 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 this approach was the default in PHANTOM for a number of years, we have now abandoned it for a more traditional approach, where the density and force calculations are done in separate sub-routines and the kd -tree is used to search for neighbours checking both h_a and h_b for the force calculation. This ‘two-loop’ algorithm is now the default, while the one-loop algorithm is implemented on the `nogravity` branch.

2.3.6 Internal energy equation

The internal energy equation (25) is discretised using the time derivative of the density sum (c.f. 30) given by (4), giving

$$\frac{du_a}{dt} = \frac{P_a}{\rho_a^2 \Omega_a} \sum_b m_b \mathbf{v}_{ab} \cdot \nabla_a W_{ab}(h_a) + \Lambda_{\text{shock}} - \Lambda_{\text{cool}}. \quad (37)$$

Indeed, in the variational formulation of SPH (Gingold & Monaghan, 1982b; Monaghan & Price, 2001; Springel & Hernquist, 2002; Price & Monaghan, 2004b; Price, 2004, 2012a), this expression is used as a constraint to derive (35), 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. In the code the first two terms in (37) can optionally be turned off at runtime by setting the options `ipdv_heating` and `ishock_heating` to zero. These options can be used to give an approximation where each particle retains its initial temperature ($u = \text{const}$) or an approximation equivalent to a polytropic equation of state (31) if $\Lambda_{\text{shock}} = 0$.

2.3.7 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 magnetohydrodynamics (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 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). In the current version of PHANTOM, this is given by

$$q_{ab}^a = \begin{cases} -\frac{1}{2} \rho_a v_{\text{sig},a} \mathbf{v}_{ab} \cdot \hat{\mathbf{r}}_{ab}, & \mathbf{v}_{ab} \cdot \hat{\mathbf{r}}_{ab} < 0 \\ 0 & \text{otherwise} \end{cases} \quad (38)$$

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

$$v_{\text{sig},a} = \alpha_a^{\text{AV}} c_{s,a} + \beta^{\text{AV}} |\mathbf{v}_{ab} \cdot \hat{\mathbf{r}}_{ab}|, \quad (39)$$

where in general $\alpha_a^{\text{AV}} \in [0, 1]$ is controlled by a switch (see Section 2.3.9, below), while $\beta^{\text{AV}} = 2$ by default. Importantly α does *not* multiply the β^{AV} term. The β^{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 $\beta^{\text{AV}} \geq 2$ is needed wherever particle penetration may occur. This is very 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 $\beta^{\text{AV}} = 4$ 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 of the more coherent shocks.

2.3.8 Shock-capturing: internal energy equation

A key aspect from Chow & Monaghan (1997) is 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 the case of hydrodynamics and resistive dissipation in the case of MHD (see Section 2.11.5). The resulting contribution to the internal energy equation is given by (e.g. Price, 2012a)

$$\begin{aligned} \Lambda_{\text{shock}} \equiv & -\frac{1}{\Omega_a} \sum_b m_b v_{\text{sig},a} \frac{1}{2} (\mathbf{v}_{ab} \cdot \hat{\mathbf{r}}_{ab})^2 F_{ab}(h_a) \\ & + \sum_b m_b \alpha_u v_{\text{sig}}^u (u_a - u_b) \frac{1}{2} \left[\frac{F_{ab}(h_a)}{\Omega_a \rho_a} + \frac{F_{ab}(h_b)}{\Omega_b \rho_b} \right] \\ & + \Lambda_{\text{artres}}, \end{aligned} \quad (40)$$

where the first term provides the viscous shock heating, the second term provides a thermal conductivity and F_{ab} is defined as in Equation (15) and Λ_{artres} is the heating due to artificial resistivity (Equation 183). The signal speed in the conductivity term differs from that used for viscosity, as discussed by Price (2008) and Price (2012a). In PHANTOM we use

$$v_{\text{sig}}^u = \sqrt{\frac{|P_a - P_b|}{\bar{\rho}_{ab}}}, \quad (41)$$

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

$$v_{\text{sig}}^u = |\mathbf{v}_{ab} \cdot \hat{\mathbf{r}}_{ab}|, \quad (42)$$

for simulations that do (Wadsley et al., 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 4.1.4). With (42) we have found there is no need

for further switches to reduce conductivity (e.g. Price 2004; Price & Monaghan 2005), since the effective thermal conductivity κ is second order ($\propto h^2$). PHANTOM therefore uses $\alpha_u = 1$ by default in (40) and we have not yet found a situation where this leads to excess smoothing.

Combining (40), (37) and (35) it may be readily shown that the total energy is conserved, since

$$\frac{dE}{dt} = \sum_a m_a \left[\mathbf{v}_a \cdot \frac{d\mathbf{v}_a}{dt} + \frac{du_a}{dt} \right] = 0. \quad (43)$$

2.3.9 Shock detection

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

$$\frac{d\alpha_a}{dt} = \max(-\nabla \cdot \mathbf{v}_a, 0) - \frac{(\alpha_a - \alpha_{\text{min}})}{\tau_a}, \quad (44)$$

where $\tau \equiv \sigma_{\text{decay}} h / v_{\text{sig}}$ and $\sigma_{\text{decay}} = 0.1$ by default (σ_{decay} is referred to as `avdecayconst` in the input file). In the PHANTOM implementation, which has been the default option since 2007, we set v_{sig} in the decay time equal to the sound speed to avoid the need to store $d\alpha/dt$, since $\nabla \cdot \mathbf{v}$ 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; Price, 2012b; Tricco et al., 2016b).

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). In the current version of PHANTOM we implement the following switch (a slightly variation on Cullen & Dehnen 2010) with a shock indicator of the form

$$A_a = \xi_a \max \left[-\frac{d}{dt} (\nabla \cdot \mathbf{v}_a), 0 \right]. \quad (45)$$

where we use a modification of the Balsara (1995) form of the viscosity limiter for shear flows in the form

$$\xi = \frac{|\nabla \cdot \mathbf{v}|^2}{|\nabla \cdot \mathbf{v}|^2 + |\nabla \times \mathbf{v}|^2}. \quad (46)$$

We use this to set α according to

$$\alpha_{\text{loc},a} = \min \left(\frac{10h_a^2 A_a}{c_{s,a}^2}, \alpha_{\text{max}} \right), \quad (47)$$

where c_s is the sound speed and $\alpha_{\text{max}} = 1$. We use c_s in the expression for α_{loc} also for magnetohydrodynamics (Section 2.11) since we found using the magnetosonic speed led to a poor treatment of MHD shocks. If $\alpha_{\text{loc},a} > \alpha_a$ we set $\alpha_a = \alpha_{\text{loc},a}$, otherwise we evolve

α_a according to

$$\frac{d\alpha_a}{dt} = -\frac{(\alpha_a - \alpha_{\text{loc},a})}{\tau_a}, \quad (48)$$

where τ is defined as in the Morris & Monaghan 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.

$$\alpha_a^{n+1} = \frac{\alpha_a^n + \alpha_{\text{loc},a} \Delta t / \tau_a}{1 + \Delta t / \tau_a}, \quad (49)$$

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 $d(\nabla \cdot \mathbf{v}_a)/dt$. That is, we first compute the gradient tensors of the velocity, \mathbf{v} , and acceleration, \mathbf{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

$$R_a^{ij} \frac{\partial v_a^k}{\partial x_a^j} = \sum_b m_b (v_b^k - v_a^k) \frac{\partial W_{ab}}{\partial x_a^i}(h_a), \quad (50)$$

where

$$R_a^{ij} = \sum_b m_b (x_b^i - x_a^i) \frac{\partial W_{ab}}{\partial x_a^j}(h_a) \approx \delta^{ij}, \quad (51)$$

and repeated tensor indices imply summation. In the code, this means that during the density loop we compute the 9 independent terms in the summation on the right hand side of (50) for both velocity and acceleration, as well as the 6 independent components of (51). We then use these to solve the 3 x 3 matrix equation implied by (50) for each component of \mathbf{v} and \mathbf{a} . Finally, we construct the time derivative of the velocity divergence according to

$$\frac{d}{dt} \left(\frac{\partial v_a^i}{\partial x_a^i} \right) = \frac{\partial a_a^i}{\partial x_a^i} - \frac{\partial v_a^i}{\partial x_a^j} \frac{\partial v_a^j}{\partial x_a^i}, \quad (52)$$

where, as previously, repeated i and j indices imply summation. In cartesian coordinates the last term in (52) can be written out explicitly using

$$\begin{aligned} \frac{\partial v_a^i}{\partial x_a^j} \frac{\partial v_a^j}{\partial x_a^i} &= \left(\frac{\partial v^x}{\partial x} \right)^2 + \left(\frac{\partial v^y}{\partial y} \right)^2 + \left(\frac{\partial v^z}{\partial z} \right)^2 \\ &+ 2 \left[\frac{\partial v^x}{\partial y} \frac{\partial v^y}{\partial x} + \frac{\partial v^x}{\partial z} \frac{\partial v^z}{\partial x} + \frac{\partial v^z}{\partial y} \frac{\partial v^y}{\partial z} \right]. \end{aligned} \quad (53)$$

2.3.10 Cooling

The cooling term Λ_{cool} can be set either from detailed chemical calculations (Section 2.15.1) or, for discs, by the simple ‘ β -cooling’ prescription of Gammie (2001),

namely

$$\Lambda_{\text{cool}} = \frac{u}{t_{\text{cool}}}, \quad (54)$$

where

$$t_{\text{cool}} = \frac{\Omega(R)}{\beta_{\text{cool}}}, \quad (55)$$

with β_{cool} an input parameter to the code specifying the cooling timescale in terms of the local orbital time. We compute Ω in (55) 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.

2.3.11 Conserved quantities

It is readily shown (e.g. Price, 2012a) that using (35) the total linear momentum,

$$\sum_a m_a \mathbf{v}_a, \quad (56)$$

and total angular momentum,

$$\sum_a m_a \mathbf{r}_a \times \mathbf{v}_a, \quad (57)$$

are exactly conserved. In practice, this means they are conserved to the accuracy with which they are conserved by the timestepping scheme. In PHANTOM linear and angular momentum are conserved to round-off error with global timestepping, but exact conservation is violated when using individual particle timesteps or when using the treecode 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), are written to the `.ev` file output by the code and should be checked 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 stops if the errors in conservation exceed 10%.

2.4 Time integration

2.4.1 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

$$\mathbf{v}^{n+\frac{1}{2}} = \mathbf{v}^n + \frac{1}{2}\Delta t \mathbf{a}^n, \quad (58)$$

$$\mathbf{r}^{n+1} = \mathbf{r}^n + \Delta t \mathbf{v}^{n+\frac{1}{2}}, \quad (59)$$

$$\mathbf{a}^{n+1} = \mathbf{a}(\mathbf{r}^{n+1}), \quad (60)$$

$$\mathbf{v}^{n+1} = \mathbf{v}^{n+\frac{1}{2}} + \frac{1}{2}\Delta t \mathbf{a}^{n+1}, \quad (61)$$

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 et al., 2003), preserves the Hamiltonian nature of the SPH algorithm (e.g. Gingold & Monaghan, 1982b; Monaghan & Price, 2001; Monaghan, 2005; Price, 2012a). 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 (61) becomes implicit. The approach taken in PHANTOM 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

$$\mathbf{v}^{n+\frac{1}{2}} = \mathbf{v}^n + \frac{1}{2}\Delta t \mathbf{a}^n, \quad (62)$$

$$\mathbf{r}^{n+1} = \mathbf{r}^n + \Delta t \mathbf{v}^{n+\frac{1}{2}}, \quad (63)$$

$$\mathbf{v}^* = \mathbf{v}^{n+\frac{1}{2}} + \frac{1}{2}\Delta t \mathbf{a}^n, \quad (64)$$

$$\mathbf{a}^{n+1} = \mathbf{a}(\mathbf{r}^{n+1}, \mathbf{v}^*), \quad (65)$$

$$\mathbf{v}^{n+1} = \mathbf{v}^* + \frac{1}{2}\Delta t [\mathbf{a}^{n+1} - \mathbf{a}^n]. \quad (66)$$

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

$$e = \frac{|\mathbf{v}^{n+1} - \mathbf{v}^*|}{|\mathbf{v}^{\text{mag}}|} < \epsilon, \quad (67)$$

where \mathbf{v}^{mag} is the mean velocity on all SPH particles (we set the error to zero if $|\mathbf{v}^{\text{mag}}| = 0$) and by default $\epsilon = 10^{-2}$ (referred to as `tolv` in the input file). If this criterion is violated, then we recompute the accelerations, iterating (65) and (66) until the criterion is satisfied. In practice this happens rarely, but occurs for example in the first few steps of the Sedov problem where a strong discontinuity is placed in the initial conditions (Section 4.1.3). As each iteration is as expensive as halving the timestep, with global timestepping we also constrain the subsequent timestep such that itera-

tions should not occur, i.e.

$$\Delta t = \min \left(\Delta t, \frac{\Delta t}{\sqrt{e_{\text{max}}/\epsilon}} \right), \quad (68)$$

where $e_{\text{max}} = \max(e)$ over all particles. Additional variables such as the internal energy u and the magnetic field \mathbf{B} are timestepped with a predictor and trapezoidal corrector step in the same manner as the velocity, following (64) and (66).

Velocity-dependent external forces are treated separately, as described in Section 2.5, below.

2.4.2 Timestep constraints

The timestep itself is determined at the end of each step, and is constrained to be less than the Courant timestep (e.g. Lattanzio et al., 1986; Monaghan, 1997),

$$\Delta t_{\text{C}}^a \equiv C_{\text{cour}} \frac{h_a}{v_{\text{sig},a}^{\text{dt}}}, \quad (69)$$

where $C_{\text{cour}} = 0.3$ by default (Lattanzio et al., 1986) and $v_{\text{sig}}^{\text{dt}}$ is taken as the maximum of (39) over the particle's neighbours assuming $\alpha^{\text{AV}} = \max(\alpha^{\text{AV}}, 1)$. An additional constraint is applied from the accelerations (the ‘force condition’), where

$$\Delta t_{\text{f}}^a \equiv C_{\text{force}} \sqrt{\frac{h_a}{|\mathbf{a}_a|}}, \quad (70)$$

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

$$\Delta t_{\text{ext}}^a \equiv C_{\text{force}} \sqrt{\frac{h}{|\mathbf{a}_{\text{ext}}^a|}}, \quad (71)$$

and for accelerations to SPH particles to/from sink particles (Section 2.9, below)

$$\Delta t_{\text{sink-gas}}^a \equiv C_{\text{force}} \sqrt{\frac{h_a}{|\mathbf{a}_{\text{sink-gas}}^a|}}. \quad (72)$$

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

$$\Delta t_{\text{phi}}^a \equiv C_{\text{force}} \eta_{\Phi} \sqrt{\frac{|\Phi_a|}{|\nabla \Phi|_a^2}}. \quad (73)$$

where $\eta_{\Phi} = 0.05$ (see Section 2.9.5).

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

$$\Delta t_a = \min (\Delta t_{\text{C}}^a, \Delta t_{\text{f}}^a, \Delta t_{\text{ext}}^a, \Delta t_{\text{sink-gas}}^a, \Delta t_{\text{phi}}^a), \quad (74)$$

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

2.4.3 Substepping of external forces

In the case where the timestep is dominated by any of the external force timesteps (71)–(73) we implement an operator splitting approach implemented according to the reversible reference system propagator algorithm (RESPA) derived by Tuckerman et al. (1992) in the context of molecular dynamics. The RESPA integrator 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.

The implementation in PHANTOM 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

$$\mathbf{v} = \mathbf{v} + \frac{\Delta t_{\text{sph}}}{2} \mathbf{a}_{\text{sph}}^n, \quad (75)$$

$$\left\{ \begin{array}{l} \mathbf{v} = \mathbf{v} + \frac{\Delta t_{\text{ext}}}{2} \mathbf{a}_{\text{ext}}^m, \\ \mathbf{r} = \mathbf{r} + \Delta t_{\text{ext}} \mathbf{v}, \\ \mathbf{a}_{\text{ext}} = \mathbf{a}_{\text{ext}}(\mathbf{r}), \\ \mathbf{v} = \mathbf{v} + \frac{\Delta t_{\text{ext}}}{2} \mathbf{a}_{\text{ext}}^{m+1}, \end{array} \right. \quad (76)$$

$$(77)$$

$$(78)$$

$$(79)$$

$$\mathbf{a}_{\text{sph}} = \mathbf{a}_{\text{sph}}(\mathbf{r}), \quad (80)$$

$$\mathbf{v} = \mathbf{v} + \frac{\Delta t_{\text{sph}}}{2} \mathbf{a}_{\text{sph}}^n. \quad (81)$$

where \mathbf{a}_{SPH} indicates the SPH acceleration evaluated from (35) and \mathbf{a}_{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 (i.e. Eqs. 72, 71 and 73, respectively) while Δt_{sph} is the timestep relating to the SPH forces (i.e. Eqs. 69, 70 and 287). The number of substeps m is set equal to $m = \text{int}(\Delta t_{\text{ext},\text{min}}/\Delta t_{\text{SPH},\text{min}}) + 1$, where the minimum is taken over all particles.

2.4.4 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 block-stepped manner by assigning timesteps in factor-of-two decrements from some maximum timestep Δt_{max} (`dtmax` in the code), which we set as the time between output files.

The practical side of individual timestepping is reasonably straightforward. We assign particles to bins numbered from zero, where zero would indicate a particle with $\Delta t = \Delta t_{\text{max}}$, such that the bin identifier is

$$i_{\text{bin},a} = \max \left\{ \text{int} \left[\log_2 \left(\frac{2\Delta t_{\text{max}}}{\Delta t_a} \right) - \epsilon_{\text{tiny}} \right], 0 \right\}, \quad (82)$$

where ϵ_{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

$$\Delta t = \frac{\Delta t_{\text{max}}}{2^{n_{\text{max}}}}, \quad (83)$$

where n_{max} is the maximum bin identifier over all the particles. Each timestep increments a counter, i_{stepfrac} , which if the timestep hierarchy remains fixed simply counts up to $2^{n_{\text{max}}}$. If n_{max} changes after each step, then i_{stepfrac} is adjusted accordingly ($i_{\text{stepfrac}} = i_{\text{stepfrac}}/2^{(n_{\text{max,old}} - n_{\text{max,new}})}$). A particle is active if

$$\text{mod} \left[i_{\text{stepfrac}}, 2^{(n_{\text{max}} - i_{\text{bin},a})} \right] = 0. \quad (84)$$

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

$$\text{mod} \left[i_{\text{stepfrac}}, 2^{(n_{\text{max}} - i_{\text{bin},a} + 1)} \right] = 0. \quad (85)$$

In keeping with the above, particles are only allowed to move to a larger timestep by one bin at any given time. We implement a check similar to Saitoh & Makino (2009) condition where the timestep for a given particle is constrained to be within a factor of 2 of its neighbours.

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

$$t_{\text{was},a} = \frac{1}{2} \left(\frac{\Delta t_{\text{max}}}{i_{\text{bin},a}} \right). \quad (86)$$

To be consistent with the RESPA algorithm (Section 2.4.3) 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 76–79). We then interpolate all velocities to the current time in order to evaluate the SPH derivative, finishing with the leapfrog corrector step. Figure 6 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,

```

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)
  enddo

```

Figure 6. 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 $\Delta t_{\text{short}} \equiv \Delta t_{\text{ext}}$. 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.

magnetic field and dust fraction, are timestepped following the velocity field.

2.5 External forces

2.5.1 Point mass potential

The simplest external force describes a point mass at the origin,

$$\Phi_a = -\frac{GM}{r_a}; \quad \mathbf{a}_{\text{ext},a} = -\nabla\Phi_a = -\frac{GM}{|r_a|^3}\hat{r}_a, \quad (87)$$

where $r_a \equiv |r_a| \equiv \sqrt{r_a \cdot r_a}$. When this potential is used, we allow for particles within a certain radius from the origin to be accreted (`accradius1` in the input file). 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 written to the `.ev` file 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 2.9) where there are mutual gravitational forces between the star and the disc, and any mass accreted is added to the point mass (Section 2.9.2).

2.5.2 Binary potential

An external force module (`extern_binary`) also exists for modelling motion in binary systems where the mass of the disc is negligible. In this case the binary motion is prescribed using

$$\mathbf{r}_1 = [(1-M)\cos(t), (1-M)\sin(t), 0], \quad (88)$$

$$\mathbf{r}_2 = [-M\cos(t), -M\sin(t), 0], \quad (89)$$

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 code units, where Ω is the angular velocity of the binary. Thus only M needs to be specified in the input file (as `binarymassr`) to fix both m_1 and m_2 . When using this module the binary remains fixed on a circular orbit at $r = 1$ in code units. The binary potential is therefore

$$\Phi_a = -\frac{M}{|r_a - r_1|} - \frac{(1-M)}{|r_a - r_2|}, \quad (90)$$

such that the external acceleration is given by

$$\mathbf{a}_{\text{ext},a} = -\nabla\Phi_a = -M\frac{(\mathbf{r}_a - \mathbf{r}_1)}{|\mathbf{r}_a - \mathbf{r}_1|^3} - (1-M)\frac{(\mathbf{r}_a - \mathbf{r}_2)}{|\mathbf{r}_a - \mathbf{r}_2|^3}. \quad (91)$$

Again, there is an option to accrete particles that fall within a certain radius from either star (specified using `accradius1` and `accradius2` in the input file). 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 et al. 2013; Facchini et al. 2013; Martin et al. 2014a,b; Doğan et al. 2015; Ragusa

et al. 2016, 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 (91).

2.5.3 Binary potential with gravitational wave decay

As PHANTOM is written in a modular way, it is simple to replace particular algorithms with equivalent functionality by simply swapping the source files at compile time. A good example is the module implementing a binary with gravitational wave decay, replacing the binary potential module described above. This 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

$$\begin{aligned} \mathbf{r}_1 &= \left[-\frac{m_2}{m_1 + m_2} a \cos(\theta), -\frac{m_2}{m_1 + m_2} a \sin(\theta), 0 \right], \\ \mathbf{r}_2 &= \left[\frac{m_1}{m_1 + m_2} a \cos(\theta), \frac{m_1}{m_1 + m_2} a \sin(\theta), 0 \right], \end{aligned} \quad (92)$$

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

$$a = a_0 \left(1 - \frac{t}{\tau} \right)^{\frac{1}{4}}, \quad (93)$$

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

$$\tau \equiv \frac{5}{256} \frac{a_0^4}{\mu(m_1 + m_2)^2}. \quad (94)$$

The angle θ is defined using

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

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

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

The above can be used as a simple example of a time-dependent external potential.

2.5.4 Galactic potentials

PHANTOM implements a wide variety 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, implemented in the `extern_spiral` module.

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 et al., 1996), Begegan et al. (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 is 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.

2.5.5 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 we have used in a number of

publications (Nixon et al., 2012b; Nealon et al., 2015, 2016). In this case the external acceleration consists of a point mass potential and the Lense-Thirring term,

$$\mathbf{a}_{\text{ext}} = -\nabla\Phi + \mathbf{v} \times \boldsymbol{\Omega}_p, \quad (97)$$

where Φ is given by (87) and $\mathbf{v} \times \boldsymbol{\Omega}_p$ is the gravitomagnetic acceleration, with $\boldsymbol{\Omega}_p$ given by a dipole approximation

$$\boldsymbol{\Omega}_p \equiv \frac{2\mathbf{S}}{R^3} - \frac{6(\mathbf{S} \cdot \mathbf{r})\mathbf{r}}{R^5}, \quad (98)$$

with $\mathbf{S} = a_{\text{spin}}(GM)^2\mathbf{k}/c^3$, where \mathbf{k} is a unit vector in the direction of the black hole spin and $R_a = |\mathbf{r}_a|$. In the code we assume geometric units such that $G = M = c = 1$ when using the Lense-Thirring force (i.e., as in Section 2.3.3 but with two additional constraints on the units from the speed of light and the mass unit that fully specify the unit system).

Since in this case the external force depends on velocity, it cannot be implemented directly into leapfrog. The method used in PHANTOM 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). We first split the acceleration into position and velocity-dependent parts, i.e.

$$\mathbf{a}_{\text{ext}} = \mathbf{a}_{\text{ext},x}(\mathbf{r}) + \mathbf{a}_{\text{ext},v}(\mathbf{r}, \mathbf{v}). \quad (99)$$

The position dependent part (i.e. $-\nabla\Phi(\mathbf{r})$) is treated as usual. We add the velocity dependent part to the predictor step (63)–(64) as usual, but we write the corrector step (66) in the form

$$\mathbf{v}^{n+1} = \mathbf{v}^{n+\frac{1}{2}} + \frac{1}{2}\Delta t \left[\mathbf{a}_{\text{SPH+ext},x}^{n+1} + \mathbf{a}_{\text{ext},v}(\mathbf{r}^{n+1}, \mathbf{v}^{n+1}) \right]. \quad (100)$$

where $\mathbf{v}^{n+\frac{1}{2}} \equiv \mathbf{v}^n + \frac{1}{2}\Delta t\mathbf{a}^n$. This equation is implicit but the trick is to notice that it can be solved analytically for simple forces⁵. In the case of Lense-Thirring precession we have

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

where $\tilde{\mathbf{v}} \equiv \mathbf{v}^{n+\frac{1}{2}} + \frac{1}{2}\Delta t\mathbf{a}_{\text{SPH+ext},x}^{n+1}$. We therefore have a matrix equation in the form

$$\mathbf{R}\mathbf{v}^{n+1} = \tilde{\mathbf{v}}, \quad (102)$$

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

$$\mathbf{R} \equiv \begin{bmatrix} 1 & -\frac{\Delta t}{2}\Omega_p^z & \frac{\Delta t}{2}\Omega_p^y \\ \frac{\Delta t}{2}\Omega_p^z & 1 & -\frac{\Delta t}{2}\Omega_p^x \\ -\frac{\Delta t}{2}\Omega_p^y & \frac{\Delta t}{2}\Omega_p^x & 1 \end{bmatrix}. \quad (103)$$

⁵The 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 both symplectic and reversible unlike the methods proposed in those papers.

Rearranging (102) we obtain \mathbf{v}^{n+1} using

$$\mathbf{v}^{n+1} = \mathbf{R}^{-1}\tilde{\mathbf{v}}, \quad (104)$$

where \mathbf{R}^{-1} is the inverse of \mathbf{R} , which we compute using a subroutine that returns the analytic solution for a 3×3 matrix.

2.5.6 Generalised Newtonian potential

Finally, PHANTOM contains a module (`extern_gnewton`) implementing the generalised Newtonian potential described by Tejada & Ross-wog (2013) (see Bonnerot et al. 2016 for a recent application), where the acceleration terms are given by

$$\mathbf{a}_{\text{ext}} = -\frac{GM\mathbf{r}}{r^3}f^2 + \frac{2R_g\mathbf{v}(\mathbf{v} \cdot \mathbf{r})}{r^3f} - \frac{3R_g\mathbf{r}(\mathbf{v} \times \mathbf{r})^2}{r^5}, \quad (105)$$

where $R_g \equiv GM/c^3$ and $f \equiv (1 - 2R_g/r)$. This potential reproduces several features of the Schwarzschild (1916) spacetime, in particular reproducing the orbital and epicyclic frequencies to better than 7% (Tejada & Ross-wog, 2013). As the acceleration involves velocity-dependent terms it also requires a semi-implicit solution. Since the matrix equation is rather involved for this case, we iterate the corrector step using fixed point iterations until the velocity is within some tolerance of the previous value from the previous iteration.

2.5.7 Poynting-Robertson drag

The `extern_prdrag` module specifies the external force caused by a central point-like, gravitating, radiating, and non-rotating object. It is intended to be as general as possible. The acceleration of a particle subject to these external forces is

$$\mathbf{a}_{\text{ext}} = \frac{(k_0\beta_{\text{PR}} - 1)GM}{r^3}\mathbf{r} - \beta_{\text{PR}} \left(k_1 \frac{GM}{r^3} \frac{v_r}{c} \mathbf{r} - k_2 \frac{GM}{r^2} \frac{\mathbf{v}}{c} \right), \quad (106)$$

where \mathbf{r} and \mathbf{v} are position and velocity, v and r are their magnitudes, and 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. New modules should calculate β_{PR} from the physical characteristics of the SPH particles, and be able to read and write associated parameters to the input file. Relativistic effects are neglected, because these are thought to be less important than radiation forces for very low ($\beta_{\text{PR}} < 0.01$) luminosities even in accreting neutron star systems, where a very strong gravitational field is present (e.g., Miller & Lamb 1993).

The three terms on the right side of Equation (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 input file parameters k_0, k_1, k_2 , whose default values are unity—they are called `RadiationPressure`, `Redshift`, and `TransverseDrag` in the input file. Rotation of the central object can be crudely emulated by changing k_2 .

As for Lense-Thirring precession, the \mathbf{a}^{n+1} term of the leapfrog prescription can be expanded into velocity-dependent and non velocity-dependent component. We obtain, after some algebra

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

where

$$\mathbf{T} = \mathbf{v}^n + \frac{1}{2}\Delta t \mathbf{a}^n - \frac{(1 - k_0\beta_{\text{PR}})GM\Delta t}{2r^3}\mathbf{r}, \quad (108)$$

and

$$Q = \frac{GM\beta_{\text{PR}}\Delta t}{2cr^2}. \quad (109)$$

Equation (107) yields a set of simultaneous equations for the three vector components that can be solved analytically. Refer to [Worpel \(2015\)](#) for a detailed derivation.

2.5.8 Coriolis and centrifugal forces

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

$$\mathbf{a}_{\text{ext}} = -\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r}) - 2(\boldsymbol{\Omega} \times \mathbf{v}), \quad (110)$$

which are the centrifugal and coriolis terms, respectively, with $\boldsymbol{\Omega}$ the angular rotation vector. The implementation, in the `extern_corotate` module, 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.

2.6 Driven turbulence

PHANTOM contains a module (`forcing`) to drive turbulence 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 et al. \(2006\)](#); [Federrath et al. \(2008\)](#); [Schmidt et al. \(2009\)](#); [Federrath et al. \(2010a\)](#) and many subsequent works. This module was first used in PHANTOM in the supersonic turbulence comparison project performed by [Price & Federrath \(2010\)](#), with subsequent application to the density variance-Mach number relation in [Price et al. \(2011\)](#); to subsonic turbulence in [Price \(2012b\)](#); and most recently to supersonic, MHD turbulence in [Tricco et al. \(2016b\)](#).

The amplitude and phase of each Fourier mode is initialised by creating a set of 6 random numbers z_n drawn from a random Gaussian distribution with unit variance. We generate these 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

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

We thus set up 6 Gaussian random numbers

$$\mathbf{x}_n = \sigma \mathbf{z}_n, \quad (112)$$

where the standard deviation, σ , is set to the square root of the energy per mode divided by the correlation time, $\sigma = \sqrt{E_m/t_{\text{decay}}}$, where both E_m and t_{decay} are code input parameters (`st_energy` and `st_decay`, respectively).

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

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

where $f = \exp(-\Delta t/t_{\text{decay}})$ 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 ($P(f) = \text{const.}$) to ‘brown noise’ ($P(f) = 1/f^2$).

The amplitudes and phases of each mode are constructed by splitting \mathbf{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

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

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

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

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

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

where $k = \sqrt{k_x^2 + k_y^2 + k_z^2}$ is the wavenumber, with non-zero amplitudes defined only for wavenumbers where $k_{\text{min}} \leq k \leq k_{\text{max}}$, and a_{min} is the amplitude of the modes at k_{min} and k_{max} in the parabolic case (we use $a_{\text{min}} = 0$ in the code). The frequency resolution is defined with a uniform grid of 21 frequencies from $k_x = 0$ to $k_x = 2\pi/L$ in the x direction, with k_y and k_z defined with 9 equally spaced frequencies between 0 and

$2\pi/L$ and with four modes defined for each frequency combination, corresponding to $[k_x, k_y, k_z]$, $[k_x, -k_y, k_z]$, $[k_x, k_y, -k_z]$ and $[k_x, -k_y, -k_z]$. The default values for k_{\min} and k_{\max} (referred to as `st_stirmin` and `st_stirmax` in the code) 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 1000.

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

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

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

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

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_{sol} referred to as `st_solweightnorm` in the code 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. The pre-generated file corresponding to the sequences used in [Price & Federrath \(2010\)](#) and [Tricco et al. \(2016b\)](#) is distributed with the code and is used by default when the `STIR_FROM_FILE` preprocessor flag is set.

2.7 Disc viscosity

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

2.7.1 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 (36) and (38) represent a Navier-Stokes viscosity term with a fixed ratio between the bulk and shear viscosity terms ([Murray, 1996](#); [Español & Revenga, 2003](#); [Monaghan, 2005](#); [Jubelgas et al., 2004](#); [Lodato & Price, 2010](#); [Price, 2012b](#); [Meru & Bate, 2012](#)). In particular,

it can be shown (e.g. [Español & Revenga, 2003](#)) that

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

where $\bar{\rho}_{ab}$ is some appropriate average of the density. This enables the artificial viscosity term (36) 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 in v_{sig} is set equal to c_s
3. We adopt a constant α^{AV} , turning off shock detection switches (Section 2.3.9)
4. The viscosity term is multiplied by a factor $h/|r_{ab}|$

The net result is that (38) becomes

$$q_{ab}^a = \begin{cases} -\frac{\rho_a h_a}{2|r_{ab}|} (\alpha^{\text{AV}} c_{s,a} + \beta_{\text{AV}} |\mathbf{v}_{ab} \cdot \hat{\mathbf{r}}_{ab}|) \mathbf{v}_{ab} \cdot \hat{\mathbf{r}}_{ab}, & \mathbf{v}_{ab} \cdot \hat{\mathbf{r}}_{ab} < 0 \\ -\frac{\rho_a h_a}{2|r_{ab}|} \alpha^{\text{AV}} c_{s,a} \mathbf{v}_{ab} \cdot \hat{\mathbf{r}}_{ab}, & \text{otherwise} \end{cases} \quad (120)$$

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](#))

$$\nu_{\text{AV}} \approx \frac{1}{10} \alpha^{\text{AV}} c_s h, \quad (121)$$

$$\zeta_{\text{AV}} = \frac{5}{3} \nu_{\text{AV}} \approx \frac{1}{6} \alpha^{\text{AV}} c_s h. \quad (122)$$

The Shakura-Sunyaev prescription is

$$\nu = \alpha_{\text{SS}} c_s H, \quad (123)$$

where H is the scale height. This implies that α_{SS} may be determined from α_{AV} using

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

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

In practice this means that to obtain a constant α_{SS} in the disc one must uniformly resolve the scale height. We have achieved this in simulations to date by choosing the surface density and temperature profiles of the disc 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 β^{AV} to zero, but this is dangerous if the disc dynamics are complex as there is nothing to prevent particle penetration (see Section 2.3.7). Hence in the current version of the code $\beta^{\text{AV}} = 2$ 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 β^{AV} term is safer with no noticeable effect on the overall dissipation because of the second order dependence of this term on resolution.

Using α^{AV} to set the disc viscosity has two useful consequences: First, it forces one to consider whether or not the scale height is resolved. Second, knowing the value of α^{AV} is helpful: $\alpha^{\text{AV}} \approx 0.1$ 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 α^{AV} ; see Bate 1995; Meru & Bate 2012), while $\alpha^{\text{AV}} > 1$ constrains the timestep (Section 2.4.2).

2.7.2 Disc viscosity using the Navier-Stokes viscosity terms

An alternative approach is to compute viscous terms directly from the Navier-Stokes equation (Section 2.8). The details of how the Navier-Stokes terms are represented are given below (Section 2.8), but for disc viscosity we need a method for determining the kinematic viscosity, which in turn requires specifying the scale height as a function of radius. We use

$$H_a \equiv \frac{c_s^a}{\Omega(R_a)}, \quad (125)$$

where we assume Keplerian rotation $\Omega = \sqrt{GM/R^3}$ and c_s is obtained for a given particle from the equation of state (which for consistency must be either isothermal or radially isothermal). *This restricts the application of this approach to discs where R can be meaningfully defined, excluding for example discs around binary stars.* The shear viscosity is then set using

$$\nu_a = \alpha_{\text{SS}} c_s^a H_a, \quad (126)$$

where α_{SS} is a parameter in the input file. The timestep is constrained using $C_{\text{visc}} h^2 / \nu$ as described in Section 2.8. 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 means that this is usually not problematic (Section 2.3.9). This formulation of viscosity was used in Facchini et al. (2013).

2.8 Physical viscosity

Physical viscosity is implemented as described in Lodato & Price (2010). Here (24) and (25) are replaced

by the compressible Navier-Stokes equations, i.e.

$$\frac{dv^i}{dt} = -\frac{1}{\rho} \frac{\partial S_{\text{NS}}^{ij}}{\partial x^j} + \Pi_{\text{shock}} + \mathbf{a}_{\text{ext}}(\mathbf{r}, t) + \mathbf{a}_{\text{sink-gas}} + \mathbf{a}_{\text{selfgrav}}, \quad (127)$$

$$\frac{du}{dt} = -\frac{P}{\rho} (\nabla \cdot \mathbf{v}) + \Lambda_{\text{visc}} + \Lambda_{\text{shock}} - \Lambda_{\text{cool}}, \quad (128)$$

with the stress tensor given by

$$S_{\text{NS}}^{ij} = \left[P - \left(\zeta - \frac{2}{3} \eta \right) \frac{\partial v^k}{\partial x^k} \right] \delta^{ij} - \eta \left(\frac{\partial v^i}{\partial x^j} + \frac{\partial v^j}{\partial x^i} \right) \quad (129)$$

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

2.8.1 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) and Watkins et al. (1996). Here (127) is discretised in the standard manner using

$$\frac{dv_a^i}{dt} = -\sum_b m_b \left[\frac{S_{\text{NS},a}^{ij}}{\Omega_a \rho_a^2} \frac{\partial W_{ab}(h_a)}{\partial x_a^j} + \frac{S_{\text{NS},b}^{ij}}{\Omega_b \rho_b^2} \frac{\partial W_{ab}(h_b)}{\partial x_a^j} \right] + \Pi_{\text{shock}}^i + a_{\text{ext}}^i(\mathbf{r}, t) + a_{\text{sink-gas}}^i + a_{\text{selfgrav}}^i, \quad (130)$$

where the velocity gradients are computed during the density loop using

$$\frac{\partial v_a^i}{\partial x_a^j} = -\frac{1}{\Omega_a \rho_a} \sum_b m_b v_{ab}^i \nabla_a^j W_{ab}(h_a). \quad (131)$$

Importantly, the differenced SPH operator is used in (131) whereas (130) uses the symmetric gradient operator. The use of conjugate operators⁶ 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

$$\frac{dE}{dt} = \sum_a m_a \left(\frac{du_a}{dt} + v_a^i \frac{dv_a^i}{dt} \right) = 0. \quad (132)$$

This implies a contribution to the thermal energy equation given by

$$\frac{du_a}{dt} = \frac{S_{\text{NS},a}^{ij}}{\Omega_a \rho_a^2} \sum_b m_b v_{ab}^i \nabla_a^j W_{ab}(h_a), \quad (133)$$

⁶The SPH difference operator is discretely skew-adjoint to the symmetric operator (Cummins & Rudman, 1999).

which can be seen to reduce to (25) in the inviscid case ($S_{\text{NS}}^{ij} = P\delta^{ij}$) but in general is an SPH expression for

$$\frac{du_a}{dt} = -\frac{S_{\text{NS},a}^{ij}}{\rho_a} \frac{\partial v_a^i}{\partial x_a^j}. \quad (134)$$

Using $S_{\text{NS}}^{ij} = S_{\text{NS}}^{ji}$ we have

$$\frac{du_a}{dt} = -\frac{1}{2} \frac{S_{\text{NS},a}^{ij}}{\rho_a} \left(\frac{\partial v_a^i}{\partial x_a^j} + \frac{\partial v_a^j}{\partial x_a^i} \right), \quad (135)$$

which, using (129), gives

$$\Lambda_{\text{visc}} = \left(\zeta_{v,a} - \frac{2}{3} \nu_a \right) (\nabla \cdot \mathbf{v})_a^2 + \frac{\nu_a}{2} \left(\frac{\partial v_a^i}{\partial x_a^j} + \frac{\partial v_a^j}{\partial x_a^i} \right)^2. \quad (136)$$

where by the square in the last term we mean the tensor summation $\sum_j \sum_i T_{ij} T_{ij}$, 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. 6 additional quantities when taking account of symmetries. This approach to physical viscosity is therefore enabled in the code by a preprocessor flag (`-DUSE_STRAIN_TENSOR`).

2.8.2 Physical viscosity with direct second derivatives

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

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

$$\begin{aligned} \nabla \cdot (C\nabla \mathbf{v}) &\approx \\ & - \sum_b m_b \left[\frac{C_a}{\rho_a} G_{ab}(h_a) + \frac{C_b}{\rho_b} G_{ab}(h_b) \right] \mathbf{v}_{ab}, \end{aligned} \quad (138)$$

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

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

$$C \equiv \frac{1}{2} \eta, \quad (140)$$

so that we can simply use

$$\begin{aligned} \left(\frac{dv_a^i}{dt} \right)_{\text{visc}} &= \sum_b \frac{m_b}{\rho_{ab}} (\tau_a + \tau_b) (\mathbf{v}_{ab} \cdot \hat{\mathbf{r}}_{ab}) \hat{\mathbf{r}}_{ab}^i G_{ab} \\ &+ \sum_b \frac{m_b}{\rho_{ab}} (\kappa_a + \kappa_b) v_{ab}^i G_{ab}, \end{aligned} \quad (141)$$

where

$$\tau = \frac{5}{2} A, \quad (142)$$

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

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

$$\begin{aligned} \Lambda_{\text{visc}} &= \frac{\tau_a}{\rho_a} \sum_b m_b (\mathbf{v}_{ab} \cdot \hat{\mathbf{r}}_{ab})^2 G_{ab}(h_a) \\ &+ \frac{\kappa_a}{\rho_a} \sum_b m_b (\mathbf{v}_{ab})^2 G_{ab}(h_a). \end{aligned} \quad (144)$$

Because this formulation does not require additional storage, it is the default formulation of physical viscosity in the code. 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).

2.8.3 Timestep constraint

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

$$\Delta t_{\text{visc}}^a \equiv C_{\text{visc}} \frac{h_a^2}{\nu_a}, \quad (145)$$

where $C_{\text{visc}} = 0.25$ by default (Brookshaw, 1994). When physical viscosity is turned on, this constraint is included with other timestep constraints according to Equation (74).

2.8.4 Physical viscosity and the tensile instability

Some caution is required in the use of physical viscosity directly, particularly at high Mach number, since negative stress can lead to the tensile instability (Morris, 1996b; Monaghan, 2000; Gray et al., 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.

2.8.5 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 e.g. Bonet & Lok 1999 or section 5 of Price & Monaghan 2004b). Recently, Hu & Adams (2006) and Müller et al. (2015) have proposed physical viscosity formulations that explicitly conserve angular momentum, but these are not yet implemented in PHANTOM and would require careful testing before being applied in astrophysics.

2.9 Sink particles

Sink particles were introduced into SPH by Bate et al. (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 sinks, 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 collisional (unsoftened) direct N^2 summation; 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 RESPA algorithm as described in Section 2.4.3, which is second order accurate but allows sink particles to evolve on shorter timesteps compared to SPH particles.

2.9.1 Sink particle accelerations

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

$$\begin{aligned} \frac{d\mathbf{v}_i}{dt} = & - \sum_{j=1}^{N_{\text{sink}}} GM_j \phi'_{ij}(\epsilon) \hat{\mathbf{r}}_{ij} \\ & - \sum_{b=1}^{N_{\text{part}}} Gm_b \phi'_{ib}(\epsilon_{ib}) \hat{\mathbf{r}}_{ib}, \end{aligned} \quad (146)$$

where ϕ'_{ab} is the usual softening kernel (Section 2.13.2), with the sink-gas softening length ϵ_{ib} defined as the maximum of the (fixed) softening length defined for the sink particles (ϵ is `h_soft_sinksink` in the input file) and the softening length of the gas particle, i.e. $\epsilon_{ib} \equiv \max(\epsilon, \epsilon_b)$. SPH particles receive a corresponding acceleration

$$\mathbf{a}_{\text{sink-gas}}^a = - \sum_{j=1}^{N_{\text{sink}}} GM_j \phi'_{aj}(\epsilon_{aj}) \hat{\mathbf{r}}_{aj}. \quad (147)$$

Softening of sink-gas interactions is not applied if the softening length of the sink is set to zero, in which case

the sink gas accelerations reduce simply to

$$\mathbf{a}_{\text{sink-gas}}^a = - \sum_{j=1}^{N_{\text{sink}}} \frac{GM_j}{|\mathbf{r}_a - \mathbf{r}_j|^3} \mathbf{r}_{aj}. \quad (148)$$

This is the default behaviour when sink particles are used in the code, as softening of sink-gas interactions is mainly useful when one wishes to model a point mass particle that does not accrete gas (e.g. by setting the accretion radius to zero). The sink-sink interaction is unsoftened by default ($\epsilon = 0$), giving the usual

$$\mathbf{a}_{\text{sink-sink}}^i = - \sum_{j=1}^{N_{\text{sink}}} \frac{GM_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} \mathbf{r}_{ij}. \quad (149)$$

Caution is required when integrating tight binary or multiple systems when $\epsilon = 0$ to ensure that the timestep conditions (Section 2.4.2) are strict enough.

2.9.2 Accretion onto sink particles

When a gas particle falls within the accretion radius r_{acc} of a sink (set in the initial conditions or when the sink particle is created; see Section 2.9.4), we perform a series of checks to determine if it will be accreted. First, if the particle falls within $f_{\text{acc}} r_{\text{acc}}$, where $0 \leq f_{\text{acc}} \leq 1$, it is indiscriminately accreted without undergoing any checks; the default is $f_{\text{acc}} = 0.5$. In the boundary region $f_{\text{acc}} r_{\text{acc}} < r < r_{\text{acc}}$ accretion occurs if

1. $|\mathbf{L}_{ai}| < |\mathbf{L}_{\text{acc}}|$; the specific angular momentum of the particle is less than that of a Keplerian orbit at r_{acc} .
2. $e = \frac{v_{ai}^2}{2} - \frac{GM_i}{r_{ai}} < 0$; the particle is bound.
3. e for this gas-sink pair is smaller than e for this gas particle and any other sink particle; the gas particle is most bound to this sink particle.

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

$$\begin{aligned} |\mathbf{L}_{ai}^2| & \equiv |\mathbf{r}_{ai} \times \mathbf{v}_{ai}|^2 \\ & = r_{ai}^2 v_{ai}^2 - (\mathbf{r}_{ai} \cdot \mathbf{v}_{ai})^2, \end{aligned} \quad (150)$$

while $|\mathbf{L}_{\text{acc}}| = r_{\text{acc}}^2 \Omega_{\text{acc}}$ is the angular momentum at r_{acc} , where $\Omega_{\text{acc}} = \sqrt{GM_i/r_{ai}^3}$ is the Keplerian angular speed at r_{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 (here labelled a) passes the accretion checks then the mass, position, velocity, acceleration and spin angular momentum of the sink are updated

as follows

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

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

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

$$\mathbf{S}_i = \mathbf{S}_i + \frac{m_a M_i}{M_i + m_a} [(\mathbf{r}_a - \mathbf{r}_i) \times (\mathbf{v}_a - \mathbf{v}_i)], \quad (154)$$

$$M_i = M_i + m_a. \quad (155)$$

This ensures that mass, linear momentum and angular momentum (but not energy) are conserved by the accretion process. We store the accreted mass as well as the total mass for each sink to avoid problems with round-off error in the case where the particle masses are much smaller than the sink mass. An accreted particle is tagged in the code by having its smoothing length set to negative. Those particles with $h \leq 0$ are subsequently excluded when the tree is built.

2.9.3 Sink particle boundary conditions

At present no special sink particle boundary conditions are implemented in PHANTOM. 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 et al. \(2013b\)](#). 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, private communication 2014) and often it is more cost effective to simply reduce the accretion radius of the sink. This may change in future code versions.

2.9.4 Dynamic sink particle creation

As described in [Bate et al. \(1995\)](#) it is also possible to create sink particles on-the-fly in PHANTOM provided certain conditions are met, provided self-gravity is turned on (Section 2.13) and the `icreate_sinks` flag is set greater than zero in the input file. The conditions required to test for sink 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_{crit} from an existing sink. Once these conditions are met on a particular particle, a , we test for creation of a sink by checking the following conditions ([Bate et al., 1995](#))

1. The particle is a gas particle.
2. $\nabla \cdot \mathbf{v}_a \leq 0$: Gas surrounding the particle is at rest or collapsing.

3. $h_a < r_{\text{acc}}/2$: The smoothing length is less than half of the accretion radius.
4. All neighbours within r_{acc} are currently active.
5. The ratio of thermal to gravitational energy of particles within r_{acc} , α_J , satisfies $\alpha_J \leq 1/2$.
6. $\alpha_J + \beta_{\text{rot}} \leq 1$; where β_{rot} is the ratio of rotational energy to the magnitude of the gravitational energy for particles within r_{acc} , i.e.

$$\beta_{\text{rot}} \equiv \frac{|e_{\text{rot}}|}{|e_{\text{grav}}|}. \quad (156)$$

7. $e_{\text{tot}} < 0$: The total energy of particles within r_{acc} is negative (i.e. the clump is bound).

If these checks pass, a sink is created at the position of particle a and we immediately accrete particles within r_{acc} by calling the routine described in Section 2.9.2. The checks above are similar to those in [Federrath et al. \(2010b\)](#), except that [Federrath et al. \(2010b\)](#) introduced an additional check, such 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

$$e_{\text{kin}} = \frac{1}{2} \sum_{b=1}^{N < r_{\text{acc}}} m_b (\mathbf{v}_b - \mathbf{v}_a)^2, \quad (157)$$

$$e_{\text{therm}} = \sum_{b=1}^{N < r_{\text{acc}}} m_b u_b, \quad (158)$$

$$e_{\text{grav}} = -\frac{1}{2} \sum_{b=1}^{N < r_{\text{acc}}} \sum_{c=b}^{N < r_{\text{acc}}} G m_b m_c \times [\phi(|\mathbf{r}_b - \mathbf{r}_c|, h_b) + \phi(|\mathbf{r}_b - \mathbf{r}_c|, h_c)], \quad (159)$$

$$e_{\text{tot}} = e_{\text{kin}} + e_{\text{therm}} + e_{\text{grav}}, \quad (160)$$

$$e_{\text{rot}} \equiv \sqrt{e_{\text{rot},x}^2 + e_{\text{rot},y}^2 + e_{\text{rot},z}^2}, \quad (161)$$

$$e_{\text{rot},x} \equiv \frac{1}{2} \sum_{b=1}^{N < r_{\text{acc}}} m_b \frac{L_{ab,x}^2}{\sqrt{(y_a - y_b)^2 + (z_a - z_b)^2}}, \quad (162)$$

$$e_{\text{rot},y} \equiv \frac{1}{2} \sum_{b=1}^{N < r_{\text{acc}}} m_b \frac{L_{ab,y}^2}{\sqrt{(x_a - x_b)^2 + (z_a - z_b)^2}}, \quad (163)$$

$$e_{\text{rot},z} \equiv \frac{1}{2} \sum_{b=1}^{N < r_{\text{acc}}} m_b \frac{L_{ab,z}^2}{\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}}, \quad (164)$$

where $\mathbf{L}_{ab} \equiv (\mathbf{r}_a - \mathbf{r}_b) \times (\mathbf{v}_a - \mathbf{v}_b)$ is the specific angular momentum between a pair of particles, and ϕ is the gravitational softening kernel (defined in Section 2.13), which has units of inverse length; to obtain the total potential of the cloud, adding the contribution from *all* pairs, $b - c$, within the cloud is required.

2.9.5 Sink particle timesteps

Sink particles are integrated together with a global but adaptive timestep, following the inner loop of the RESPA algorithm (76)–(79), corresponding to a second-order leapfrog integration. The timestep is controlled by the minimum of the sink-gas timestep (Equation 72) and (Dehnen & Read, 2011)

$$\Delta t_{\text{sink-sink}} \equiv \min_j \left(C_{\text{force}} \eta_{\Phi} \sqrt{\frac{|\Phi_a^{\text{sink-sink}}|}{|\nabla \Phi_a^{\text{sink-sink}}|^2}} \right). \quad (165)$$

where the potential and gradient include other sink particles plus any external potentials applied to sink particles, but not 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 $C_{\text{force}} = 0.25$; see Section 4.5.1.

More accurate integrators such as the fourth-order Hermite scheme (Makino & Aarseth, 1992) or the fourth order symplectic schemes proposed by Omelyan et al. (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 4th order Hermite scheme for sink particles in SPH).

2.10 Stellar physics

A tabulated equation of state (EOS) can be used to take account of the departure from an ideal gas, for example due to changes in ionization or molecular dissociation and recombination. This tabulated EOS in PHANTOM is adapted from the $\log P_{\text{gas}} - T$ 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 et al. (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 the PHANTOM implementation (adapted from original routines by Thomas Constantino written for the MUSIC code; see Goffrey et al. 2016), we compute the EOS for a particular composition using bilinear 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 then 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 $10.5 \leq \log u \leq 17.5$ and $0.0 \leq \log V \leq 14.0$.

2.11 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,b, 2005). PHANTOM was also used for testing of the vector potential formulation (Price, 2010) but this implementation has been deleted from the current code due to its poor performance (see Price 2010). The recent improvements by Tricco & Price (2012, 2013) and Tricco et al. (2016a) are the main differences compared to the GADGET implementation of SPMHD (Dolag & Stasyszyn, 2009; Bürzle et al., 2011a,b) which also implements the core algorithms from Price & Monaghan (2004a,b, 2005). These improvements are vital for preserving the divergence-free (no monopoles) condition on the magnetic field. For recent applications of PHANTOM to MHD problems, see Wurster et al. (2016, 2017), Tricco et al. (2016b), Dobbs et al. (2016), Bonnerot et al. (2017) and Forgan et al. (2017).

2.11.1 Continuum equations

PHANTOM solves the equations of magnetohydrodynamics in the form

$$\frac{dv^i}{dt} = -\frac{1}{\rho} \frac{\partial M^{ij}}{\partial x^j} + \Pi_{\text{shock}} + f_{\text{divB}}^i + a_{\text{ext}}^i + a_{\text{sink-gas}}^i + a_{\text{selfgrav}}^i, \quad (166)$$

$$\frac{du}{dt} = -\frac{P}{\rho} (\nabla \cdot \mathbf{v}) + \Lambda_{\text{shock}} - \Lambda_{\text{cool}}, \quad (167)$$

$$\frac{d\mathbf{B}}{dt} = (\mathbf{B} \cdot \nabla) \mathbf{v} - \mathbf{B} (\nabla \cdot \mathbf{v}) - \nabla \psi + \mathcal{D}_{\text{diss}}, \quad (168)$$

$$\frac{d}{dt} \left(\frac{\psi}{c_h} \right) = -c_h (\nabla \cdot \mathbf{B}) - \frac{1}{2} \frac{\psi}{c_h} (\nabla \cdot \mathbf{v}) - \frac{\psi}{c_h \tau_c}, \quad (169)$$

where \mathbf{B} is the magnetic field, ψ is a scalar used to control the divergence error in the magnetic field (see Section 2.11.8, below), $\mathcal{D}_{\text{diss}}$ represents magnetic dissipation terms (Sections 2.11.5 and 2.12, below), M_{ij} is the Maxwell stress tensor, given by

$$M^{ij} = \left(P + \frac{1}{2} \frac{B^2}{\mu_0} \right) \delta^{ij} - \frac{B^i B^j}{\mu_0}, \quad (170)$$

where δ^{ij} is the Kronecker delta and μ_0 is the permeability of free space, and f_{divB}^i is a source term related to the divergence of the magnetic field,

$$f_{\text{divB}}^i \equiv -\frac{B^i}{\rho} (\nabla \cdot \mathbf{B}), \quad (171)$$

which is necessary to prevent the tensile instability in SPMHD (Phillips & Monaghan, 1985; Monaghan, 2000; Børve et al., 2001; Price, 2012a). With this source term the equation set for ideal MHD is formally the same as in the Powell et al. (1999) 8-wave scheme (Price, 2012a). An isothermal or barotropic equation of state may be used in place of (167).

2.11.2 Code units

When using magnetic fields an additional unit is required to describe the unit of magnetic field. As is usual practice, we adopt code units such that $\mu_0 = 1$. The unit scalings for the magnetic field can be determined from the definition of the Alfvén speed,

$$v_A \equiv \sqrt{\frac{B^2}{\mu_0 \rho}}. \quad (172)$$

Since the Alfvén speed has dimensions of length per unit time, this implies a unit for the magnetic field given by

$$u_{\text{mag}} = \left(\frac{\mu_0 u_{\text{mass}}}{u_{\text{dist}} u_{\text{time}}} \right)^{\frac{1}{2}}. \quad (173)$$

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.

2.11.3 Discrete equations

The discrete version of (166) follows the same procedure as for physical viscosity (Section 2.8), i.e.

$$\begin{aligned} \frac{dv_a^i}{dt} = & - \sum_b m_b \left[\frac{M_a^{ij}}{\Omega_a \rho_a^2} \frac{\partial W_{ab}(h_a)}{\partial x_a^j} + \frac{M_b^{ij}}{\Omega_b \rho_b^2} \frac{\partial W_{ab}(h_b)}{\partial x_b^j} \right] \\ & + \Pi_{\text{shock}}^a + f_{\text{divB},a}^i + a_{\text{ext},a}^i + a_{\text{sink-gas}}^i \\ & + a_{\text{selfgrav}}^i, \end{aligned} \quad (174)$$

where M_a^{ij} is defined according to (170), f_{divB} is a correction term for stability (discussed below), and accelerations due to external forces are as described in Section 2.5.

Equations (168) and (169) are discretised according to (Price & Monaghan, 2005; Tricco & Price, 2012; Tricco et al., 2016a)

$$\begin{aligned} \frac{d\mathbf{B}_a}{dt} = & - \frac{1}{\Omega_a \rho_a} \sum_b m_b \mathbf{v}_{ab} [\mathbf{B}_a \cdot \nabla_a W_{ab}(h_a)] \\ & + \frac{\mathbf{B}_a}{\Omega_a \rho_a} \sum_b m_b \mathbf{v}_{ab} \cdot \nabla_a W_{ab}(h_a) \\ & - \rho_a \sum_b m_b \left[\frac{\psi_a}{\Omega_a \rho_a^2} \nabla_a W_{ab}(h_a) + \frac{\psi_b}{\Omega_b \rho_b^2} \nabla_a W_{ab}(h_b) \right] \\ & + \mathcal{D}_{\text{diss}}^a, \end{aligned} \quad (175)$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\psi}{c_h} \right)_a = & \frac{c_h^a}{\Omega_a \rho_a} \sum_b m_b \mathbf{B}_{ab} \cdot \nabla_a W_{ab}(h_a) \\ & + \frac{\psi_a}{2c_h^a \Omega_a \rho_a} \sum_b m_b \mathbf{v}_{ab} \cdot \nabla_a W_{ab}(h_a) - \frac{\psi_a}{c_h^a \tau_c^a}, \end{aligned} \quad (176)$$

The first term in Equation (176) uses the divergence of the magnetic field discretised according to

$$(\nabla \cdot \mathbf{B})_a = - \frac{1}{\Omega_a \rho_a} \sum_b m_b (\mathbf{B}_a - \mathbf{B}_b) \cdot \nabla_a W_{ab}(h_a), \quad (177)$$

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

2.11.4 Tensile instability correction

The correction term f_{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 $\frac{1}{2}B^2 > P$ (Phillips & Monaghan, 1985). We compute this using the symmetric divergence operator (Børve et al., 2001; Price, 2012a; Tricco & Price, 2012)

$$f_{\text{divB},a}^i = -\hat{B}_a^i \sum_b m_b \left[\frac{\mathbf{B}_a \cdot \nabla_a W_{ab}(h_a)}{\Omega_a \rho_a^2} + \frac{\mathbf{B}_b \cdot \nabla_a W_{ab}(h_b)}{\Omega_b \rho_b^2} \right]. \quad (178)$$

Since this term violates momentum conservation to the extent that the $\nabla \cdot \mathbf{B}$ 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 $\beta = \frac{1}{2}B^2/P < 1$. Barnes et al. (2012) similarly advocated using a factor of $\frac{1}{2}$ in this term. However, we showed in Tricco & Price (2012) that this could lead to problematic effects (their Figure 12). In PHANTOM we use

$$\hat{B}^i = \begin{cases} B^i & \beta < 1; \\ (2 - \beta)B^i & 1 < \beta < 2; \\ 0 & \text{otherwise.} \end{cases} \quad (179)$$

which retains stability but eliminates the non-conservation of momentum in the weak field regime.

2.11.5 Shock capturing

The shock capturing term in the momentum equation for MHD is identical to (36) and (38) except that the signal speed becomes (Price & Monaghan, 2004a, 2005; Price, 2012a)

$$v_{\text{sig},a} = \alpha_a^{\text{AV}} v_a + \beta^{\text{AV}} |\mathbf{v}_{ab} \cdot \hat{\mathbf{r}}_{ab}|, \quad (180)$$

where

$$v_a = \sqrt{c_{s,a}^2 + v_{A,a}^2}, \quad (181)$$

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

$$\mathcal{D}_{\text{diss}}^a = \frac{\rho_a}{2} \sum_b m_b \left[\frac{v_{\text{sig},a}^B}{\rho_a^2} \frac{F_{ab}(h_a)}{\Omega_a} + \frac{v_{\text{sig},b}^B}{\rho_b^2} \frac{F_{ab}(h_b)}{\Omega_b} \right] \mathbf{B}_{ab}, \quad (182)$$

where $v_{\text{sig},a}^B$ 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 Equation (40) is given by

$$\Lambda_{\text{artres}} = -\frac{1}{4} \sum_b m_b \left[\frac{v_{\text{sig},a}^B}{\rho_a^2} \frac{F_{ab}(h_a)}{\Omega_a} + \frac{v_{\text{sig},b}^B}{\rho_b^2} \frac{F_{ab}(h_b)}{\Omega_b} \right] (\mathbf{B}_{ab})^2. \quad (183)$$

As with the artificial viscosity, Equations (182) and (183) are the SPH representation of a physical resistivity term, $\eta \nabla^2 \mathbf{B}$, but with a coefficient that is proportional to resolution (Price & Monaghan, 2004a). The resistive dissipation rate from the shock capturing term is given by

$$\eta \approx \frac{1}{2} \alpha^B v_{\text{sig}}^B |r_{ab}|, \quad (184)$$

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

2.11.6 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, setting v_{sig}^B equal to the magnetosonic speed (Equation 181) multiplied by the dimensionless coefficient α_B , set according to

$$\alpha_a^B = \min \left(h_a \frac{|\nabla \mathbf{B}_a|}{|\mathbf{B}_a|}, \alpha_{\text{max}}^B \right), \quad (185)$$

where $|\nabla \mathbf{B}_a|$ is the 2-norm of the gradient tensor, i.e. the root mean square of all 9 components of this tensor and $\alpha_{\text{max}}^B = 1.0$ by default. Unlike the viscous dissipation, this is set based on the instantaneous values of h and \mathbf{B} and there is no source/decay equation involved, since Tricco & Price (2013) found it to be unnecessary. Since α^B is proportional to resolution, from (184) we see that this results in dissipation that is effectively second order ($\propto h^2$). When individual particle timesteps are used, inactive particles retained their value of α^B from the last time they were active.

However, more recently we have found that a better approach, similar to that used for artificial conductivity, is to simply set the signal speed for resistivity according to

$$v_{\text{sig}}^B = |\mathbf{v}_{ab} \times \hat{\mathbf{r}}_{ab}|, \quad (186)$$

and setting $\alpha_B = 1$. We found that this produces better results on all of our tests (Section 4.6), in particular producing zero numerical dissipation on the current loop

advection test (Section 4.6.5) and, as with the Tricco & Price (2013) switch, giving second-order dissipation in the magnetic field (demonstrated in Section 4.6.1; Figure 27). Hence this is now the default treatment for artificial resistivity in PHANTOM.

2.11.7 Conservation properties

The total energy when using MHD is given by

$$E_{\text{tot}} = \sum_a m_a \left(\frac{1}{2} v_a^2 + u_a + \Phi_a + \frac{1}{2} \frac{B_a^2}{\mu_0 \rho_a} \right). \quad (187)$$

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

$$\frac{dE}{dt} = \sum_a m_a \left[\mathbf{v}_a \cdot \frac{d\mathbf{v}_a}{dt} + \frac{du_a}{dt} + \mathbf{v}_a \cdot \nabla \Phi_a - \frac{B_a^2}{\mu_0 \rho_a^2} \frac{d\rho_a}{dt} + \frac{\mathbf{B}_a}{\mu_0 \rho_a} \cdot \frac{d\mathbf{B}_a}{dt} \right] = 0, \quad (188)$$

while the total momentum and angular momentum are as previously. Total energy is conserved by the ideal MHD and shock capturing terms, but the presence of the f_{divB} correction term (necessary for numerical stability) violates the conservation of momentum and energy. Hence inadequate divergence control (see below) usually manifests as a loss of momentum conservation in the code (see Tricco & Price, 2012, for details).

2.11.8 Divergence cleaning

To preserve the divergence-free condition on the magnetic field, PHANTOM implements the constrained hyperbolic/parabolic divergence cleaning algorithm described in detail by Tricco & Price (2012) and Tricco et al. (2016a), who fixed a number of issues with earlier formulations by Dedner et al. (2002) and Price & Monaghan (2005). The key point is that the cleaning term (3rd term on the right hand side in Equation (168)) and (169) combine to give the following equation for the propagation and damping of the ψ field

$$\frac{d}{dt} \left[\frac{1}{\sqrt{\rho} c_h} \frac{d}{dt} \left(\frac{\psi}{\sqrt{\rho} c_h} \right) \right] - \frac{\nabla^2 \psi}{\rho} + \frac{d}{dt} \left[\frac{1}{\sqrt{\rho} c_h} \left(\frac{\psi}{\sqrt{\rho} c_h \tau} \right) \right] = 0. \quad (189)$$

When c_h , ρ , τ_c and the fluid velocity are constant this reduces to the usual damped wave equation in the form

$$\frac{\partial^2 \psi}{\partial t^2} - c_h^2 \nabla^2 \psi + \frac{1}{\tau_c} \frac{\partial \psi}{\partial t} = 0, \quad (190)$$

with the same equation holding for the evolution of $\nabla \cdot \mathbf{B}$ itself, i.e.

$$\frac{\partial^2 (\nabla \cdot \mathbf{B})}{\partial t^2} - c_h^2 \nabla^2 (\nabla \cdot \mathbf{B}) + \frac{1}{\tau_c} \frac{\partial (\nabla \cdot \mathbf{B})}{\partial t} = 0, \quad (191)$$

where we set the cleaning speed c_h equal to the magnetosonic speed (Equation 181), as recommended by Tricco & Price (2013). The decay time is given by

$$\tau_c^a \equiv \frac{h_a}{\sigma_c c_{h,a}}. \quad (192)$$

The dimensionless factor σ_c sets the ratio of parabolic to hyperbolic cleaning (referred to as `psidecayfac` in the input file), set to $\sigma_c = 1.0$ by default since this was found by Tricco & Price 2012 to provide the 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)

$$\left(\frac{dE}{dt}\right)_{\text{cleaning}} = - \sum_a m_a \frac{\psi_a^2}{\mu_0 \rho_a c_{h,a}^2 \tau_c^a}. \quad (193)$$

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.

2.11.9 Monitoring of divergence errors and over-cleaning

While the divergence cleaning algorithm is guaranteed to either conserve or dissipate magnetic energy, it is not always sufficient to keep the divergence error small (though it is adequate for many applications). Hence the onus is on the user to ensure that simulation results are not corrupted by divergence errors, which may be checked by monitoring the dimensionless quantity

$$\epsilon_{\text{div}B} \equiv \frac{h|\nabla \cdot \mathbf{B}|}{|\mathbf{B}|}. \quad (194)$$

The maximum and mean values of this quantity on the particles are written to the `.ev` file and should be used to check the fidelity of any simulation where the MHD algorithm is used. A good rule-of-thumb is that the mean value should remain below $\sim 10^{-2}$ for the simulation to remain unaffected by divergence errors.

For the above reason, in the code we introduce an additional factor to the cleaning speed in the form

$$c_h = f_{\text{clean}} v, \quad (195)$$

where f_{clean} is an ‘over-cleaning’ factor (`overcleanfac`) which can be used to arbitrarily increase the cleaning wave speed to reduce divergence errors, at the cost of a corresponding reduction in the timestep.

2.12 Non-ideal magnetohydrodynamics

PHANTOM includes a module implementing physical dissipation in the magnetic field, as described in Wurster et al. (2014, 2016). This includes implementations of physical resistivity, ambipolar (ion-neutral) diffusion and the Hall effect, See Wurster et al. 2016,

2017 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, similar to the ‘one fluid’ formulation of dust (Section 2.14.12), but is similar to other recent implementations, in particular the one by Tsukamoto et al. (2013, 2015).

2.12.1 Continuum equations

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 Equation (168) contains three additional terms in the form

$$\left(\frac{d\mathbf{B}}{dt}\right)_{\text{non-ideal}} = \nabla \times \left[\frac{\mathbf{J}}{\sigma_e} + \frac{\mathbf{J} \times \mathbf{B}}{en_e} + \frac{(\mathbf{J} \times \mathbf{B}) \times \mathbf{B}}{\gamma_{\text{AD}} \rho_i} \right], \quad (196)$$

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

$$\left(\frac{d\mathbf{B}}{dt}\right)_{\text{non-ideal}} = \nabla \times \left[\eta_{\text{O}} \mathbf{J} + \eta_{\text{H}} \mathbf{J} \times \hat{\mathbf{B}} + \eta_{\text{AD}} (\mathbf{J} \times \hat{\mathbf{B}}) \times \hat{\mathbf{B}} \right], \quad (197)$$

where $\hat{\mathbf{B}}$ is the unit vector in the direction of \mathbf{B} such that η_{O} , η_{AD} and η_{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 corresponding resistive and ambipolar heating terms in the thermal energy equation in the form

$$\left(\frac{du}{dt}\right)_{\text{non-ideal}} = \frac{\eta_{\text{O}}}{\rho} J^2 + \frac{\eta_{\text{AD}}}{\rho} [J^2 B^2 - (\mathbf{J} \cdot \mathbf{B})^2]. \quad (198)$$

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

2.12.2 Discrete equations

Our main constraint is that the algorithm should exactly conserve energy, which is achieved by discretisation in the form (Wurster et al., 2014)

$$\left(\frac{d\mathbf{B}_a}{dt}\right)_{\text{non-ideal}} = -\rho_a \sum_b \left[\frac{\mathbf{D}_a}{\Omega_a \rho_a^2} \times \nabla_a W_{ab}(h_a) + \frac{\mathbf{D}_b}{\Omega_b \rho_b^2} \times \nabla_a W_{ab}(h_b) \right], \quad (199)$$

where

$$\mathbf{D} \equiv \eta_{\text{O}} \mathbf{J} + \eta_{\text{H}} (\mathbf{J} \times \hat{\mathbf{B}}) + \eta_{\text{AD}} [(\mathbf{J} \times \hat{\mathbf{B}}) \times \hat{\mathbf{B}}]. \quad (200)$$

The corresponding term in the energy equation is given by

$$\left(\frac{du_a}{dt} \right)_{\text{non-ideal}} = - \frac{\mathbf{D}_a \cdot \mathbf{J}_a}{\rho_a}, \quad (201)$$

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

$$\mathbf{J} = \frac{1}{\Omega_a \rho_a} \sum_b m_b (\mathbf{B}_a - \mathbf{B}_b) \times \nabla_a W_{ab}(h_a). \quad (202)$$

Non-ideal MHD therefore utilises a ‘two first derivatives’ approach, similar to the formulation of physical viscosity described in Section 2.8.1.

2.12.3 Computing the non-ideal MHD coefficients

To compute the coefficients η_{O} , η_{Hall} and η_{AD} , we use the NICIL library (Wurster, 2016) for cosmic ray ionisation chemistry, which is distributed with PHANTOM. We refer the reader to Wurster (2016) and Wurster et al. (2016) for details, since this is maintained and versioned as a separate package.

2.12.4 Timestepping

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

$$\Delta t = \frac{C_{\text{nimhd}} h^2}{\max(\eta_{\text{O}}, \eta_{\text{AD}}, |\eta_{\text{H}}|)} \quad (203)$$

where $C_{\text{nimhd}} = 1/(2\pi)$ by default. This can prove prohibitive, so we employ the so-called ‘super-timestepping’ algorithm from Alexiades et al. (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.

2.13 Self-gravity

For self-gravity we split the computation into a ‘short-range’ interaction, computed by direct summation over nearby particles, and a ‘long-range’ interaction computed by hierarchical grouping of particles using the tree. By self-gravity we mean a solution to Poisson’s equation

$$\nabla^2 \Phi = 4\pi G \rho(\mathbf{r}), \quad (204)$$

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

$$\mathbf{a}_{\text{selfgrav}} = -\nabla \Phi. \quad (205)$$

Since (204) is an elliptic equation, implying instant action, it requires a global solution.

2.13.1 Short-range interaction

How to solve (204) on particles is one of the most widely misunderstood problems in astrophysics. In SPH or collisionless N-body simulations (i.e. stars, 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, and then solve (204) 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 (3) to both reconstruct the density field and set the softening length, i.e.⁷

$$\rho_a = \sum_b m_b W_{ab}(\epsilon_a); \quad \epsilon_a = h_{\text{fac}}(m_a/\rho_a)^{1/3}, \quad (206)$$

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

$$\Phi_a = -G \sum_b m_b \phi(|\mathbf{r}_a - \mathbf{r}_b|, \epsilon_a), \quad (207)$$

where ϕ is the softening kernel derived from the density kernel via Poisson’s equation (Section 2.13.2, below). For a variable smoothing length energy conservation necessitates taking an average of the kernels, i.e.

$$\Phi_a = -G \sum_b m_b \left[\frac{\phi_{ab}(\epsilon_a) + \phi_{ab}(\epsilon_b)}{2} \right]. \quad (208)$$

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

$$\begin{aligned} \mathbf{a}_{\text{selfgrav}}^a &= -\nabla \Phi_a, \\ &= -G \sum_b m_b \left[\frac{\phi'_{ab}(\epsilon_a) + \phi'_{ab}(\epsilon_b)}{2} \right] \hat{\mathbf{r}}_{ab} \\ &\quad - \frac{G}{2} \sum_b m_b \left[\frac{\zeta_a}{\Omega_a^\epsilon} \nabla_a W_{ab}(\epsilon_a) + \frac{\zeta_b}{\Omega_b^\epsilon} \nabla_a W_{ab}(\epsilon_b) \right], \end{aligned} \quad (209)$$

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

$$\zeta_a \equiv \frac{\partial \epsilon_a}{\partial \rho_a} \sum_b m_b \frac{\partial \phi_{ab}(\epsilon_a)}{\partial \epsilon_a}. \quad (210)$$

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

⁷Strictly one should use the number density instead of the mass density when computing the softening length via (206), but as we enforce equal masses for each particle type in PHANTOM, the two methods are equivalent.

For SPH it is natural to set the softening length equal to the smoothing length $\epsilon = h$ 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 approach ([Bagla & Khandai, 2009](#); [Iannuzzi & Dolag, 2011](#)).

2.13.2 Functional form of the softening kernel

The density kernel and softening potential kernel are related using Poisson’s equation (204), i.e.

$$W(r, \epsilon) = \frac{1}{4\pi r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial \phi}{\partial r} \right), \quad (211)$$

where $r \equiv |\mathbf{r} - \mathbf{r}'|$. Integrating this equation gives the softening kernels used for the force and gravitational potential. As with the standard kernel functions (Section 2.1.6) we define the potential and force kernels in terms of a dimensionless functions

$$\phi(r, \epsilon) \equiv \frac{1}{\epsilon} \tilde{\phi}(q), \quad (212)$$

$$\phi'(r, \epsilon) \equiv \frac{1}{\epsilon^2} \tilde{\phi}'(q), \quad (213)$$

where the dimensionless force kernel is obtained from the density kernel $f(q)$ (Section 2.1.6–2.1.7) using

$$\tilde{\phi}'(q) = \frac{4\pi}{q^2} C_{\text{norm}} \int f(q') q'^2 dq', \quad (214)$$

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

$$\tilde{\phi}(q) = \int \tilde{\phi}'(q') dq'. \quad (215)$$

The derivative of ϕ with respect to ϵ required in (210) is also written in terms of a dimensionless function, i.e.

$$\frac{\partial \phi(r, \epsilon)}{\partial \epsilon} \equiv \frac{1}{\epsilon^2} h(q), \quad (216)$$

where from differentiating (212) we have

$$h(q) = -\tilde{\phi}(q) - q\tilde{\phi}'(q). \quad (217)$$

Included in the PHANTOM source code is a PYTHON script (`kernels.py`) using the SYMPY library to solve (214), (215) and (217) 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

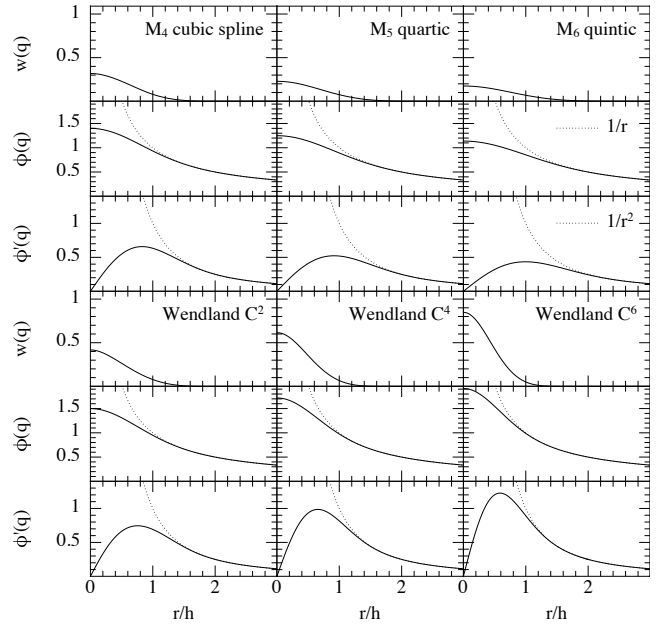


Figure 7. Functional form of the softening kernel functions $-\phi(r, h)$ and $\phi'(r, h)$ used to compute the gravitational force in PHANTOM, shown for each of the available kernel functions $w(r, h)$ (see Figure 1). Dotted lines show the functional form of the unsoftened potential ($-1/r$) and force ($1/r^2$) for comparison.

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 7 shows the functional form of the softening kernels for each of the kernels available in PHANTOM. The kernel function $f(q)$ is shown for comparison (top row in each case).

2.13.3 Softening of 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 2.14.3). 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 (209). This naturally symmetrises the softening between different types and ensures that momentum, angular momentum and energy are conserved.

2.13.4 Long-range interaction

At long range $r > R_{\text{kern}}\epsilon_a$ and $r > R_{\text{kern}}\epsilon_b$, the second term in (209) tends to zero (since $\zeta = 0$ for $q \geq R_{\text{kern}}$) while the first term simply becomes $1/r^2$ direct summation which would have an associated $\mathcal{O}(N^2)$ computational cost. We adopt the usual practice of using the treecode 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). As usual, we split the gravitational acceleration into short range and long range contributions,

$$\mathbf{a}_{\text{selfgrav}}^a = \mathbf{a}_{\text{short}}^a + \mathbf{a}_{\text{long}}^a, \quad (218)$$

where ‘long range’ and ‘short range’ are determined for each node-node pair n - m according to the tree opening criterion,

$$\theta^2 < \left(\frac{s_m}{r_{nm}} \right)^2, \quad (219)$$

where $0 \leq \theta \leq 1$ is the tree opening parameter; or by nodes falling within each others smoothing spheres,

$$r_{nm}^2 < [s_n + s_m + \max(R_{\text{kern}}h_{\text{max}}^n, R_{\text{kern}}h_{\text{max}}^m)]^2, \quad (220)$$

where as previously s is the node size (the radius about the centre of mass that contains all the particles). Node pairs satisfying either of these criterion have the particles contained within them added to the trial neighbour list. Setting $\theta = 0$ therefore leads to a direction summation of (209), while $\theta = 1$ gives the fastest possible but least accurate gravitational force evaluation. $\theta = 0.5$ is the code default.

The short range acceleration is then evaluated for each particle in the leaf node n by summing (209) over the trial neighbour list. For pairs outside the softening radius of either particle, the short range interaction reduces to

$$\mathbf{a}_{\text{short}, r > R_{\text{kern}}\epsilon}^a = -G \sum_b m_b \frac{\mathbf{r}_a - \mathbf{r}_b}{|\mathbf{r}_a - \mathbf{r}_b|^3}. \quad (221)$$

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

The long range acceleration on a given leaf node n consists of a sum over distant nodes m that satisfy neither (219) nor (220),

$$\mathbf{a}_n = \sum_m \mathbf{a}_{nm}, \quad (222)$$

where the acceleration along the node centres between a given pair is given (using index notation) by

$$a_{nm}^i = -\frac{GM_m}{r^3} r^i + \frac{1}{r^4} \left(\hat{r}^k Q_{ik}^m - \frac{5}{2} \hat{r}^i Q^m \right), \quad (223)$$

where $r^i \equiv x_n^i - x_m^i$ is the relative position vector, \hat{r} is the corresponding unit vector, M_m is the total mass in node m , Q_{ij}^m is the quadrupole moment of node m , repeated indices imply summation and we define the following scalar and vector quantities for convenience:

$$\mathcal{Q} \equiv \hat{r}^i \hat{r}^j Q_{ij}, \quad (224)$$

$$\mathcal{Q}_i \equiv \hat{r}^j Q_{ij}. \quad (225)$$

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

$$\frac{\partial a_n^i}{\partial r^j} = \sum_m \frac{\partial a_{nm}^i}{\partial r^j}, \quad (226)$$

where

$$\begin{aligned} \frac{\partial a_{nm}^i}{\partial r^j} = & \frac{GM_m}{r^3} \left[3\hat{r}^i \hat{r}^j - \delta^{ij} \right] \\ & + \frac{1}{r^5} \left[Q_{ij}^m + \left(\frac{35}{2} \hat{r}^i \hat{r}^j - \frac{5}{2} \delta_{ij} \right) \mathcal{Q}^m \right. \\ & \left. - 5\hat{r}^i \mathcal{Q}_j^m - 5\hat{r}^j \mathcal{Q}_i^m \right], \quad (227) \end{aligned}$$

together with the ten independent components of the second derivatives, given by

$$\begin{aligned} \frac{\partial^2 a_{nm}^i}{\partial r^j \partial r^k} = & -\frac{3GM_m}{r^4} \left[5\hat{r}^i \hat{r}^j \hat{r}^k - \delta_{jk} \hat{r}^i - \delta_{ik} \hat{r}^j - \delta_{ij} \hat{r}^k \right] \\ & + \frac{1}{r^6} \left[-5(\hat{r}^k Q_{ij}^m + \hat{r}^i Q_{jk}^m + \hat{r}^j Q_{ik}^m) \right. \\ & - \frac{315}{2} \hat{r}^i \hat{r}^j \hat{r}^k \mathcal{Q}^m \\ & + \frac{35}{2} (\delta_{ij} \hat{r}^k + \delta_{ik} \hat{r}^j + \delta_{jk} \hat{r}^i) \mathcal{Q}^m \\ & + 35 (\hat{r}^j \hat{r}^k \mathcal{Q}_i^m + \hat{r}^i \hat{r}^k \mathcal{Q}_j^m + \hat{r}^i \hat{r}^j \mathcal{Q}_k^m) \\ & \left. - 5(\delta_{ij} \mathcal{Q}_k^m + \delta_{ik} \mathcal{Q}_j^m + \delta_{jk} \mathcal{Q}_i^m) \right]. \quad (228) \end{aligned}$$

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

$$a_{\text{long},a}^i = a_n^i + \Delta x^j \frac{\partial a_n^i}{\partial r^j} + \frac{1}{2} \Delta x^j \Delta x^k \frac{\partial^2 a_n^i}{\partial r^j \partial r^k}, \quad (229)$$

where $\Delta x_a^i \equiv x_a^i - x_0^i$ is the relative distance of each particle from the node centre of mass. Pseudo-code for the resulting force evaluation is shown in Figure 8.

```

!$omp parallel do
do node = 1, number of nodes
  if (node is leaf node)
    call get_neigh(node,listneigh,nneigh,fnode)
    i = firstinnode(node)
    do while (i > 0)
      do n=1,nneigh
        j = listneigh(n)
        if (n <= cache size)
          get j position from cache
        else
          get j position from memory
        endif
        if (dr < Rkern*hi or dr < Rkern*hj)
          evaluate kernel and softening
          add to force sum
        else
          add 1/r^2 forces to force sum
        endif
      enddo
      get i distance from node centre
      expand fnode at position of i
      add long range terms to force sum
      i = next(i)
    enddo
  endif
enddo
!$omp end parallel do

```

Figure 8. 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.

The quadrupole moments are computed during the tree build using

$$Q_{ij} = \sum_a m_a [3\Delta x^i \Delta x^j - (\Delta x)^2 \delta^{ij}], \quad (230)$$

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 $\mathcal{O}(N_{\text{leafnodes}} \log N_{\text{part}})$ 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 highly desirable in a future code version.

2.14 Dust-gas mixtures

Modelling dust-gas mixtures is the first step towards the ‘grand challenge’ of protoplanetary disc modelling (Haworth et al., 2016). The public version of PHANTOM implements dust-gas mixtures using the two fluid formulation presented in Laibe & Price (2012a,b) and, for small grains, the one-fluid method described in Price & Laibe (2015a). Various combinations of these algorithms have been used in a number of our recent papers using PHANTOM, including Dipierro et al. (2015, 2016) and Ragusa 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.

2.14.1 Continuum equations

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

$$\frac{\partial \rho_g}{\partial t} + (\mathbf{v}_g \cdot \nabla) \rho_g = -\rho_g (\nabla \cdot \mathbf{v}_g), \quad (231)$$

$$\frac{\partial \rho_d}{\partial t} + (\mathbf{v}_d \cdot \nabla) \rho_d = -\rho_d (\nabla \cdot \mathbf{v}_d), \quad (232)$$

$$\frac{\partial \mathbf{v}_g}{\partial t} + (\mathbf{v}_g \cdot \nabla) \mathbf{v}_g = -\frac{\nabla P}{\rho_g} + \frac{K}{\rho_g} (\mathbf{v}_d - \mathbf{v}_g), \quad (233)$$

$$\frac{\partial \mathbf{v}_d}{\partial t} + (\mathbf{v}_d \cdot \nabla) \mathbf{v}_d = -\frac{K}{\rho_d} (\mathbf{v}_d - \mathbf{v}_g), \quad (234)$$

$$\frac{\partial u}{\partial t} + (\mathbf{v}_g \cdot \nabla) u = -\frac{P}{\rho_g} (\nabla \cdot \mathbf{v}_g) + \Lambda_{\text{drag}}, \quad (235)$$

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

$$\Lambda_{\text{drag}} \equiv K (\mathbf{v}_d - \mathbf{v}_g)^2. \quad (236)$$

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.

2.14.2 Stopping time

The stopping time

$$t_s = \frac{\rho_g \rho_d}{K(\rho_g + \rho_d)}, \quad (237)$$

is the characteristic timescale for the exponential decay of the differential velocity between the two phases caused by the drag. How t_s relates to other characteristic timescales defines the physics of the mixture. In the code t_s is by default specified in physical units, which means that code units need to be defined appropriately when simulating dust-gas mixtures.

2.14.3 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 Eqs. 231–232 are discretised with a density summation over neighbours of the same type (c.f. Section 2.13.3), giving

$$\rho_a = \sum_b m_b W_{ab}(h_a); \quad h_a = h_{\text{fact}} \left(\frac{m_a}{\rho_a} \right)^{1/3}, \quad (238)$$

for a gas particle, and

$$\rho_i = \sum_j m_j W_{ij}(h_i); \quad h_i = h_{\text{fact}} \left(\frac{m_i}{\rho_i} \right)^{1/3}, \quad (239)$$

for a dust particle. The kernel used for density is the same as usual (Section 2.1.7). We discretise the equations of motion for the gas particles (233) using

$$\left(\frac{d\mathbf{v}_a}{dt} \right)_{\text{drag}} = -3 \sum_j m_j \frac{\mathbf{v}_{aj} \cdot \hat{\mathbf{r}}_{aj}}{(\rho_a + \rho_j) t_{aj}^s} \hat{\mathbf{r}}_{aj} D_{aj}(h_a), \quad (240)$$

and for dust (234) is given by

$$\left(\frac{d\mathbf{v}_i}{dt} \right)_{\text{drag}} = -3 \sum_b m_b \frac{\mathbf{v}_{ib} \cdot \hat{\mathbf{r}}_{ib}}{(\rho_i + \rho_b) t_{ib}^s} \hat{\mathbf{r}}_{ib} D_{ib}(h_b), \quad (241)$$

where D is a double hump kernel, defined in Section 2.14.4, below. The drag heating term in the energy equation (235) is discretised using

$$\Lambda_{\text{drag}} = 3 \sum_j m_j \frac{(\mathbf{v}_{aj} \cdot \hat{\mathbf{r}}_{aj})^2}{(\rho_a + \rho_j) t_{aj}^s} D_{aj}(h_a). \quad (242)$$

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 grains are km-sized or smaller, much less than typical simulation scales of \sim au or larger.

2.14.4 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 2.1.6),

$$D(r, h) = \frac{\sigma}{h^3} g(q), \quad (243)$$

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

$$g(q) = q^2 f(q), \quad (244)$$

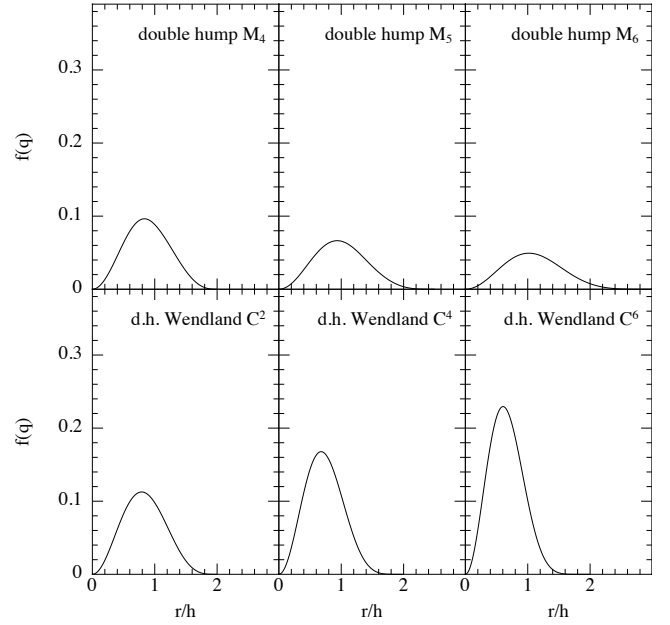


Figure 9. Double hump smoothing kernels $D(r, h)$ available in PHANTOM, used in the computation of the dust-gas drag force.

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

$$\int D(r, h) dV = 1. \quad (245)$$

Figure 9 shows the functional form of the double hump cubic spline kernel. 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 (240), (241) and (242) — it does not matter since D is also zero at the origin.

The double hump kernel functions can be generated automatically from the corresponding density kernel by the `kernels.py` script distributed with PHANTOM, as described in Section 2.13.2. The pre-generated modules implementing each of the kernels described in Section 2.1.7 hence automatically contain the corresponding double hump kernel function, which is used to compute the drag terms.

2.14.5 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.

$$t_{aj}^s = \frac{\rho_a \rho_j}{K_{aj}(\rho_a + \rho_j)}. \quad (246)$$

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). An option to use a constant K between pairs is also implemented, useful for testing and debugging (c.f. Section 4.10).

2.14.6 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

$$K_n = \frac{9\lambda_g}{4s_{\text{grain}}}, \quad (247)$$

where s_{grain} is the grain size and λ_g is the gas mean free path (see Section 2.14.8, below, for how this is evaluated). For $K_n \geq 1$ 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

$$K_{aj} = \rho_g^a \rho_d^j \frac{4}{3} \sqrt{\frac{8\pi}{\gamma}} \frac{s_{\text{grain}}^2}{m_{\text{grain}}} c_s^a f, \quad (248)$$

where

$$m_{\text{grain}} \equiv \frac{4}{3} \pi \rho_{\text{grain}} s_{\text{grain}}^3, \quad (249)$$

and ρ_{grain} is the intrinsic grain density, which is 3 g/cm^3 by default but appears as a runtime parameter (`graindenscgs`) in the PHANTOM input file. The variable f is a correction for supersonic drift velocities given by (Kwok, 1975)

$$f \equiv \sqrt{1 + \frac{9\pi}{128} \frac{\Delta v^2}{c_s^2}}, \quad (250)$$

where $\Delta v \equiv |\mathbf{v}_d - \mathbf{v}_g| = v_d^j - v_g^a$. The stopping time is therefore

$$t_s = \frac{\rho_{\text{grain}} s_{\text{grain}}}{\rho c_s f} \sqrt{\frac{\pi \gamma}{8}}, \quad (251)$$

where $\rho \equiv \rho_d + \rho_g$. This formula (251) reduces to the standard expression for the linear Epstein regime in the limit where the drift velocity is small compared to the sound speed (i.e. $f \rightarrow 1$). Figure 10 shows the difference between the above simplified prescription and the exact expression for Epstein drag (Epstein 1924; c.f. Eqs. 11 and 38 in Laibe & Price 2012b) as a function of $\Delta v/c_s$, which is less than 1% everywhere.

2.14.7 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

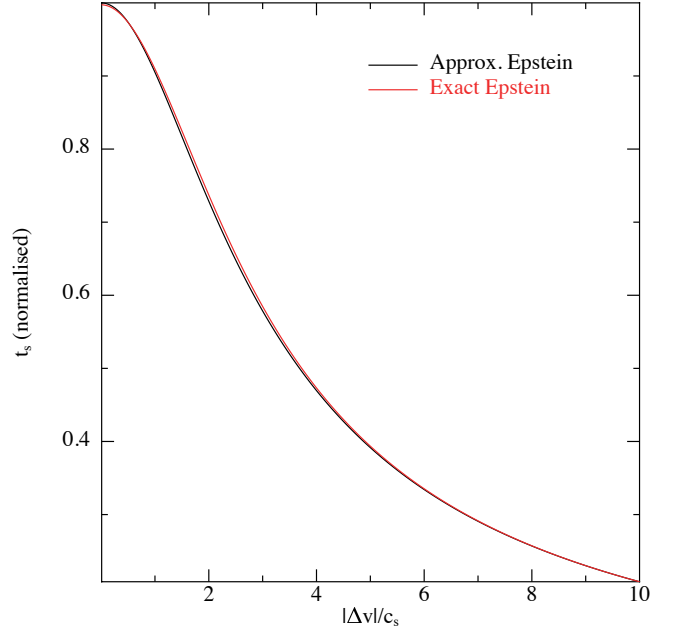


Figure 10. Dependence of the drag stopping time t_s on differential Mach number, showing the increased drag (decrease in stopping time) as Δv becomes supersonic. The black line shows the analytic approximation used in PHANTOM (Equation 251) and the exact expression from Epstein (1924) (red line). The difference is less than 1% everywhere.

use (Laibe & Price, 2012b)

$$K_{aj} = \rho_g^a \rho_d^j \frac{1}{2} C_D \frac{\pi s_{\text{grain}}^2}{m_{\text{grain}}} |\Delta v|, \quad (252)$$

where the coefficient C_D is given by

$$C_D = \begin{cases} 24R_e^{-1}, & R_e < 1, \\ 24R_e^{-0.6}, & 1 < R_e < 800, \\ 0.44, & R_e > 800, \end{cases} \quad (253)$$

where R_e is the local Reynolds number around the grain

$$R_e \equiv \frac{2s_{\text{grain}} |\Delta v|}{\nu}. \quad (254)$$

where ν is the microscopic viscosity of the gas (see below; not to be confused with the disc viscosity). Other, 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

$$t_s = \frac{8\rho_{\text{grain}} s_{\text{grain}}}{3\rho |\Delta v| C_D}, \quad (255)$$

where it remains to evaluate ν and λ_g .

2.14.8 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

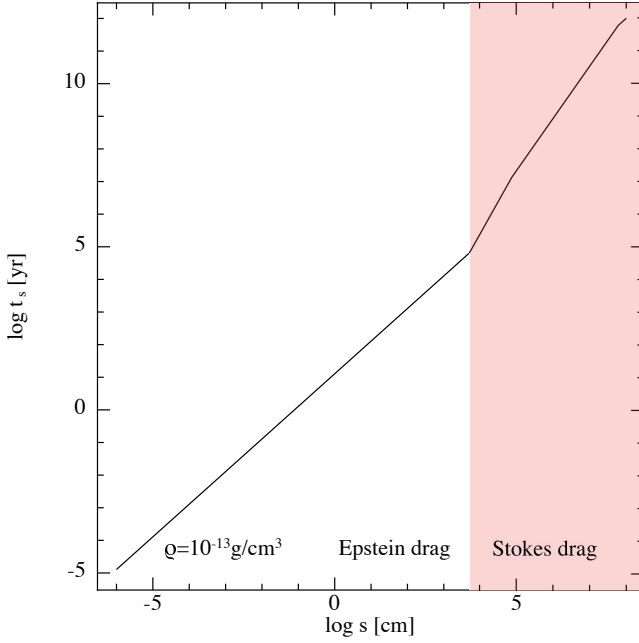


Figure 11. Drag stopping time t_s (in years) as a function of grain size, showing the continuous transition between the Epstein and Stokes drag regimes. The example shown assumes fixed density $\rho = 10^{-13} \text{g/cm}^3$ and sound speed $c_s = 6 \times 10^4 \text{cm/s}$ with subsonic drag $\Delta v = 0.01c_s$ and material density $\rho_{\text{grain}} = 3 \text{g/cm}^3$.

from the mean free path and sound speed according to

$$\nu = \sqrt{\frac{2}{\pi\gamma}} c_s \lambda_g, \quad (256)$$

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

$$\mu_\nu = \frac{5m}{64\sigma_s} \sqrt{\frac{\pi}{\gamma}} c_s, \quad (257)$$

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

$$\lambda_g = \frac{5\pi}{64\sqrt{2}} \frac{1}{n_g \sigma_s}, \quad (258)$$

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

$$m = 2m_{\text{H}}, \quad (259)$$

$$\sigma_s = 2.367 \times 10^{-15} \text{cm}^2. \quad (260)$$

2.14.9 Stokes/Epstein transition

At the transition between the two regimes, assuming $R_e < 1$, Equation (255) reduces to

$$t_s = \frac{2\rho_{\text{grain}} s_{\text{grain}}^2}{9\rho c_s \lambda_g} \sqrt{\frac{\pi\gamma}{2}}, \quad (261)$$

which is the same as the Epstein drag in the subsonic regime when $\lambda_g = 4s_{\text{grain}}/9$, i.e. $K_n = 1$. That this transition is indeed continuous in the code is demonstrated in Figure 11, which shows the transition from Epstein to Stokes drag and also through each of the Stokes regimes in Equation (253) by varying the grain size while keeping the other parameters fixed. For simplicity we assumed $\Delta v = 0.01c_s$ for this plot, even though in general one would expect Δv to increase with stopping time (c.f. Equation 271).

2.14.10 Self gravity of dust

With self-gravity turned on the dust particles are handled in the same way as stars or dark matter (Section 2.13.3), i.e. 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 2.9.2), but a sink cannot currently be created from dust particles (Section 2.9.4). There is currently no mechanism in the code to handle the collapse of dust to form a self-gravitating object independent of the gas.

2.14.11 Timestep constraint

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

$$\Delta t_{\text{drag}}^a = \min_j (t_s^{aj}). \quad (262)$$

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.

2.14.12 Continuum equations: One fluid

In Laibe & Price (2014a) we showed that the two fluid equations (231)–(235) can be rewritten as a single fluid mixture using a change of variables given by

$$\rho \equiv \rho_g + \rho_d, \quad (263)$$

$$\epsilon \equiv \rho_d/\rho, \quad (264)$$

$$\mathbf{v} \equiv \frac{\rho_g \mathbf{v}_g + \rho_d \mathbf{v}_d}{\rho_g + \rho_d}, \quad (265)$$

$$\Delta \mathbf{v} \equiv \mathbf{v}_d - \mathbf{v}_g, \quad (266)$$

In Laibe & Price (2014a) we derived the full set of evolution equations in these variables, and we implemented and tested an algorithm to solve these equations in SPH in Laibe & Price (2014c). 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 can use the ‘terminal velocity approximation’ (e.g. Youdin & Goodman, 2005; Chiang, 2008; Barranco, 2009; Lee et al., 2010; Jacquet et al., 2011) and the evolution equations reduce to (Laibe & Price, 2014a; Price & Laibe, 2015a)

$$\frac{d\rho}{dt} = -\rho\nabla \cdot \mathbf{v}, \quad (267)$$

$$\frac{d\mathbf{v}}{dt} = -\frac{\nabla P}{\rho} + \mathbf{a}_{\text{ext}}, \quad (268)$$

$$\frac{d\epsilon}{dt} = -\frac{1}{\rho}\nabla \cdot (\epsilon(1-\epsilon)\rho\Delta\mathbf{v}), \quad (269)$$

$$\frac{du}{dt} = -\frac{P}{\rho}\nabla \cdot \mathbf{v} + \epsilon(\Delta\mathbf{v} \cdot \nabla)u, \quad (270)$$

where

$$\Delta\mathbf{v} \equiv t_s(\mathbf{a}_d - \mathbf{a}_g), \quad (271)$$

where \mathbf{a}_d refers to any acceleration acting only on the dust phase and correspondingly for the gas. For the simple case of pure hydrodynamics the only difference is the pressure gradient, giving

$$\Delta\mathbf{v} \equiv t_s \frac{\nabla P}{\rho_g} = \frac{t_s}{(1-\epsilon)} \frac{\nabla P}{\rho}, \quad (272)$$

such that Equation (269) becomes

$$\frac{d\epsilon}{dt} = -\frac{1}{\rho}\nabla \cdot (\epsilon t_s \nabla P). \quad (273)$$

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

2.14.13 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 according to

$$\rho_g = (1-\epsilon)\rho, \quad (274)$$

$$\rho_d = \epsilon\rho, \quad (275)$$

$$\mathbf{v}_g = \mathbf{v} - \epsilon\Delta\mathbf{v}, \quad (276)$$

$$\mathbf{v}_d = \mathbf{v} + (1-\epsilon)\Delta\mathbf{v}. \quad (277)$$

This is handled automatically by SPLASH (Price, 2007) when reading PHANTOM data. In order to visualise the one-fluid results in a similar manner to those from the two-fluid method SPLASH reconstructs 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. The drift velocity is computed from (272) using the pressure gradient computed as in Equation (35) multiplied by the stopping time and the gas fraction, as described in Price & Laibe (2015a).

2.14.14 One fluid dust-gas implementation

The implementation of the one fluid method in PHANTOM follows Price & Laibe (2015a) with a few minor changes and corrections⁸. 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 (269) expressed in terms of s is given by

$$\frac{ds}{dt} = -\frac{1}{2s}\nabla \cdot \left(\frac{\rho_g \rho_d}{\rho} \Delta\mathbf{v} \right) - \frac{s}{2}\nabla \cdot \mathbf{v}, \quad (278)$$

which for the case of hydrodynamics becomes

$$\begin{aligned} \frac{ds}{dt} &= -\frac{1}{2s}\nabla \cdot \left(\frac{s^2}{\rho} t_s \nabla P \right) - \frac{s}{2}\nabla \cdot \mathbf{v}, \\ &= -\frac{1}{2}\nabla \cdot \left(\frac{s}{\rho} t_s \nabla P \right) - \frac{t_s}{2\rho} \nabla P \cdot \nabla s - \frac{s}{2}\nabla \cdot \mathbf{v}. \end{aligned} \quad (279)$$

The SPH discretisation of this equation is implemented in the form

$$\begin{aligned} \frac{ds_a}{dt} &= -\frac{1}{2} \sum_b \frac{m_b s_b}{\rho_b} \left(\frac{t_{s,a}}{\rho_a} + \frac{t_{s,b}}{\rho_b} \right) (P_a - P_b) \frac{\bar{F}_{ab}}{|r_{ab}|} \\ &\quad + \frac{s_a}{2\rho_a \Omega_a} \sum_b m_b \mathbf{v}_{ab} \cdot \nabla_a W_{ab}(h_a), \end{aligned} \quad (280)$$

where $\bar{F}_{ab} \equiv \frac{1}{2}[F_{ab}(h_a) + F_{ab}(h_b)]$. The thermal energy equation (270) takes the form

$$\frac{du}{dt} = -\frac{P}{\rho}\nabla \cdot \mathbf{v} + \frac{s^2 t_s}{\rho \rho_g} \nabla P \cdot \nabla u, \quad (281)$$

the first term of which is identical to (37) and the second term of which is discretised in PHANTOM according to

$$-\frac{\rho_a}{2\rho_a^g} \sum_b m_b \frac{s_a s_b}{\rho_a \rho_b} \left(\frac{t_{s,a}}{\rho_a} + \frac{t_{s,b}}{\rho_b} \right) (P_a - P_b) (u_a - u_b) \frac{\bar{F}_{ab}}{|r_{ab}|}. \quad (282)$$

2.14.15 Conservation properties

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

$$\frac{d}{dt} \left(\sum_a \frac{m_a}{\rho_a} s_a^2 \right) = \sum_a m_a \left(\frac{2s_a}{\rho_a} \frac{ds_a}{dt} - \frac{s_a^2}{\rho_a^2} \frac{d\rho_a}{dt} \right) = 0. \quad (283)$$

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 manually cap the dust fraction to unity in the

⁸One 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. Furthermore, we also implemented a check of the SPH results against the analytic expressions in the nightly test suite (see Section 5).

Process	Description	Reference
HI (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 Boulrot 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)
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	Temperature independent 20eV cosmic-ray photons	Goldsmith & Langer (1978)
Photo-electric heating	UV e ⁻ excitation from dust and PAH	Wolfire et al. (2003)

Table 3 Heating and cooling processes in the PHANTOM cooling module.

code which violates the conservation of dust mass. The user should therefore be cautious to check the conservation of dust mass when using the one fluid algorithm.

Total energy with the one fluid scheme can be expressed via

$$E = \sum_a m_a \left[\frac{1}{2} v_a^2 + (1 - \epsilon_a) u_a \right], \quad (284)$$

which is conserved exactly by the discretisation since

$$\sum_a m_a \left[\mathbf{v}_a \cdot \frac{d\mathbf{v}_a}{dt} + \rho_a^g \frac{du_a}{dt} - u_a \left(\frac{2s_a}{\rho_a} \frac{ds_a}{dt} - \frac{s_a^2}{\rho_a^2} \frac{d\rho_a}{dt} \right) \right] = 0. \quad (285)$$

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

2.14.16 Timestep constraint

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

$$\Delta t_{\text{drag}}^a = C_{\text{force}} \frac{h^2}{\epsilon t_s c_s^2}. \quad (286)$$

This becomes prohibitive precisely in the regime where the one fluid method is no longer applicable ($t_s > t_{\text{Cour}}$; 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.

2.15 ISM physics

There is a range of cooling and chemistry included in PHANTOM to capture the thermodynamics of gas in the interstellar medium (ISM).

2.15.1 Cooling function

The cooling in PHANTOM is based on the routines written by Simon Glover (Glover & Mac Low, 2007) updated further by Glover et al. (2010). It includes cooling from atomic lines (HI), molecular excitation (H₂), fine structure metal lines (Si I, Si II, O I, C I, C II), gas-dust collisions and PAH recombination (see Glover & Mac Low 2007 for references on each process). 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 3 summarises these various heating and cooling processes.

These routines were originally adapted for use in SPHNG (Dobbs et al., 2008) and result in an approximate two-phase ISM with temperatures of 100K and 10000K. Note that the cooling depends on a vast array of parameters (dust to gas ratio, cosmic ray ionisation rate, etc.) many of which can be specified at runtime and are set to sensible default values for the Milky Way’s ISM by default. When cooling is used, the timestep is constrained according to Equation (287).

2.15.2 Timestep constraint

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

$$\Delta t_{\text{cool}}^a = C_{\text{cool}} \left| \frac{\Lambda_{\text{cool}}}{\dot{\Lambda}_{\text{cool}}} \right|, \quad (287)$$

where $C_{\text{cool}} = 0.3$ (as in Glover & Mac Low 2007).

2.15.3 Chemical network

A basic chemical network is included for ISM gas that evolves the abundances of H, H⁺, e⁻, and the molecules H₂ and CO. The number density of each species, n_X , is evolved using simple rate equations of the form:

$$\frac{dn_X}{dt} = C_X - D_X n_X, \quad (288)$$

Reaction	Description	Reference
$\text{H}^+ + \text{e}^- + \text{grain} \rightarrow \text{H} + \text{grain}$	Grain surface recombination	Weingartner & Draine (2001)
$\text{H}^+ + \text{e}^- \rightarrow \text{H} + \gamma$	Gas-phase recombination	Ferland et al. (1992)
$\text{H} + \text{e}^- \rightarrow \text{H}^+ + 2\text{e}^-$	e^- collisional ionisation	Abel et al. (1997)
$\text{H} + \text{c.r.} \rightarrow \text{H}^+ + \text{e}^-$	Cosmic ray ionisation	Glover & Mac Low (2007)
$\text{H} + \text{H} + \text{grain} \rightarrow \text{H}_2 + \text{grain}$	Grain surface formation	Bergin et al. (2004)
$\text{H}_2 + \gamma \rightarrow 2\text{H}$	UV photodissociation	Draine & Bertoldi (1996)
$\text{H}_2 + \text{c.r.} \rightarrow \text{H}_2^+ + \text{e}^-$	Cosmic ray ionisation*	Bergin et al. (2004)
$\text{C}^+ + \text{H}_2 \rightarrow \text{CH}_2^+ + \gamma$	Radiative association	Nelson & Langer (1997)
$\text{CH}_2^+ + \text{various} \rightarrow \text{CH}_X + \text{various}$	Rapid neutralisation [†]	-
$\text{CH}_X + \text{O I} \rightarrow \text{CO} + \text{H}_X$	Gas phase formation	-
$\text{CH}_X + \gamma \rightarrow \text{C} + \text{H}_X$	UV photodissociation	-
$\text{CO} + \gamma \rightarrow \text{C}^+ + \text{O} + \text{e}^-$	UV photodissociation ^{††}	-

Table 4 Processes and references for the Phantom ISM chemistry module tracing the evolution of H, H₂ and CO.

* H₂⁺ ions produced by cosmic ray ionisation of H₂ are assumed to dissociatively recombine to H + H, so that the effective reaction in the code is actually H₂ + c.r. → H + H.

† Process is intermediate and is assumed rather than fully represented.

†† C is not present in our chemistry, but is assumed to very rapidly photoionise to C⁺.

so that the density of each species (X) at the next timestep $t + \Delta t$ is

$$n_X(t + \Delta t) = n_X(t) + \frac{dn_X}{dt} \Delta t, \quad (289)$$

where C_X and D_X are creation and destruction coefficients for each species and are functions of density, temperature, and abundances of other species. There are in effect only three species to evolve (H, H₂ and CO) as the H⁺ and e⁻ abundances are directly related to the H abundance. A chemical time-stepping criterion is utilised, where the chemistry routine is subcycled during the main time step at the interval Δt_{chem} . H chemistry is evolved on the cooling timestep since the chemical timescale on which the H⁺ abundance changes significantly is generally comparable to or longer than the cooling time. This is not true for H₂ and CO which are further subcycled. If the abundance is decreasing then the chemical timestep is

$$\Delta t_{\text{chem}} = -0.1 n_X / (C_X - D_X n_X), \quad (290)$$

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

$$\Delta t_{\text{chem}} = 0.005 \Delta t_{\text{hydro}}, \quad (291)$$

which 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.

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. H₂ chemistry is based on the model of Bergin et al. (2004), with H₂ 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 H₂ (where Z encompasses many similar type species). CO and CH_Z are subject to independent photodestruction, 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, H₂ and CO chemistry are given in Table 4, with relevant references for each process.

2.16 Particle injection

PHANTOM includes modules for injecting material into an existing simulation in a variety of ways (see Toupin et al. 2015b,a for recent applications). Modules currently exist for 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

PHANTOMSETUP is a standalone utility to set up initial conditions on the particles. Like many of the other utilities, for the user this simply means writing a plug-in ‘module’ that is called by PHANTOMSETUP to generate the initial conditions. The more advanced setup modules come with their own parameter files (.setup) such that generating initial conditions with different parameters is straightforward. Utility routines are provided for the more advanced tasks, described below.

3.1 Uniform distributions

The simplest setup utility simply sets particles in 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 3 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 requires 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 e.g. for setting up stars in hydrostatic equilibrium (Section 3.3).

3.2 Accretion discs

3.2.1 Particle placement

The accretion disc setup module uses a Monte-Carlo particle placement in cylindrical geometry to construct density profiles of the form

$$\rho(x, y, z) = \Sigma_0 f_s \left(\frac{R}{R_{\text{in}}} \right)^{-p} \exp \left(\frac{-z^2}{2H^2} \right), \quad (292)$$

where Σ_0 is the surface density at $R = R_{\text{in}}$ (if $f_s = 1$), $H \equiv c_s/\Omega$ is the scale height (with $\Omega \equiv \sqrt{GM/R^3}$) and $f_s \equiv (1 - \sqrt{R_{\text{in}}/R})$ is an optional factor to smooth the surface density near the inner disc edge.

We first choose the azimuthal angle as a uniform random deviate $u_1 \in [\phi_{\text{min}}, \phi_{\text{max}}]$ ($0 \rightarrow 2\pi$ 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 $u_2 \in [0, f_{\text{max}}]$ and iterating

until we find a random number that satisfies

$$u_2 < f, \quad (293)$$

where $f \equiv R\Sigma = R^{1-p}$ and $f_{\text{max}} = R_{\text{in}}^{1-p}$ (or $f_{\text{max}} = R_{\text{out}}^{1-p}$ if $p \leq 1$). Finally the z position is chosen with a third random number $u_3 \in [-z_{\text{max}}, z_{\text{max}}]$ such that $u_3 < g$, where $g = \exp[-z^2/(2H^2)]$ and $z_{\text{max}} = \sqrt{6}H$.

Several authors have argued that a more uniform particle placement is preferable for setting up discs in SPH (Cartwright et al., 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). However, the initial random distribution can cause problems with the one fluid dust method (Section 2.14.14) so it is recommended that when simulating discs with small dust grains the user should first relax the gas disc into equilibrium before adding dust.

3.2.2 Velocity field

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

$$v_\phi^2 = \frac{GM}{R} - f_p - 2v_\phi f_{\text{BH}}, \quad (294)$$

where the correction from radial pressure gradients is given by

$$f_p = -c_s^2(R) \left(\frac{3}{2} + p + q + \frac{1}{2f_s} \right), \quad (295)$$

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

$$f_{\text{BH}} = -\frac{2a}{c^3} \left(\frac{GM}{R} \right)^2, \quad (296)$$

where a is the black hole spin parameter. The latter assumes Lense-Thirring precession is implemented as in Section 2.5.5. Where self-gravity is used M is the enclosed mass⁹ at a given radius $M(< R)$, otherwise it is simply the mass of the central object. Equation (294) is a quadratic for v_ϕ which is solved analytically.

3.2.3 Warps

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

$$x' = x \cos(i) + z \sin(i), \quad (297)$$

$$y' = y, \quad (298)$$

$$z' = -x \sin(i) + z \cos(i), \quad (299)$$

⁹Using the enclosed mass is an approximation because the disc is not spherically symmetric but the difference is small and as a result the disc relaxes quickly into the true equilibrium.

and velocities are similarly adjusted using

$$v'_x = v_x \cos(i) + v_z \sin(i), \quad (300)$$

$$v'_y = v_y, \quad (301)$$

$$v'_z = -v_x \sin(i) + v_z \cos(i). \quad (302)$$

3.2.4 Setting the disc viscosity

As described in Section 2.7.1, the simplest approach to mimicking an α -disc viscosity in SPH is to perform some slight modifications to the shock viscosity term and set the desired α_{SS} according to Equation (124). Since the factor $\langle h \rangle / H$ is dependent both on resolution and temperature profile (i.e. the q -index), the `set_disc` routine computes this during the setup, taking the desired α_{SS} as input and writing the required α_{AV} to the input file on output. Although this does not guarantee that α_{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 for the user to prescribe the disc viscosity.

3.3 Stars and binary stars

PHANTOM contains a general method for setting up ‘realistic’ stellar profiles, based on either analytic functions (e.g. for setting up 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 method is to set up a uniform density sphere of particles and set the density profile by stretch mapping (see below). 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 equation of state. We recommend then relaxing the star into equilibrium for several dynamical times using a damping parameter (Section 3.6) before setting it on an appropriate orbit. Setting the star, or multiple stars, into orbit is usually achieved using a plugin for the `phantommoddump` utility which reads PHANTOM data files and passes the particle information to a user-defined subroutine for modification before writing the data to a new file. For example, the `moddump_binary` module can be used to place the relaxed star into a binary orbit with a sink particle.

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 but instead set a softening length for the sink particle consistent with the original core radius (see Section 2.9.1).

3.4 Stretch mapping

PHANTOMSETUP allows the set up of general non-uniform density profiles using ‘stretch mapping’. The

procedure for spherical distributions is outlined by [Herant \(1994\)](#) and has been used by numerous authors (e.g. [Fryer et al., 2007](#); [Rosswog & Price, 2007](#); [Rosswog et al., 2009](#)). It can be generalised to any density profile that is along one coordinate direction (e.g. [Price & Monaghan, 2004b](#)). Starting with particles setup 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

$$f(x) = \frac{M(x)}{M(x_{\text{max}})} - \frac{x_0 - x_{\text{min}}}{(x_{\text{max}} - x_{\text{min}})} = 0, \quad (303)$$

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

$$M(x) \equiv \int_{x_{\text{min}}}^x \rho(x') dS(x') dx', \quad (304)$$

where the area element $dS(x')$ depends on the geometry and the chosen direction, given by

$$dS(x) = \begin{cases} 1 & \text{cartesian or cyl./sph. along } \phi, \theta \text{ or } z \\ 2\pi x & \text{cylindrical along } r \\ 4\pi x^2 & \text{spherical along } r \end{cases} \quad (305)$$

We solve (303) for each particle using Newton-Raphson iterations

$$x = x - \frac{f(x)}{f'(x)}, \quad (306)$$

where

$$f'(x) = \frac{\rho(x) dS(x)}{M(x_{\text{max}})}. \quad (307)$$

and we iterate until $|f(x)| < 10^{-9}$. The Newton-Raphson iterations can fail to converge in extreme cases, so if the root-finding has failed to converge after 30 iterations then we revert to a bisection method.

This is implemented in the code in a very general way so that the standard routines for setting up uniform particle distributions (Section 3.1) need only be modified by an optional argument specifying the desired density function, which can be either as a function or a tabulated dataset. Since the mass integral (Equation 304) may not be known analytically, we compute this numerically using a trapezoidal rule integration of the density function.

The disadvantage of stretch mapping is that in spherical or cylindrical geometry it produces defects in the particle distribution 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, e.g. by simulating the star in isolation with artificial damping added. Alternative approaches are i) to relax the simulation using an external potential chosen to produce the desired density profile in equilibrium (e.g. [Zurek & Benz 1986](#);

Nagasawa et al. 1988) or ii) to iteratively ‘cool’ the particle distribution to generate ‘optimal’ initial conditions (Diehl et al., 2015).

3.5 Galactic initial conditions

In addition to simulating ISM gas in galactic discs with analytic stellar potentials, there is also the capability to use resolved bulge-halo-disc components which are represented by collisionless N -body particles (see Section 2.13.3). To replace a potential with a resolved system requires great care be taken with the initial conditions (i.e. position, velocity, 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 a separate initial conditions generating routine with PHANTOM, which was used to initialise 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 in PHANTOM and care must be taken to ensure the mass and scale lengths correspond to the rotation curve used to generate the initial conditions. If the user wishes to remove the static halo, we direct them to one of the many codes in the literature to seed multi-component N -body disc galaxies (e.g. Kuijken & Dubinski 1995a; Boily et al. 2001; McMillan & Dehnen 2007; Yurin & Springel 2014). If the user wishes to use MAGALIE (Boily et al., 2001) or GALIC (Yurin & Springel, 2014) we also include routines to convert the outputs of these codes into formats readable by the PHANTOM `galdisc` setup routine.

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 et al., 2005; Wang et al., 2010); this may be included in a future update.

3.6 Damping

To relax the particle distribution into equilibrium PHANTOM adopts the standard approach (e.g. Gingold & Monaghan, 1977) of adding a damping parameter, f_d (called `damp` in the input file) such that a percentage of the kinetic energy is removed each timestep by adding an external acceleration in the form

$$\mathbf{a}_{\text{ext,damp}}^a = -f_d \mathbf{v}, \quad (308)$$

Sensible values for f_d are of order a few percent (e.g. $f_d = 0.03$) such that a small fraction of the kinetic energy of the particles is removed each timestep.

4 Numerical tests

Unless otherwise stated we use the M_6 quintic spline kernel with $h_{\text{fac}} = 1.0$ 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 $h_{\text{fac}} = 1.2$ (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, and hence it should be straightforward for even a novice user to reproduce any of the results shown here.

When referencing errors we refer to L_1 and L_2 norms computed according to

$$L_1 \equiv \frac{1}{N} \sum_{i=1}^N |y_i - y_{\text{exact}}|, \quad (309)$$

$$L_2 \equiv \sqrt{\frac{1}{N} \sum_{i=1}^N |y_i - y_{\text{exact}}|^2}, \quad (310)$$

where y_{exact} is the exact or comparison solution interpolated or computed at the location of each particle i . These are computed using SPLASH (Price, 2007).

4.1 Hydrodynamics

4.1.1 Sod shock tube

Figure 12 shows the results of the standard Sod (1978) shock tube test, performed in 3D using $[\rho, P] = [1, 1]$ in the ‘left state’ ($x \leq 0$) and $[\rho, P] = [0.125, 0.1]$ for the ‘right state’ ($x > 0$) with a discontinuity initially at $x = 0$ and zero initial velocity and magnetic field. We perform the test using an adiabatic equation of state 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. Hubber et al. 2011; 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,a; 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

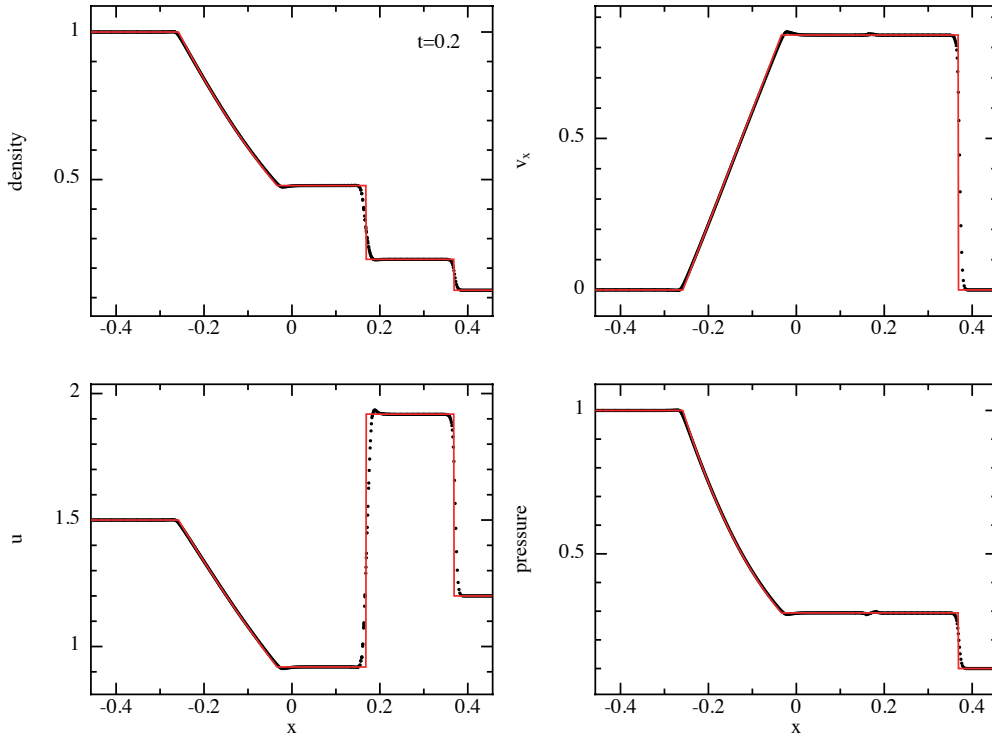


Figure 12. Results of the Sod shock tube test in 3D, showing projection of all particles (black dots) compared to the analytic solution (red line). The problem is set up with $[\rho, P] = [1, 1]$ for $x \leq 0$ and $[\rho, P] = [0.125, 0.1]$ for $x > 0$ with $\gamma = 5/3$, with zero velocities and no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

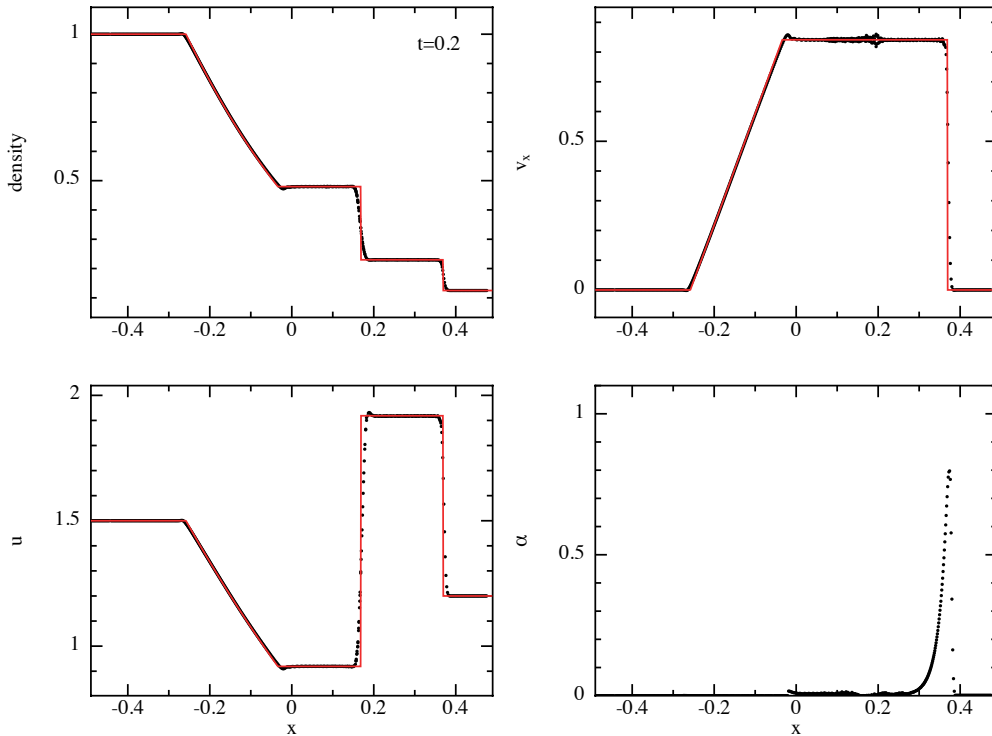


Figure 13. As in Figure 12, but with code defaults for all dissipation terms, which provide second order convergence away from shocks. The lower right panel in this case shows the spatial variation of the viscosity parameter α^{AV} .

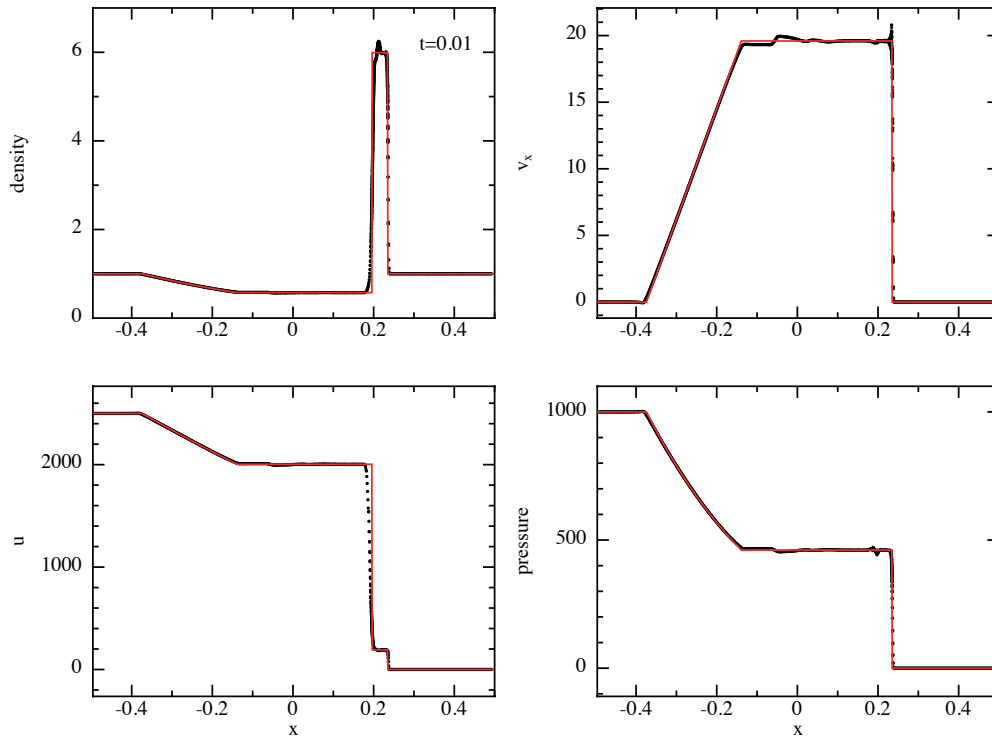


Figure 14. Results of the 3D blast wave test, showing projection of all particles (black dots) compared to the analytic solution (red line). The problem is set up with $[\rho, P] = [1, 1000]$ for $x \leq 0$ and $[\rho, P] = [1, 0.1]$ for $x > 0$ with $\gamma = 7/5$, with zero velocities and no magnetic 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 $\alpha^{AV} = 1$

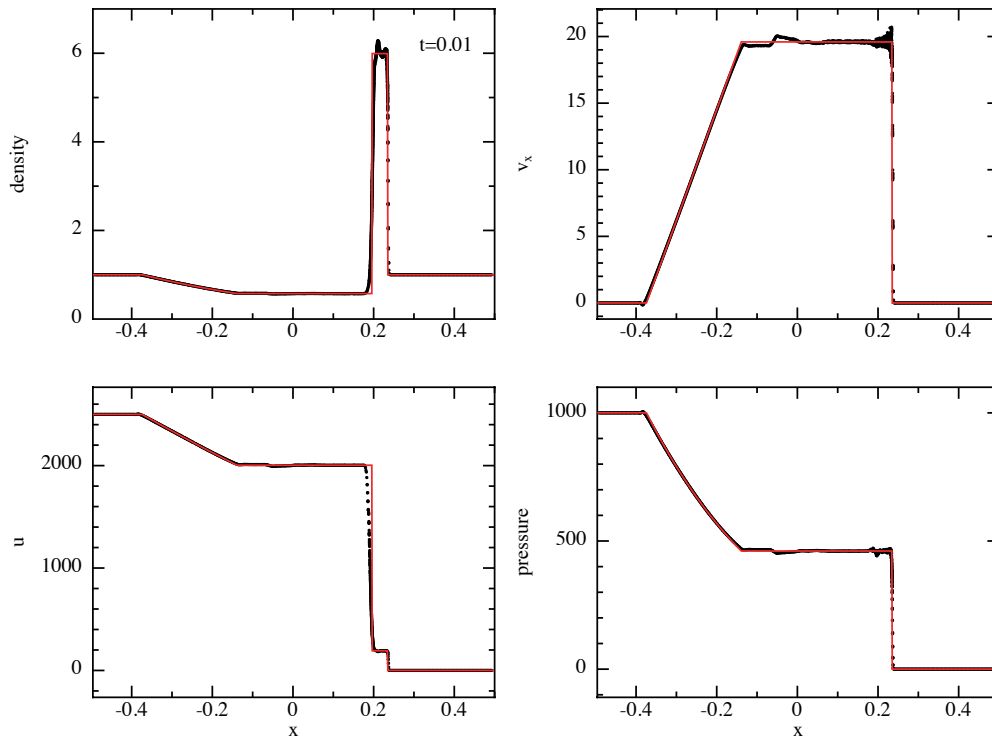


Figure 15. As in Figure 14 but with code defaults for dissipation switches.

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 12 use $256 \times 24 \times 24$ particles initially in $x \in [-0.5, 0]$ and $128 \times 12 \times 12$ particles in $x \in [0, 0.5]$ with code defaults for the artificial conductivity ($\alpha_u = 1$ with v_{sig}^u given by Equation 41) and artificial viscosity ($\alpha^{\text{AV}} = 1, \beta^{\text{AV}} = 2$). The results are identical whether global or individual particle timesteps (Section 2.4.4) are used. Figure 13 shows the results when code defaults for viscosity are also employed, resulting in a time-dependent α_{AV} (see lower right panel). There is slightly more noise in the velocity field around the contact discontinuity in this case but otherwise the results are identical.

4.1.2 Blast wave

As a more extreme version of the shock test, Figure 14 shows the results of the blast wave problem from Monaghan (1997), set up initially with $[\rho, P] = [1, 1000]$ for $x \leq 0$ 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 $800 \times 12 \times 12$ particles in the domain $x \in [-0.5, 0.5]$. Results are shown at $t = 0.01$. This is a much more challenging problem for the code than the Sod test due to the higher Mach number, but the close match to the analytic solution demonstrates that the conservation properties are satisfied. We employ code defaults for the artificial viscosity and conductivity with $\alpha \in [0, 1]$ and $\alpha_u = 1$ with Equation (41) for signal speed in the artificial conductivity. The main noticeable error is that the contact discontinuity appears a little over-smoothed due to the artificial conductivity, while there is a small amount of noise seen in the post-shock velocity profile due to the viscosity switch (which as in the previous test can be reduced using a constant α^{AV}).

4.1.3 Sedov blast wave

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

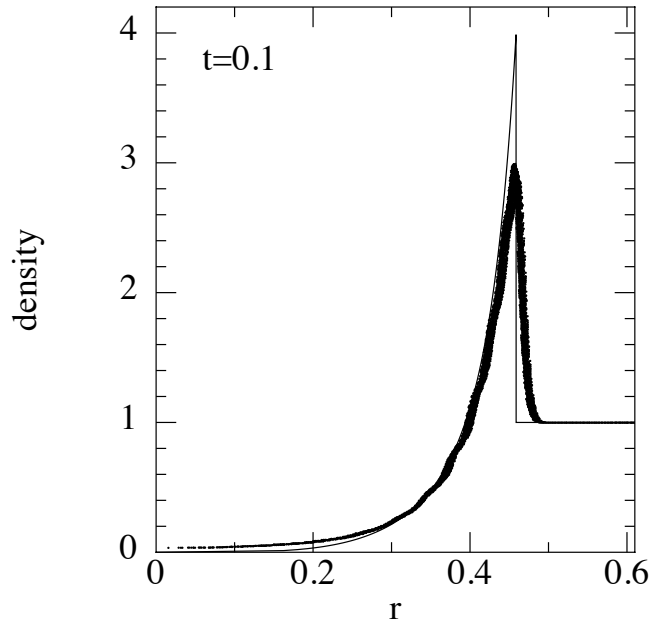


Figure 16. Density as a function of radius in the Sedov blast wave problem, showing all particles (black dots) compared to the analytic similarity solution given by the solid line. Results are shown using $100 \times 100 \times 100$ particles placed initially on a cubic lattice.

wog & 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 equation of state with $\gamma = 5/3$. The total energy is normalised such that the total thermal energy in the blast is $E_0 = \sum_a m_a u_a = 1$, distributed on the particles within $r < R_{\text{kern}} h_0$ using the smoothing kernel, i.e.

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

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.

Figure 16 shows the results of the Sedov test for a simulation with 100^3 particles, showing density as a function of radius for all particles, compared to the analytic solution given by the solid line. The accuracy of the shock position is a reflection of the degree to which the total energy is conserved. Given the initial strong gradient in thermal energy, artificial conductivity is also important for reducing the noise on this problem, as first noted by Rosswog & Price (2007).

4.1.4 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;

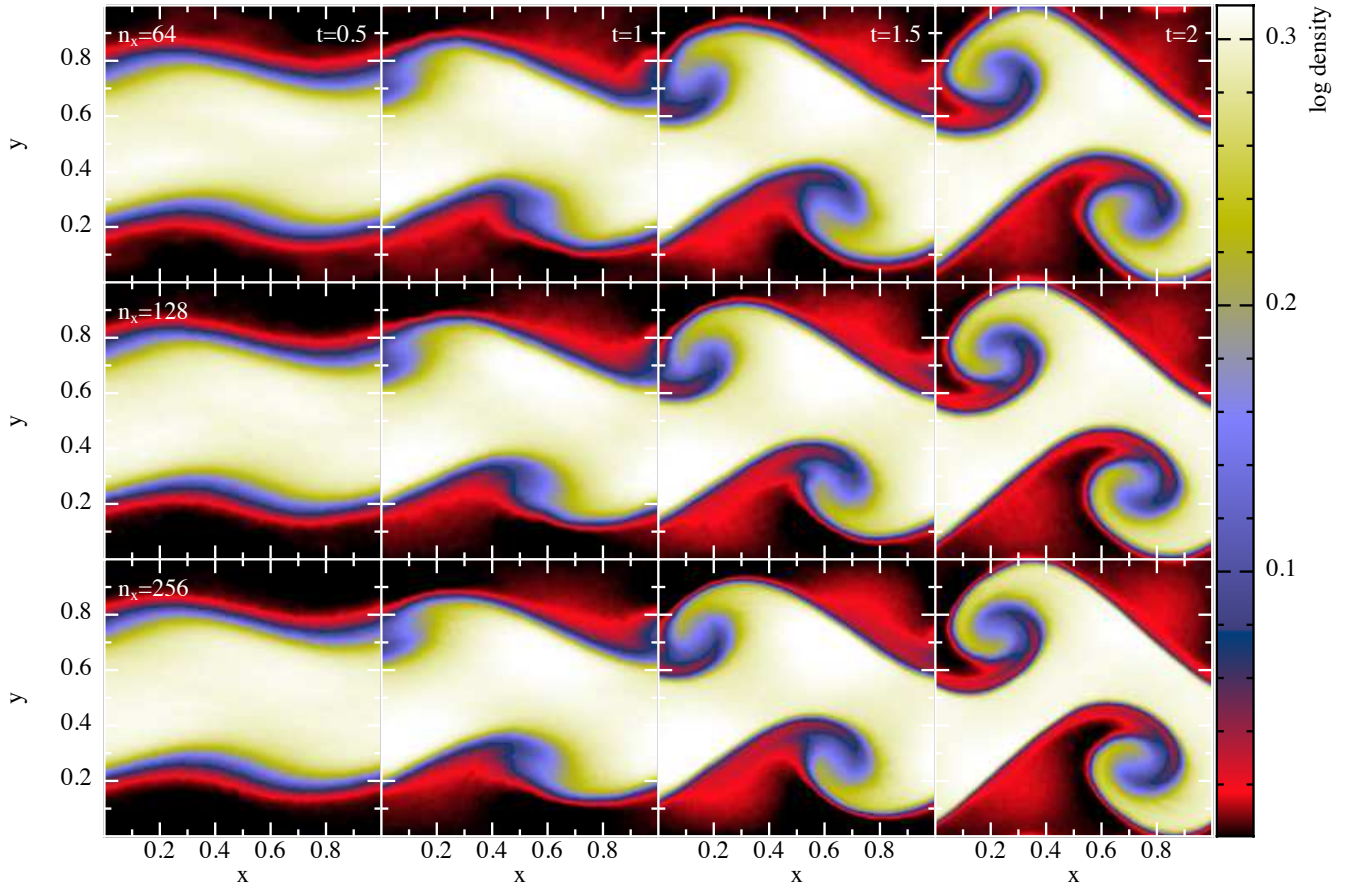


Figure 17. Results of the well-posed Kelvin Helmholtz instability test from Robertson et al. (2010), shown at a resolution of (from top to bottom) $64 \times 74 \times 12$, $128 \times 148 \times 12$ and $256 \times 296 \times 12$ equal mass SPH particles. We use stretch mapping (Section 3.4) to achieve the initial density profile, consisting of a 2:1 density jump with a smoothed transition.

Hubber et al., 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, in the sense that the results will always change with numerical resolution, since the Kelvin-Helmholtz instability grows fastest at small scales in the absence of physical dissipation or other regularising forces such as magnetic fields or surface tension. The ill-posed nature of the test problem has been pointed out by numerous authors (Robertson et al., 2010; McNally et al., 2012; Lecoanet et al., 2016) who have each proposed well-posed alternatives. Here 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, so that the density and shear velocity in the y direction are given by

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

and

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

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

$$R(y) \equiv [1 - f(y)][1 - g(y)], \quad (314)$$

where

$$f \equiv \frac{1}{1 + \exp[2(y - 0.25)/\Delta]},$$

$$g \equiv \frac{1}{1 + \exp[2(0.75 - y)/\Delta]}, \quad (315)$$

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

$$v_y = 0.1 \sin(2\pi x). \quad (316)$$

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 three dimensional 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 den-

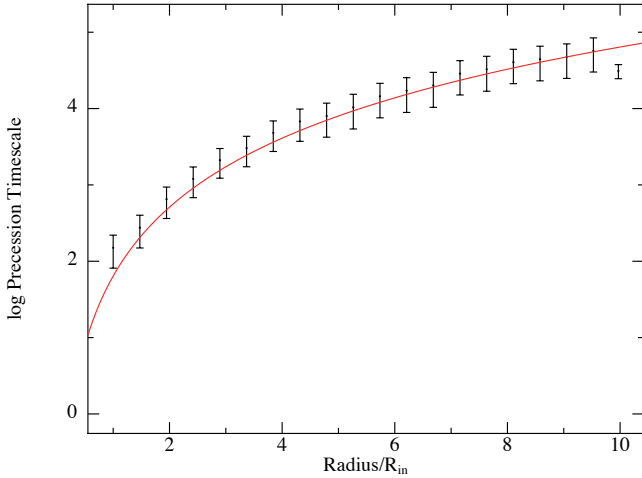


Figure 18. Lense-Thirring precession test from a disc inclined by 30° . Here the precession time-scale is measured from the cumulative twist in the disc and the exact solution, $t_p = R^3/2a$ is represented by the red line.

sity profile by stretch-mapping in the y direction using equation (312) as the input function (c.f. Section 3.4).

Figure 17 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). It may be seen that SPH converges slower than the results shown in that paper, with convergence achieved only at $n_x = 256$, compared to $n_x = 128$ for the results shown from the ENZO code in Robertson et al. (2010). 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 2.3.9).

4.2 External forces

4.2.1 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 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 18 shows the precession timescale measured from the simulation as a function of the radius compared to the analytically derived pre-

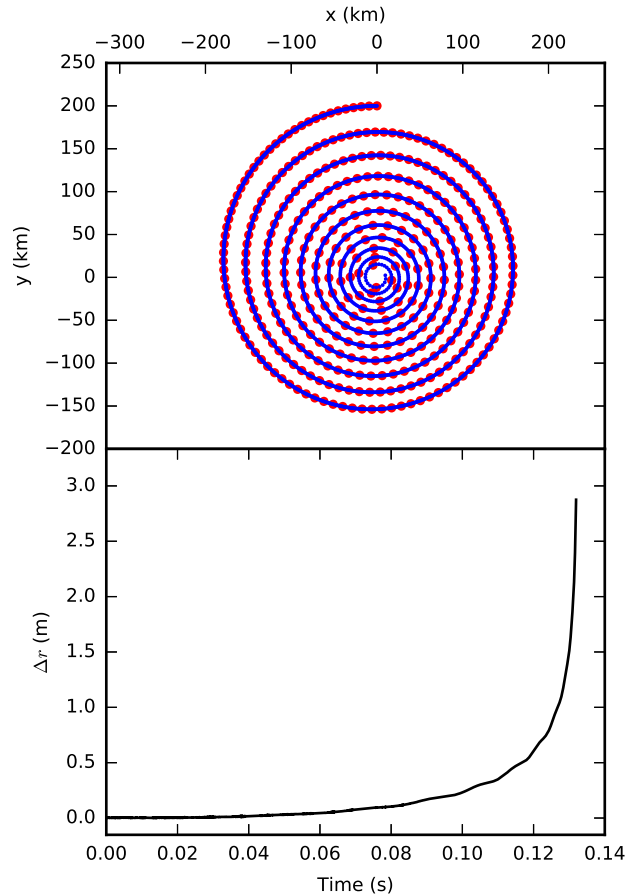


Figure 19. Inspiral of a test particle subjected to Poynting-Robertson drag using PHANTOM (blue curve) compared to the expected solution (red curve). For clarity, only some of the points are plotted. The results are indistinguishable on this scale. Bottom panel shows the distance (Δr) between the PHANTOM particle and the test code particle. This demonstrates that the implementation of Poynting-Robertson drag in PHANTOM is correct.

cession timescale, with uncertainties derived from the calculation of the gradient.

4.2.2 Poynting-Robertson drag

Figure 19 shows the trajectory of a spherical assembly of 89 pressureless SPH particles subject to Poynting-Robertson drag with a fixed value of $\beta_{\text{PR}} = 0.1$, assuming a central neutron star of mass $1.4M_\odot$ 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 4th order Runge Kutta scheme. As shown in Figure 19, 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.

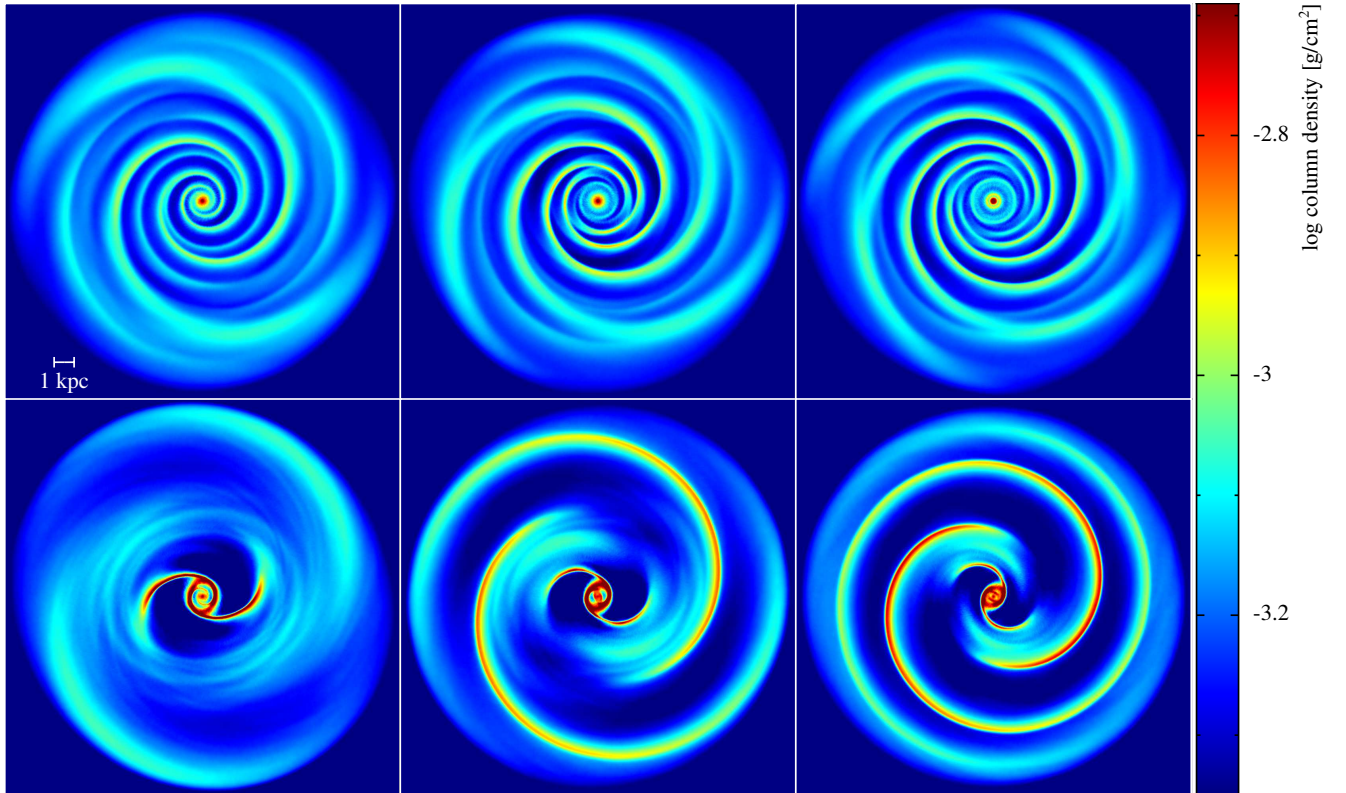


Figure 20. Gas in a galactic disc under the effect of different galactic potentials. Top row shows models with a 2, 4, and 6 armed spiral (left to right) with a pitch angle of 15° and pattern speed of $20 \text{ km s}^{-1} \text{ kpc}^{-1}$. Bottom row shows a bar potential with pattern speeds of 40, 60 and $80 \text{ km s}^{-1} \text{ kpc}^{-1}$ (left to right).

4.2.3 Galactic potentials

Figure 20 shows 6 calculations with gas embedded within different galactic potentials (Section 2.5.4). We set up an isothermal gas disc with $T = 10,000 \text{ K}$, with a total gas mass of $1 \times 10^9 M_\odot$ 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+disc+halo, the same as Pettitt et al. 2014). The left column shows gas exposed to spiral potentials of Cox & Gómez (2002) with three different arm numbers (2, 3, 4), while the left hand column simulations within the bar potential of Wada & Koda (2001) at three different pattern speeds ($40, 60$ and $80 \text{ km s}^{-1} \text{ kpc}^{-1}$). 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.

4.3 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 et al. (1996) showed how to use the SPH artificial viscosity term to mimic a Shakura & Sunyaev (1973) disc viscosity.

4.3.1 Calibration of disc viscosity

The simplest test is to calibrate the disc viscosity by measuring the diffusion rate of the disc surface density. Figure 21 shows the results of an extensive study of this with PHANTOM performed by Lodato & Price (2010). For this study we set up a disc from $R_{\text{in}} = 0.5$ to $R_{\text{out}} = 10$ with surface density profile

$$\Sigma = \Sigma_0 R^{-p} \left(1 - \sqrt{\frac{R_{\text{in}}}{R}} \right), \quad (317)$$

and a locally isothermal equation of state $c_s = c_{s,0} R^{-q}$. We set $p = 3/2$ and $q = 3/4$ such that the disc is uniformly resolved, i.e. $h/H \sim \text{const}$ (Lodato & Pringle, 2007), giving a constant value of the Shakura & Sunyaev (1973) α parameter according to Equation 124. We set $c_{s,0}$ such that the aspect ratio $H/R = 0.02$ at $R = 1$. We used 2 million particles by default, with several addi-

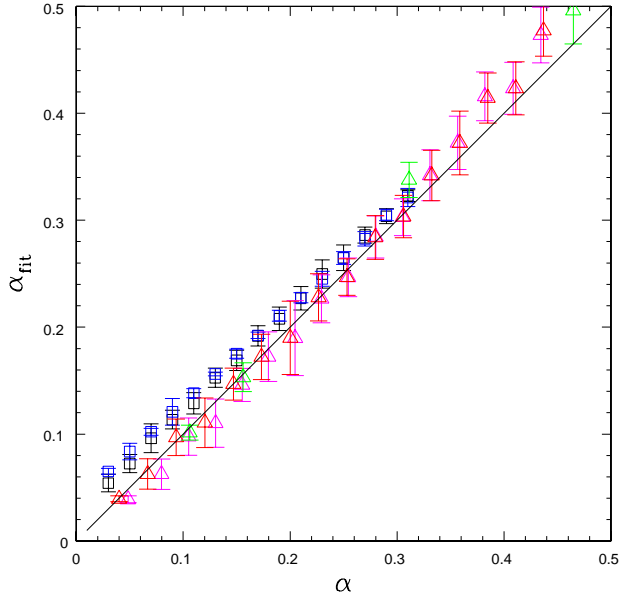


Figure 21. Calibration of the disc viscosity in PHANTOM, comparing the input value of the Shakura-Sunyaev α from Equation (124) (x -axis) with the measured diffusion rate of the surface density by fitting to a 1D code (y -axis). Triangles indicate simulations with the disc viscosity computed using the artificial viscosity (Section 2.7.1) while squares represent simulations using physical viscosity (Section 2.8.1). 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).

tional calculations performed using 20 million particles. We perform the simulation to $t = 1000$ in code units.

We measured the diffusion rate 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) — we use a Newton-Raphson iteration 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 (α_{fit}) and error bars. Figure 21 shows that the measured diffusion rates agree with the expected values to within the error bars. The exception is for very 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 2.7.1) while squares represent simulations with physical viscosity set according to Equation (126).

This test demonstrates that the implementation of disc viscosity in PHANTOM works, and also the translation of the artificial viscosity term according to Equation (124) is correct.

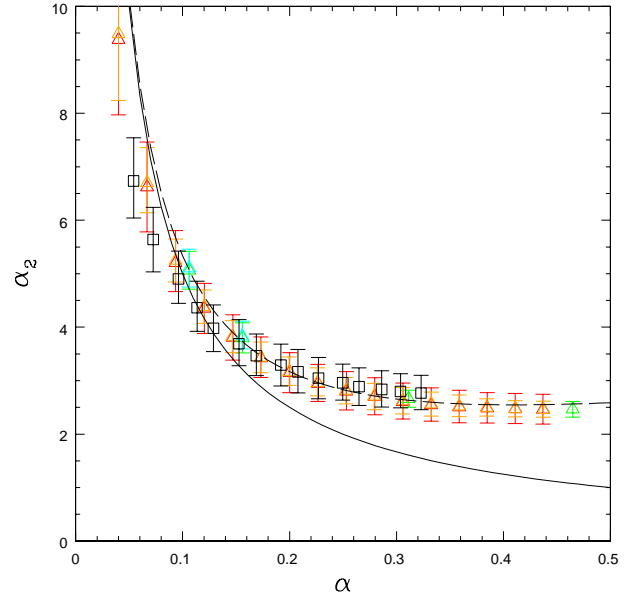


Figure 22. Warp diffusion rate as a function of disc viscosity, showing the PHANTOM results compared to the non-linear theory of Ogilvie (1999) (dashed line). The linear prediction $\alpha_2 = 1/(2\alpha)$ is shown by the solid line. Colouring of points is as in Figure 21. Figure taken from Lodato & Price (2010).

4.3.2 Warp diffusion

A more demanding test of disc physics involves the dynamics of warped discs. Here an 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 et al., 2002).

PHANTOM was born to simulate warped discs — with our first study in Lodato & Price (2010) designed to test the Ogilvie (1999) theory in 3D simulations. Figure 22 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

$$\alpha_2 = \frac{1}{2\alpha} \frac{4(1 + 7\alpha^2)}{4 + \alpha^2}. \quad (318)$$

Significantly, the PHANTOM results show a measurable difference between this 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 22, 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,b) 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).

4.3.3 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 23 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).

Several 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 $r = 2.5$ using 1 million gas 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 (Dipierro et al., 2015, 2016; Ragusa et al., 2016, 2017), for this test we thus employed the fixed binary potential (Section 2.5.2) 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

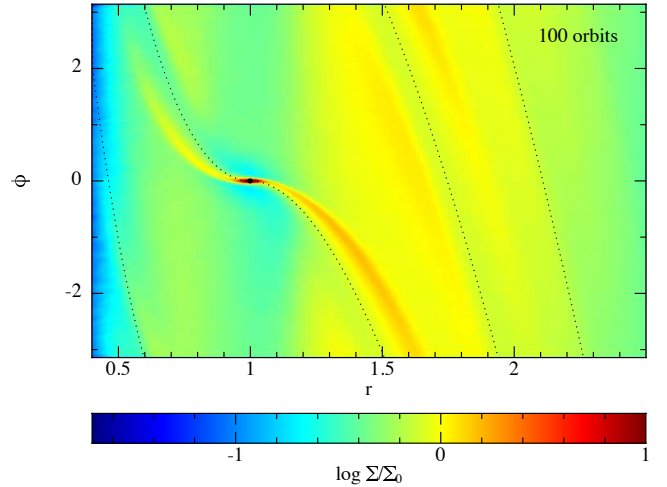


Figure 23. Planet-disc interaction in 3D, showing the ‘viscous Jupiter’ calculation comparable to the 2D results shown with various grid and SPH codes in Figure 10 of de Val-Borro et al. (2006). The dotted lines shows the estimated position of the planetary shocks from Ogilvie & Lubow 2002.

from the planet,

$$\phi_{\text{planet}} = -\frac{m_{\text{planet}}}{\sqrt{r^2 + \epsilon^2}}, \quad (319)$$

where $\epsilon = 0.6H$. We also implemented the prescribed increase of planet mass with time for the first 5 orbits according to

$$m_{\text{planet}} = \sin^2\left(\frac{\pi t}{10P_{\text{orbit}}}\right), \quad (320)$$

where P_{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 equation of state $c_s \propto R^{-0.5}$ such that $H/R \approx 0.05 \approx \text{const}$. We assumed an initially constant surface density $\Sigma_0 = 0.002M_*/(\pi a^2)$. 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 $\alpha_{\text{AV}} \in [0, 1]$ and $\beta_{\text{AV}} = 2$. We use the Navier-Stokes viscosity implementation described in Section 2.8.2, which is the code default. We also employed the cubic spline kernel (with $h_{\text{fac}} = 1.2$) rather than the quintic since we found the latter unnecessary.

Despite the different assumptions, the results in Figure 23 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 slightly 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 very poorly on this test, for reasons unclear. This may be simply due to the low resolution employed (the two SPH calculations used 250,000 and 300,000 particles, respectively but the calculations were performed only in 2D rather than 3D), but the extent of the differences suggest that other improvements in the algorithms used in PHANTOM compared to these older codes are also important.

As per the original comparison, Figure 23 shows the estimated position of the planetary shocks from Ogilvie & Lubow (2002) plotted as dotted lines, namely

$$\phi(r, t) = \begin{cases} t - \frac{2}{3\varepsilon} \left(r^{3/2} - \frac{3}{2} \ln r - 1 \right); & r > r_{\text{planet}}; \\ t + \frac{2}{3\varepsilon} \left(r^{3/2} - \frac{3}{2} \ln r - 1 \right); & r < r_{\text{planet}}, \end{cases} \quad (321)$$

where $\varepsilon = 0.05$ is the disc aspect ratio.

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 Equation (45) we found the shock viscosity switch would simply trigger to $\alpha_{\text{AV}} \approx 1$. The original Morris & Monaghan (1997) switch (Section 2.3.9) 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.

4.4 Physical viscosity

4.4.1 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 equation of state is used ($P = c_s^2 \rho$) with speed of sound $c_s = 1$. 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 24 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.

4.5 Sink particles

4.5.1 Binary orbit

Figure 25 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 around their common centre of mass, a widely used test for N -body integrators (e.g. Hut et al., 1995; Quinn

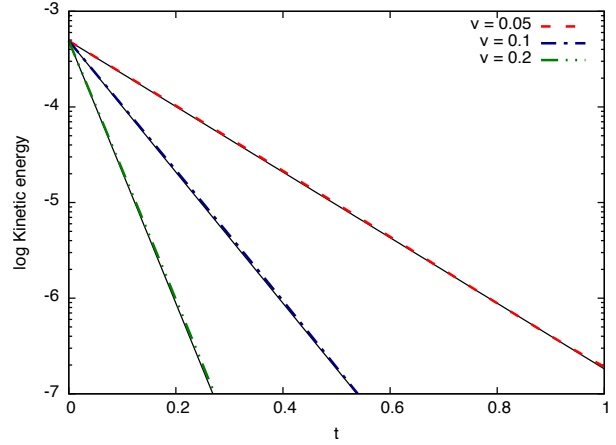


Figure 24. Test of physical Navier-Stokes viscosity in the Taylor-Green vortex using kinematic shear viscosity $\nu = 0.05, 0.1, 0.2$. The exponential decay rate of kinetic energy may be compared to the analytic solution in each case (solid black lines), demonstrating that the calibration of physical viscosity in PHANTOM is correct.

et al., 1997; Farr & Bertschinger, 2007; Dehnen & Read, 2011). We set up the problem by fixing the initial semi-major axis $a = 1$ with masses $m_1 = m_2 = 0.5$ and positions set using the `set_binary` subroutine in PHANTOM, which places the two sinks initially at periastron $\mathbf{x}_1 = [-m_2/(m_1 + m_2)\Delta, 0, 0]$ and $\mathbf{x}_2 = [m_1/(m_1 + m_2)\Delta, 0, 0]$ where $\Delta = a(1 - e)$ is the initial separation. The corresponding initial velocities are $\mathbf{v}_1 = [0, -m_2/(m_1 + m_2)|v|, 0]$ and $\mathbf{v}_2 = [0, m_1/(m_1 + m_2)|v|, 0]$, where $|v| = \sqrt{a(1 - e^2)(m_1 + m_2)/\Delta}$. The 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).

We plot the corresponding energy error as a function of time for the first 1000 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 1000 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 Equation (73), 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_{force} from the default setting of 0.25 to obtain more accurate orbital dynamics.

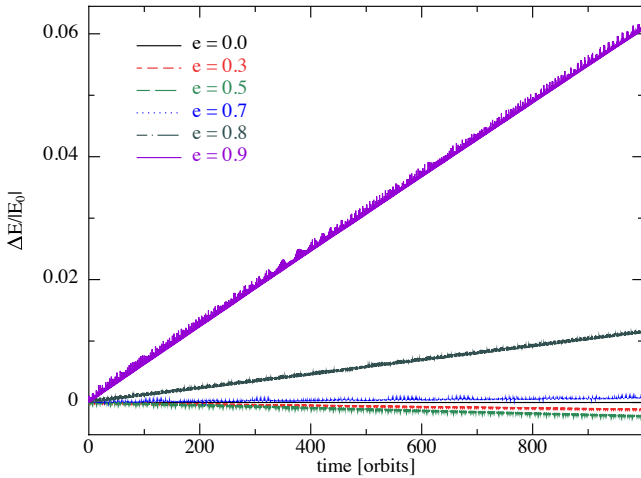


Figure 25. Errors in energy conservation in a sink particle binary integration with code default parameters for the timestep control, showing the energy drift caused by the adaptive timestepping. Angular momentum is conserved to machine precision.

In addition to calibrating the timestep constraint, Figure 25 also validates the sink particle substepping via the RESPA algorithm (Section 2.4.3) 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 $\Delta t_{\max} = 1$ 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.

4.5.2 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. To set up this problem we set up a single sink particle with $\mathbf{x} = [0, 0.0580752367, 0]$ and $\mathbf{v} = [0.489765446, 0, 0]$. We turn on the time-dependent binary potential as described in Section 2.5.2 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 $r_{\text{acc}} = 0.1$ although both are irrelevant to the problem.

Figure 26 shows the resulting ‘Chinese coin’ 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

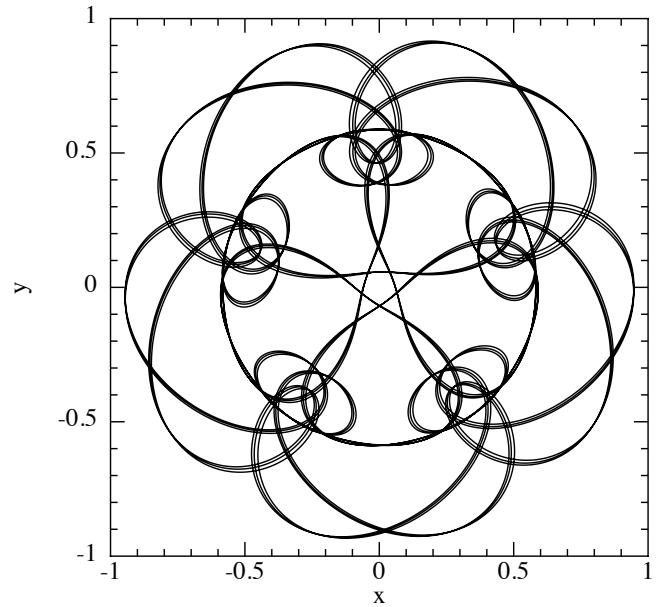


Figure 26. ‘Chinese coin’ orbit of a sink particle in the restricted 3-body problem from Chin & Chen (2005) using default code parameters, testing both the time-integrator for sink particles and the interaction with a time-dependent binary potential (Section 2.5.2). The trajectory of the sink is plotted every timestep for 3 periods ($t = 27\pi$).

only a second-order integrator, the orbital trajectory is remarkably accurate, showing 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).

4.6 Magnetohydrodynamics

4.6.1 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 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 $\sin b = 2/\sqrt{5}$ where the unit vector along the direction of propagation is given by $\mathbf{r} = [\cos a \cos b, \cos a \sin b, \sin a]$. We use an initial density $\rho = 1$, an adiabatic equation of state with $\gamma = 5/3$ and $P = 0.1$. We perform the ‘travelling wave test’ from Gardiner & Stone (2008) where $[v_1, v_2, v_3] = [0, 0.1 \sin(2\pi x_1/\lambda), 0.1 \cos(2\pi x_1/\lambda)]$ and $[B_1, B_2, B_3] = [1, 0.1 \sin(2\pi x_1/\lambda), 0.1 \cos(2\pi x_1/\lambda)]$ where the wavelength $\lambda = 1$ and with these vectors projected back into the x , y and z components using the transformations

given by (Gardiner & Stone, 2008)

$$\begin{aligned} x &= x_1 \cos a \cos b - x_2 \sin b - x_3 \sin a \cos b, \\ y &= x_1 \cos a \sin b + x_2 \cos b - x_3 \sin a \sin b, \\ z &= x_1 \sin a + x_3 \cos a. \end{aligned} \quad (322)$$

Figure 27 shows the results of this test using $32 \times 18 \times 18$, $64 \times 36 \times 39$ and $128 \times 74 \times 78$ 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 28 shows a convergence study on this problem, showing, as in Gardiner & Stone (2008), the 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 2.3.9 and 2.11.6). 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), in the absence of correction terms this problem is unstable to the SPMHD tensile instability (e.g. Phillips & Monaghan, 1985) since the plasma $\beta \equiv \frac{1}{2}B^2/P \approx 0.2$. Our results demonstrate that the correction term (Section 2.11.3) effectively stabilises the numerical scheme without affecting the convergence properties.

4.6.2 MHD shock tubes

The classic Brio & Wu (1988) shock tube test generalises the Sod shock tube (Section 4.1.1) to MHD. It has provoked debate over the years (e.g. Wu 1988; Dai & Woodward 1994a; Falle & Komissarov 2001; Takahashi et al. 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 et al., 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,b) and 1.5D results on this test with SPMHD, for comparison, can be found in e.g. Price & Monaghan (2005); Rosswog & Price (2007); Dolag &

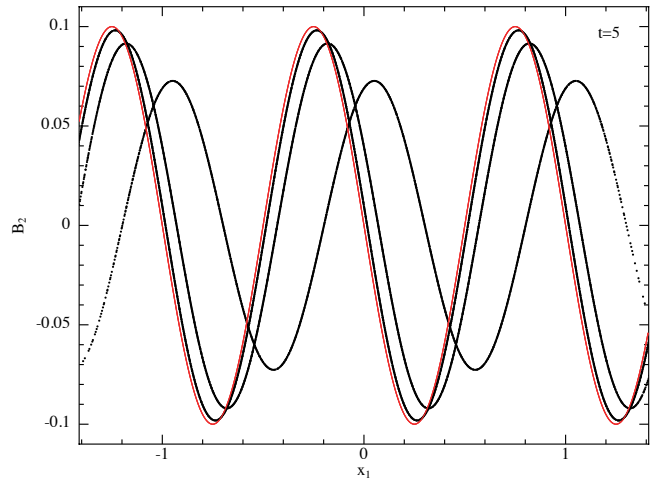


Figure 27. Results of the 3D circularly polarised Alfvén wave test after 5 periods, showing perpendicular component of the magnetic field on all particles (black dots) as a function of distance along the axis parallel to the wave vector. Results are shown using $32 \times 18 \times 18$, $64 \times 36 \times 39$ and $128 \times 74 \times 78$ particles (most to least damped, respectively), compared to the exact solution given by the solid red line. Convergence is shown in Figure 28.

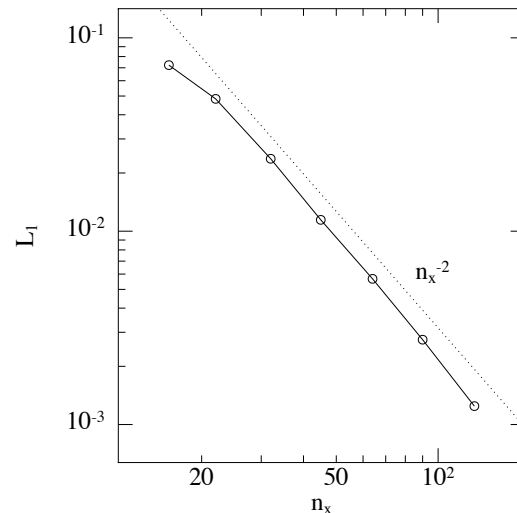


Figure 28. Convergence in the 3D circularly polarised Alfvén wave test, showing the L_1 error as a function of the number of particles along the x -axis, n_x alongside the expected slope for second order convergence (dotted line). Significantly, this demonstrates second order convergence *with all dissipation switched on*.

Stasyszyn (2009); Price (2010) and Vanaverbeke et al. (2014), with 2D versions shown in Price (2012a), Tricco & Price (2013) and Tricco et al. (2016a). We handle the boundary conditions in PHANTOM by setting the first and last few planes of particles to be ‘boundary particles’ (Section 2.1.5), meaning that the gas properties on these particles are fixed.

Figure 29 shows the results of the Brio & Wu (1988) problem using PHANTOM, performed in 3D with $256 \times$

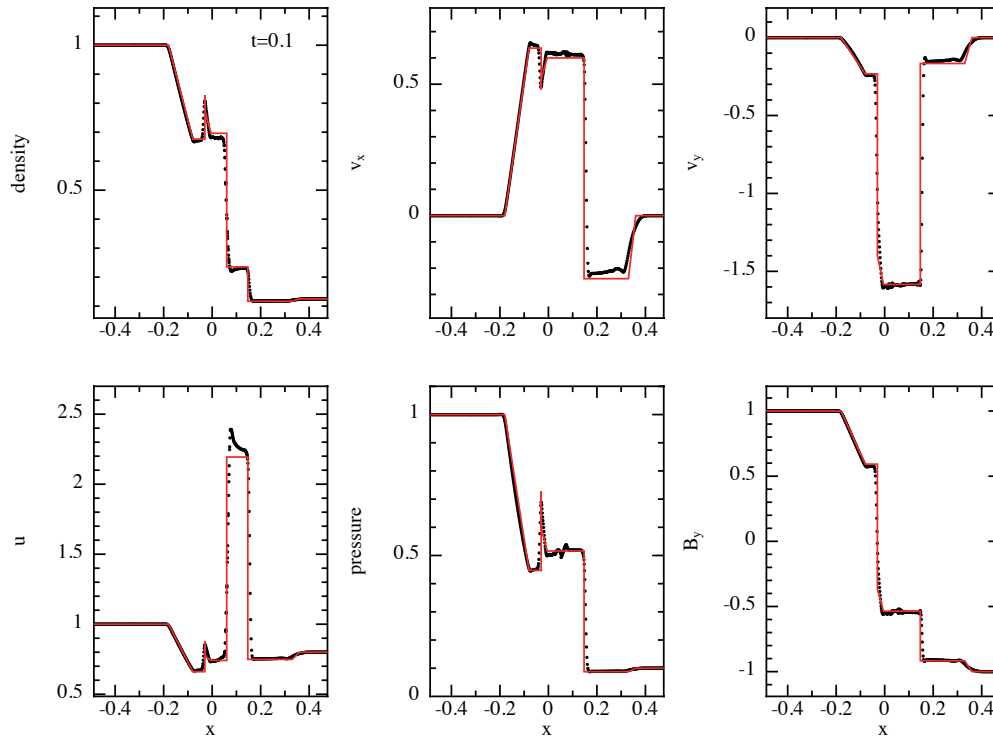


Figure 29. Results of the [Brio & Wu \(1988\)](#) shock tube test in 3D, showing projection of all particles (black dots) compared to the reference solution (red line). The problem is set up with $[\rho, P, B_y, B_z] = [1, 1, 1, 0]$ for $x \leq 0$ and $[\rho, P, B_y, B_z] = [0.125, 0.1, -1, 0]$ for $x > 0$ 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 $x \in [0, 0.5]$.

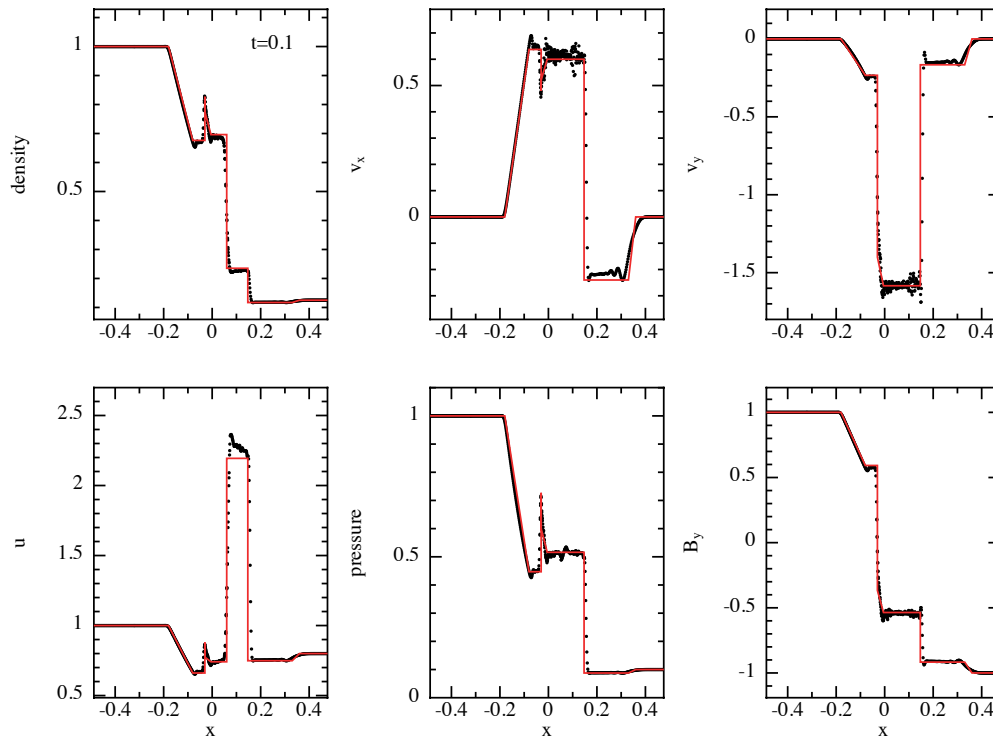


Figure 30. As in Figure 29 but using code defaults which give second order convergence away from shocks. Some additional noise in the velocity field is visible, while otherwise the solutions are similar.

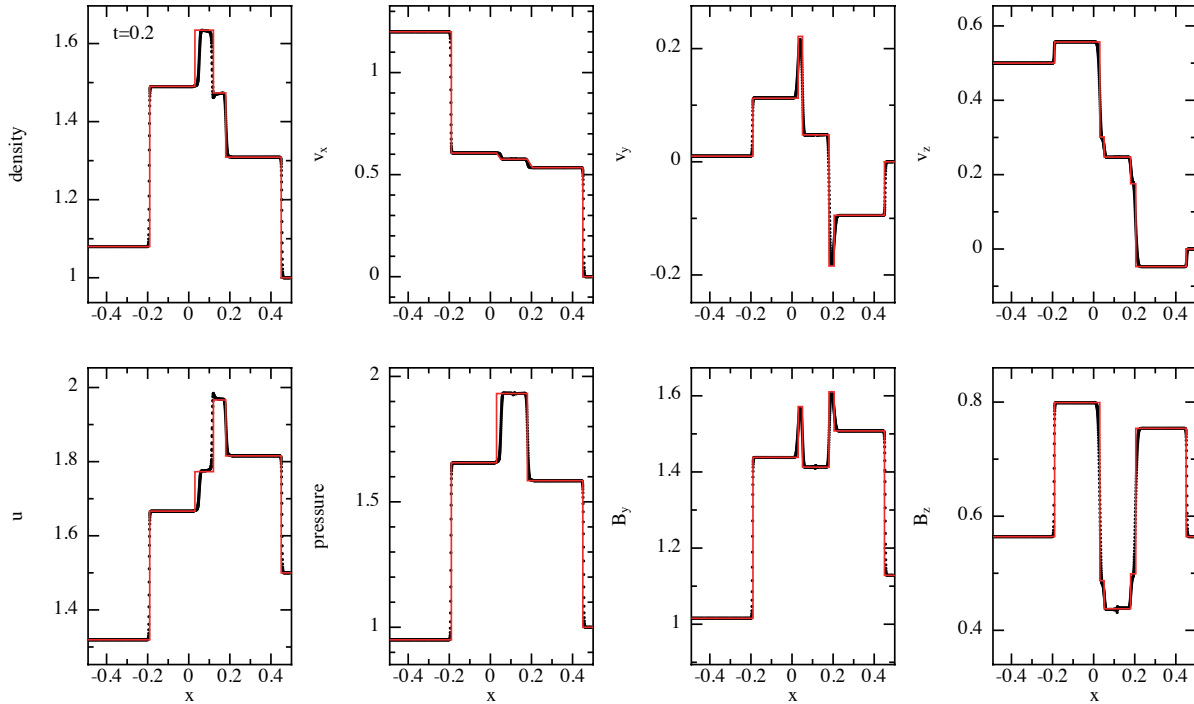


Figure 31. Results of the 7-discontinuity MHD shock tube test 2a from [Ryu & Jones \(1995\)](#) in 3D, showing projection of all particles (black dots) compared to the reference solution (red line). The problem is set up with $[\rho, P, v_x, v_y, v_z, B_x, B_y, B_z] = [1.08, 0.95, 1.2, 0.01, 0.5, 2/\sqrt{4\pi}, 3.6/\sqrt{4\pi}, 2/\sqrt{4\pi}]$ for $x \leq 0$ and $[\rho, P, v_x, v_y, v_z, B_x, B_y, B_z] = [1, 1, 0, 0, 2/\sqrt{4\pi}, 4/\sqrt{4\pi}, 2/\sqrt{4\pi}]$ 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 $x \in [0, 0.5]$.

24×24 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 the numerical solution taken from [Balsara \(1998\)](#). Figure 29 shows the results when a constant $\alpha^{\text{AV}} = 1$ is employed, while Figure 30 shows the results with default code parameters, giving second order dissipation away from shocks.

Figure 31 shows the result of the “7 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. The reference solution is shown by the red lines for comparison. Here the boundary particles are moved with a fixed velocity in the x -direction.

4.6.3 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](#); [Stone & Norman, 1992](#); [Ryu et al., 1995](#); [Dai & Woodward, 1998](#); [Tóth, 2000](#); [Londrillo & Del Zanna, 2000](#); [Price & Monaghan, 2005](#); [Stone et al., 2008](#)). 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 4.1.4). We use an initial plasma $\beta_0 = 10/3$, initial Mach number $\mathcal{M}_0 = v_0/c_{s,0} = 1$, initial velocity field $[v_x, v_y, v_z] = [-v_0 \sin(2\pi y'), v_0 \cos(2\pi x'), 0.01v_0]$ and magnetic field $[B_x, B_y, B_z] = [-B_0 \sin(2\pi y'), B_0 \sin(4\pi x'), 0]$, where $v_0 = 1$, $B_0 = 1/\sqrt{4\pi}$, $x' \equiv x - x_{\min}$ and $y' \equiv y - y_{\min}$; 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 equation of state with $\gamma = 5/3$.

Figure 32 shows the results at $t = 0.5$ (top row) and at $t = 1$ (bottom) at three different resolutions (left to right). The main noticeable change with resolution at $t = 0.5$ is that the shocks become more well defined, as well as the dense filament consisting of material trapped in the reconnecting layer of magnetic field in the centre of the domain. This current sheet eventually be-

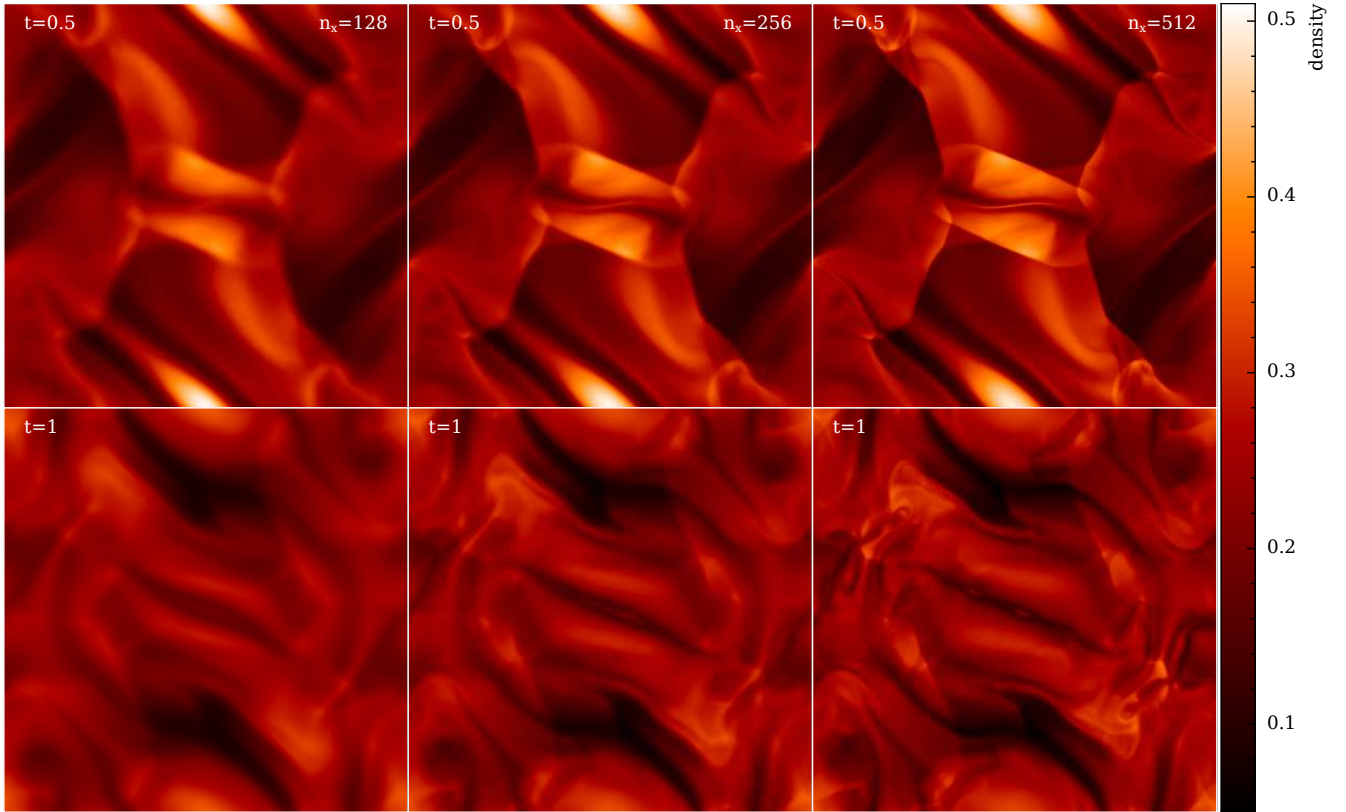


Figure 32. Density in a $z = 0$ cross section of the Orszag-Tang vortex test performed in 3D. Results are shown at $t = 0.5$ (top) and $t = 1$ (bottom) at a resolution of $128 \times 148 \times 12$, $256 \times 296 \times 12$ and $512 \times 590 \times 12$ particles (left to right). Compare e.g. to Figure 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).

comes unstable to the tearing mode instability (e.g. Furth et al., 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 et al. 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 2.11.6) and divergence cleaning (Section 2.11.3) 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, 2013).

4.6.4 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 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 $n_x = 256$

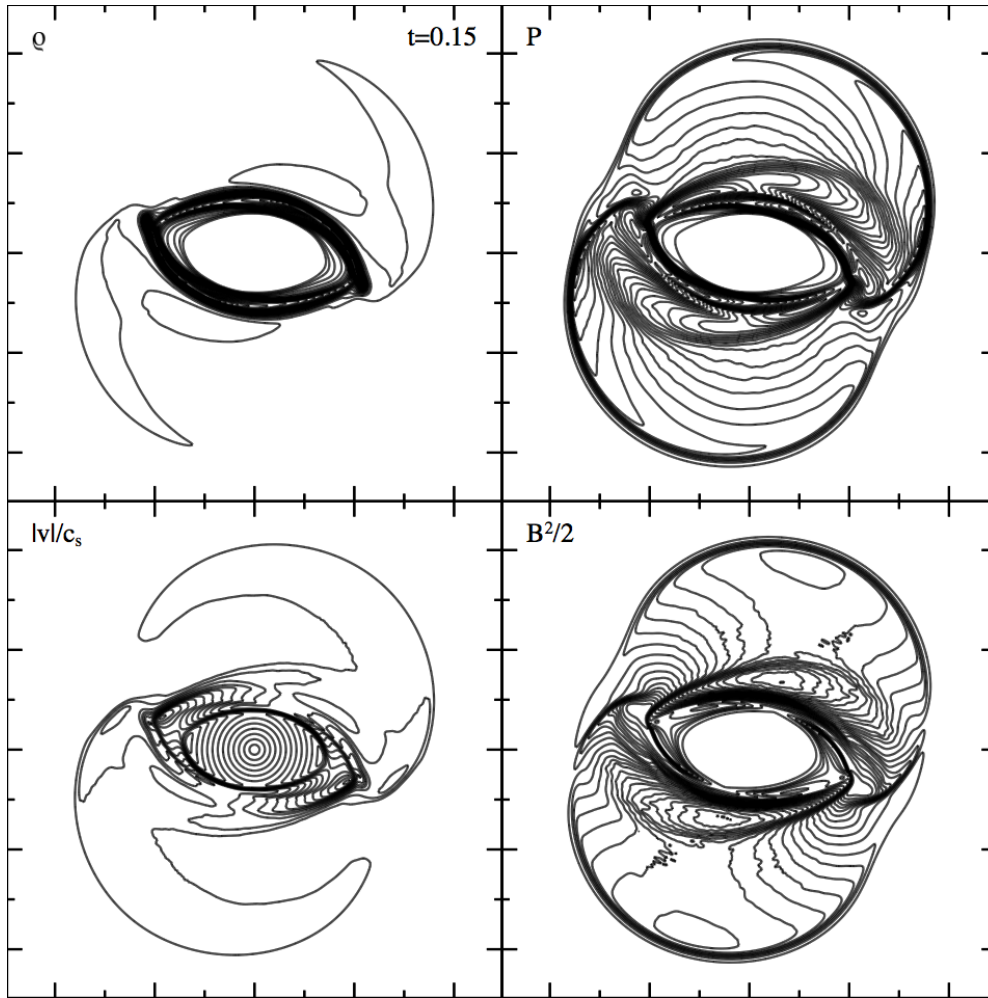


Figure 33. Density, pressure, Mach number and magnetic pressure shown in a $z = 0$ cross section at $t = 0.15$ in the 3D MHD rotor problem, using $n_x = 256$, equivalent to $\sim 300^2$ resolution elements in 2D. The plots show 30 contours with limits identical to those given by [Tóth \(2000\)](#); $0.483 < \rho < 12.95$, $0.0202 < P < 2.008$, $0 < |v|/c_s < 1.09$ and $0 < \frac{1}{2}B^2 < 2.642$.

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 33 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 2.11; see also recent results in [Tricco et al. 2016a](#))

4.6.5 Current loop advection

The current loop advection test was introduced by [Gardiner & Stone \(2005, 2008\)](#), 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 $B_x = -A_0y/r$, $B_y = A_0x/r$ 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 equation of state 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).

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 perfectly (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 34 shows the results of this test in PHANTOM with $128 \times 74 \times 12$ 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 4.6.2–4.6.4). Importantly, our implementation of artificial resistivity (Section 2.11.6) 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 2.11.3), in particular to any spurious divergence of \mathbf{B} that is measured by the SPMHD divergence operator (Equation 177). For this reason the results using the quintic kernel (Equation 19) are substantially better than those using the cubic spline, because the initial measurement of $\nabla \cdot \mathbf{B}$ is smaller and so the evolution is less affected by the divergence cleaning.

4.6.6 MHD blast wave

The MHD blast wave problem consists of an overpressurised 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 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 $\mathbf{B} = [10/\sqrt{2}, 0, 10/\sqrt{2}]$. 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 35 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).

4.7 Non-ideal MHD

The following tests demonstrate the non-ideal MHD algorithms (Section 2.12). We adopt periodic boundary conditions for both tests, initialising the particles on a close-packed lattice with an isothermal equation of state, $P = c_s^2 \rho$, and using the C^4 Wendland smoothing kernel.

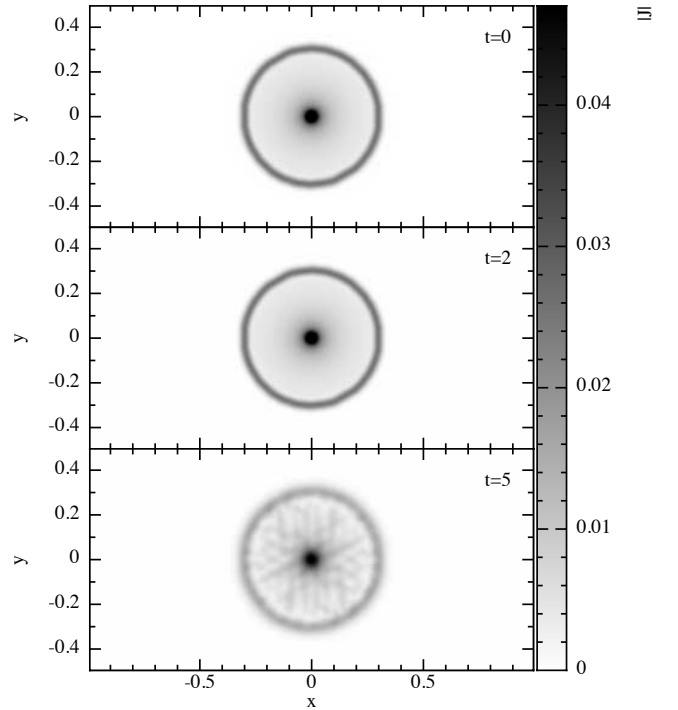


Figure 34. Magnitude of the current density $|\nabla \times \mathbf{B}|$ in the current loop advection test performed in 3D, showing comparing the initial conditions (top) to the result after two (middle) and five (bottom) crossings of the box, using $128 \times 74 \times 12$ particles. *Full dissipation, shock capturing and divergence cleaning terms were applied for this test, without which the advection is perfect. The advection is affected mainly by the divergence cleaning acting on the outer (infinite) current sheet.*

4.7.1 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 et al. (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 $\mathbf{B} = B_0 \hat{\mathbf{x}}$ with $B_0 = 1$, sound speed of $c_s = 1$, and velocity of $\mathbf{v} = v_0 \sin(kx) \hat{\mathbf{z}}$ 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 128 particles in the x -direction.

The solution to the dispersion relation for Alfvén waves (Balsara, 1996),

$$\omega^2 + \eta_{AD} k^2 \omega i - v_A^2 k^2 = 0, \quad (323)$$

where $\omega = \omega_R + \omega_I i$ is the complex angular frequency of the wave, giving a damped oscillation in the form

$$h(t) = h_0 |\sin(\omega_R t)| e^{\omega_I t}. \quad (324)$$

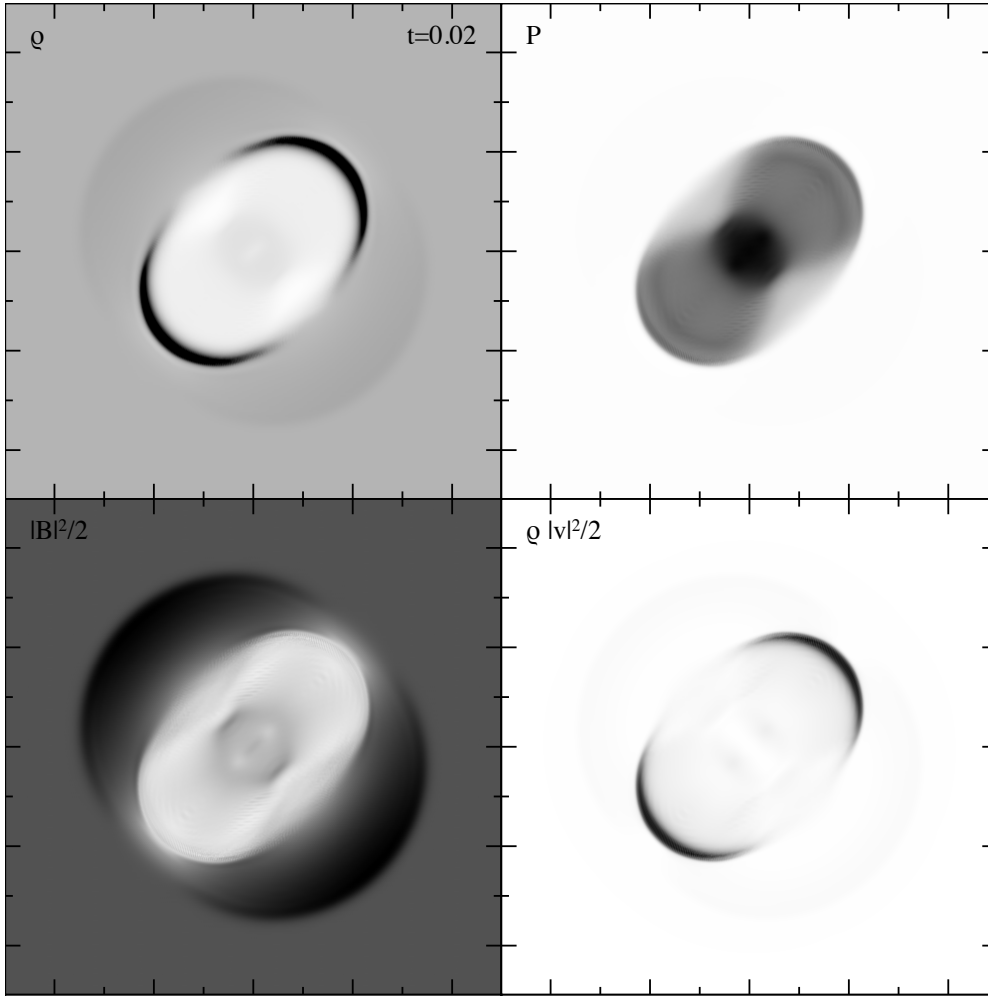


Figure 35. Slices through $y = 0$ for the MHD blast wave problem showing density (top left), gas pressure (top right), magnetic energy density (bottom left) and kinetic energy density (bottom right). The plot limits are $\rho \in [0.19, 2.98]$, $P \in [1, 42.4]$, $[25.2, 64.9]$ for the magnetic energy density and $[0, 33.1]$ for the kinetic energy density. These are directly comparable to Figure 8 in [Gardiner & Stone \(2008\)](#).

In our test, $h(t)$ corresponds to 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 36 shows the time evolution of $\langle B_z^2 \rangle^{1/2}$ to $t = 5$ from the code (blue line), which may be compared with the analytic solution given by the red line. At the end of the test, the L_2 error is 7.5×10^{-5} (evaluated at intervals of $dt = 0.01$), demonstrating close agreement between the numerical and analytical results.

Given that, by design, there is very little motion of the particles and that we have excluded artificial dissipation, the particles tend to fall off the lattice given other initial configurations. 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 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.

4.7.2 Standing shock

To test the Hall effect, we compare our solutions against the 1D isothermal steady-state equations for the strong Hall effect regime, as done in (e.g.) [Falle \(2003\)](#), [O’Sullivan & Downes \(2006\)](#) and [Wurster et al. \(2016\)](#); the numerical solution to this problem is given in these texts, and is also summarised in Appendix C1.2 of [Wurster et al. \(2016\)](#).

The left- and right-hand side of the shock are initialised with $(\rho_0, v_{x,0}, v_{y,0}, v_{z,0}, B_{x,0}, B_{y,0}, B_{z,0}) = (1.7942, -0.9759, -0.6561, 0.0, 1.0, 1.74885, 0.0)$ and $(1.0, -1.751, 0.0, 0.0, 1.0, 0.6, 0.0)$, 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

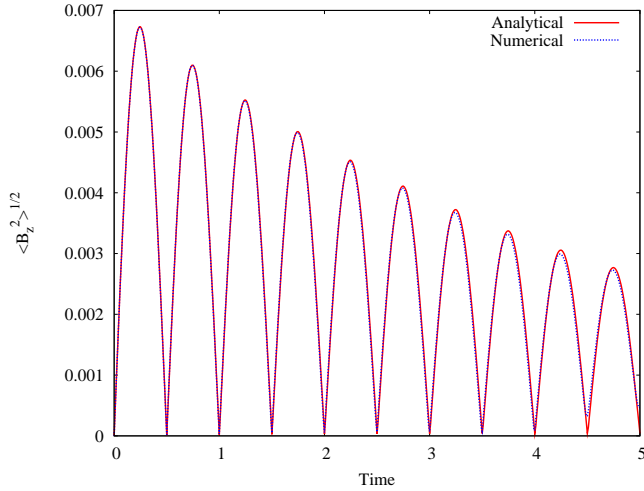


Figure 36. Wave damping test showing the decay of Alfvén waves in the presence of ambipolar diffusion, using the coefficient $\eta_{AD} = 0.01v_A^2$. The L_2 error between the analytic and numerical solution is 7.5×10^{-5} .

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 $x'_r = x_r - v_0 t_{\max}$, where t_{\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 $\eta_{OR} = 1.12 \times 10^{-12}$, $\eta_{HE} = -3.53 \times 10^{-2}B$, and $\eta_{AD} = 7.83 \times 10^{-3}v_A^2$. 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 $x \geq 0$, with the domain extending from $x_l = x'_l = -2$ to $x_r = 2$ with $x'_r = 3.75$.

Figure 37 shows v_x and B_y for both the numerical and analytical results, which agree within 3 per cent at any given position. On the left-hand side of the shock interface, the numerical results are slightly lower than the analytical solution as a result of the artificial dissipation terms, which are required to properly model a shock. These results are robust against variation in kernel choice and initial particle lattice configurations compared to the wave damping test (see Section 4.7.1).

4.8 Self-gravity

4.8.1 Polytrope

To test the self-gravitating gas dynamics, we first test the static structure of a polytrope, whose equation of state is $P = K\rho^\gamma$. Similar tests have been shown for SPH codes dating back to the original papers of [Gingold & Monaghan \(1977, 1978, 1980\)](#). As simple stellar models we have used them in applications ranging from

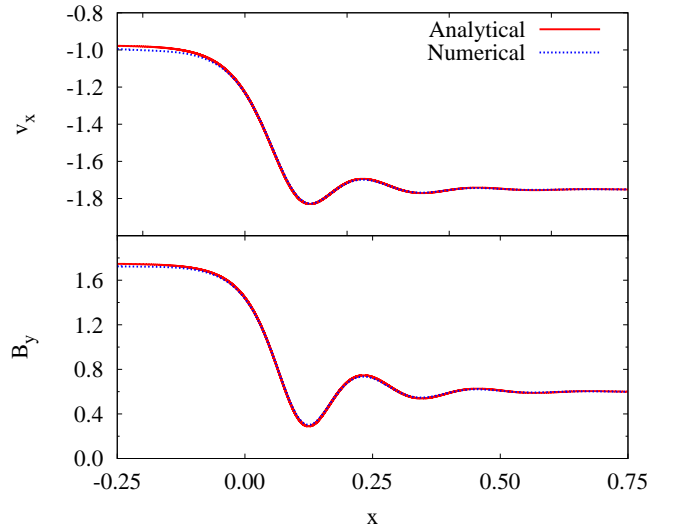


Figure 37. Hall-dominated standing shock using $\eta_{OR} = 1.12 \times 10^{-12}$, $\eta_{HE} = -3.53 \times 10^{-2}B$, and $\eta_{AD} = 7.83 \times 10^{-3}v_A^2$. The numerical and analytical results agree to within 3 per cent everywhere.

common envelope evolution ([Iaconi et al., 2017](#)) to tidal disruption events ([Coughlin & Nixon, 2015](#); [Coughlin et al., 2016b,a](#); [Bonnerot et al., 2016, 2017](#)).

We compute the exact solution is by numerically solving

$$\frac{\gamma K}{4\pi G(\gamma - 1)} \frac{d^2}{dr^2} (r\rho^{\gamma-1}) + r\rho = 0, \quad (325)$$

which is then scaled to give a polytrope of radius $R = 1$. In code units for a sphere of unit mass and $\gamma = 5/3$, we set $K = 0.4244$.

The particles are initially set up on a hexagonal close-packed lattice, truncated at $R = 1$. This sphere of uniform density is then stretched using the `stretchmap` algorithm (see Section 3.4) so that the initial radial density profile matches the exact solution. Figure 38 shows the solution at $t = 100$ in code units. As a result of the setup, the polytrope relaxes almost immediately, with only a slight rearrangement of the particles moving off of the stretched lattice; for all time, the density profile is equal to the exact solution within 3 per cent for $r \leq 0.7$.

Other initial density profiles have been tested, and in each case, the polytrope relaxes to agree with the exact solution; the relaxation time varies depending on how far out of equilibrium the initial configuration is.

Once the static solution is obtained, we check energy conservation by considering radial oscillations. We apply a velocity perturbation of $v_r = 0.2r$ to the $N \approx 10^5$ particle model, and the polytrope is evolved 100 time units without (alpha and beta) artificial viscosity. The total energy is conserved within 0.03 per cent.

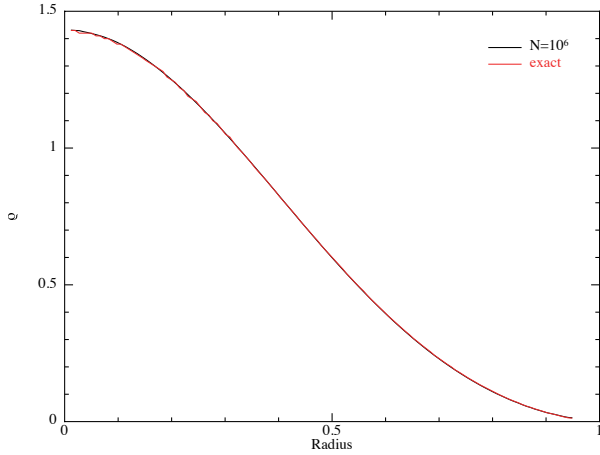


Figure 38. Polytrope static structure using 10^6 particles (black), compared to the exact solution (red), shown at $t = 100$.

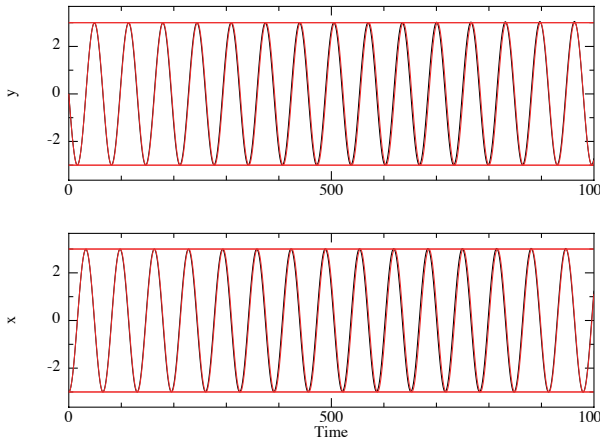


Figure 39. y - and x -positions of the centre of mass of one star of a binary system (top and bottom, respectively). Each star is a polytrope with mass and radius of unity, and $N = 10^4$ particles; the initial separation is 6.0 in code units. After ~ 15 orbits, the separation remains within 1 percent, 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.

4.8.2 Binary polytrope

Next, we placed two initially unrelaxed, identical polytropes each with $N = 10^4$ particles on a circular orbit with a separation of $6R$ and evolved them for ~ 15 orbits (1000 code units). Figure 39 demonstrates that the separation remains within 1 per cent of the initial separation over the 15 orbits, and that the orbital period remains constant. After the initial relaxation, total energy is conserved to within 0.06 per cent.

The stars are far enough apart that they do not experience any tidal deformation as they orbit. The final density profile of each star agrees with the expected profile within 3 per cent for $r \leq 0.7$.

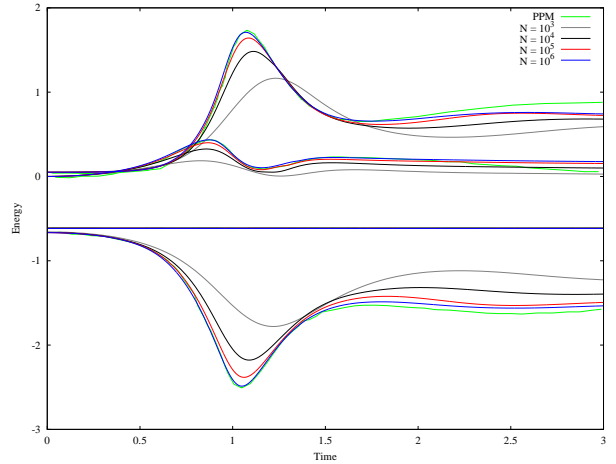


Figure 40. Kinetic, total, thermal and potential energies as a function of time during the Evrard collapse (Evrard, 1988). Green lines are calculated using a 1D PPM code, taken from Figure 6 of Steinmetz & Mueller (1993), while remaining colours show SPH simulations of different resolutions. Both energy and time are given in code units, where $R = M = G = 1$.

4.8.3 Evrard collapse

A common test of SPH gravity codes is the so-called ‘Evrard collapse’ (Evrard, 1988), used in (e.g.) Hernquist & Katz (1989), Steinmetz & Mueller (1993), Thacker et al. (2000), Escala et al. (2004) and many others. 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

$$\rho(r) = \frac{M(R)}{2\pi R^2} \frac{1}{r}. \quad (326)$$

We create the density profile using the same method as for the polytrope (see Section 4.8.1). The sphere is initially isothermal with the specific internal energy set to $u = 0.05GM/R$ with an adiabatic index of $\gamma = 5/3$ and is then allowed to undergo an adiabatic collapse.

In the literature, the results of the Evrard collapse are typically normalised to some characteristic value. Here, we simply show the results in code units (Section 2.3.3) since these units already represent a normalised state. We adopt code units of `udist` = $R = 1$ and `umass` = $M(R) = 1$; the time unit, `utime`, is automatically calculated such that $G \equiv 1$, where G is Newton’s gravitational constant.

Figure 40 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 take 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

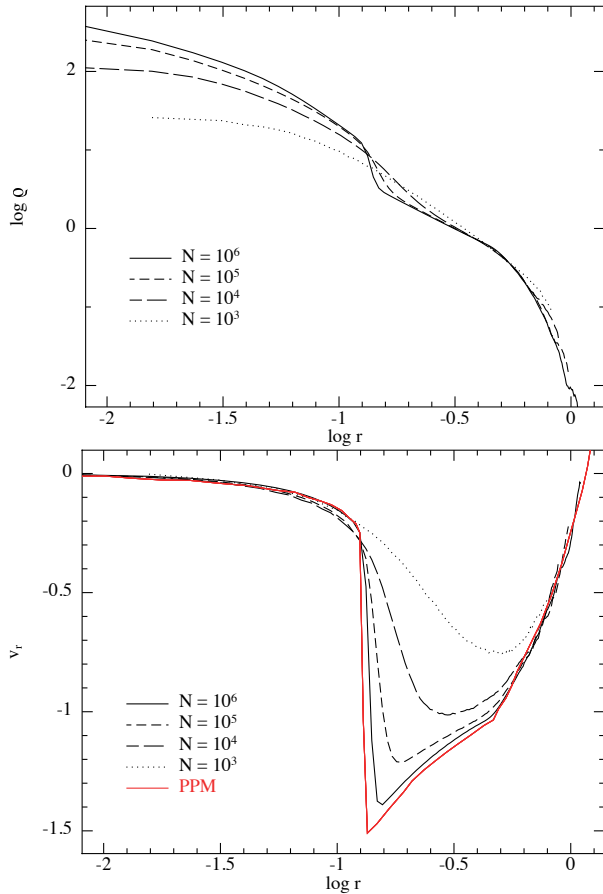


Figure 41. Radial profile of the Evrard collapse (Evrard, 1988) at $t = 0.77$; red lines are taken from Figure 7 of Steinmetz & Mueller (1993), with panels showing density (top) and radial velocity (bottom) as a function of (log) radius. All values are given in code units, where $R = M = G = 1$. The outward propagating shock at $r \approx 0.1$ is sharper at high resolution.

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 41 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 green lines taken from Figure 7 of Steinmetz & Mueller (1993). At this time, the outward propagating shock is at $r \approx 0.1$, with the shock profile being better-defined at higher resolution.

4.9 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 et al., 1995; Thacker & Couchman, 2006. This test requires gravity along with multiple par-

	M/M_{\odot} (10^{10})	m/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

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 .

ticle 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 1).

To create a Milky Way-like galaxy, we used GALACTICS (Kuijken & Dubinski, 1995b; Widrow & Dubinski, 2005; Widrow et al., 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, we duplicated the galaxy and placed the two galaxies 70 kpc apart on a parabolic trajectory. These initial conditions are identical to those used in Wurster & Thacker (2013b,a).

Figure 42 shows the gas column density evolution from $t = 100$ Myr to $t = 1.4$ Gyr, comparing PHANTOM (left) to HYDRA (right). 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.

The two galaxy mergers agree qualitatively with one another, although the trajectories and evolution times slightly differ 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

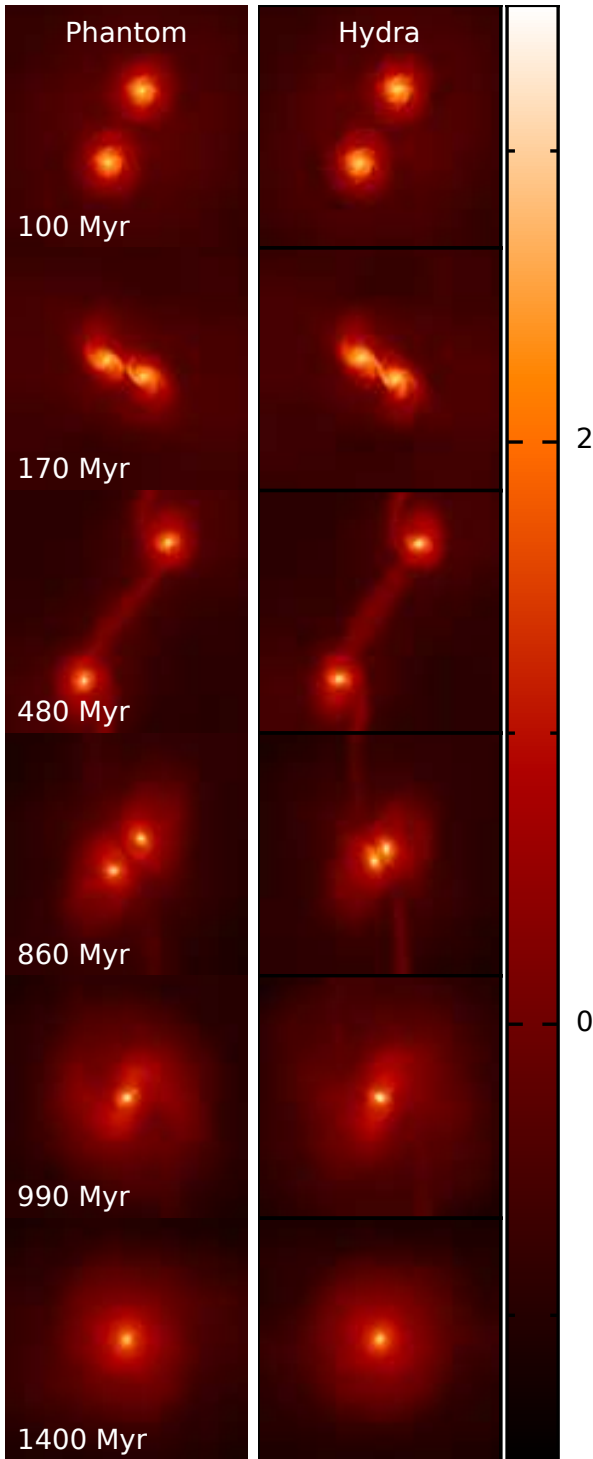


Figure 42. Evolution of the gas column density in a major merger of two Milky Way-sized galaxies, comparing PHANTOM to the HYDRA code. Times shown are from the onset of the simulation, with each frame 100 kpc². The colour bar is log (Column density / (M_⊙ pc⁻¹)).

HYDRA and PHANTOM, respectively. Second periapsis occurs at 875 and 905 Myr for HYDRA and PHANTOM, respectively, which is a difference of 3.4 per cent since the beginning of the simulation. The maximum gas density is ≈ 10 times lower in PHANTOM, indicating a more diffuse core.

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 the tree (c.f. Section 2.13.4). 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 $\epsilon_{S2} = 2.34\epsilon_{\text{Plummer}}$; for this simulation, $\epsilon_i \equiv \epsilon_{\text{Plummer}} = 300$ 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 $h_{\min} = \epsilon_{\text{Plummer}}/8$ ($= 37.5$ pc). In PHANTOM, h_i is always calculated self-consistently and thus has no imposed lower limit.

Finally, PHANTOM contains an artificial conductivity term (Sec 2.3.8) 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 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 per cent of one another, as measured from the beginning of the simulation.

The quantitative 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_{\min} and the suppression of ‘mixing’ (which occurs when no thermal conductivity is applied). We reach a similar conclusion, finding a result with PHANTOM much closer to the RAMSES solution.

4.10 Dust-gas mixtures

The SPH algorithms used in PHANTOM for dust-gas mixtures have been extensively benchmarked in Laibe & Price (2012a,b) (for the two fluid method) and in Price & Laibe (2015a) (for the one fluid method; here-

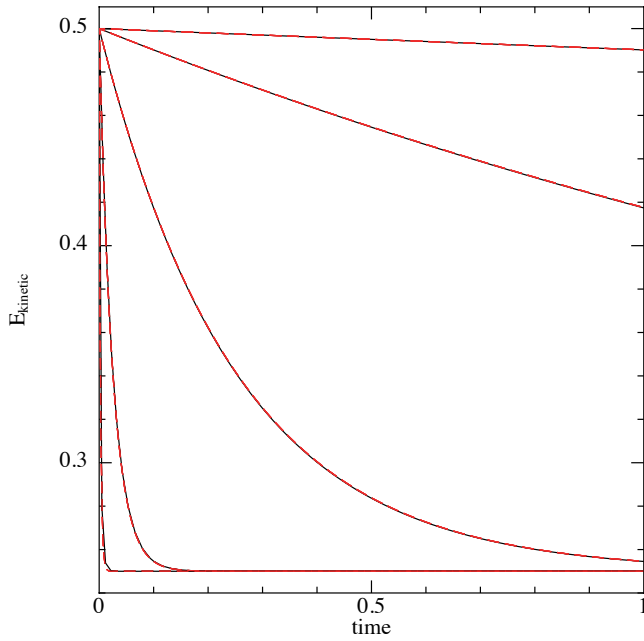


Figure 43. Decay of kinetic energy as a function of time in the DUSTYBOX test, involving a uniformly translating mixture of gas and dust coupled by drag. Solid lines show the PHANTOM results for drag coefficients $K = 0.01, 0.1, 1.0, 10$ and 100 (top to bottom), which may be compared to the corresponding analytic solutions given by the dashed red lines.

after 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) and Ragusa et al. (2017).

4.10.1 DUSTYBOX

Figure 43 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 $32 \times 36 \times 39$ gas particles set on a close-packed lattice and $32 \times 36 \times 39$ dust particles also set on a close-packed lattice. The gas particles are initially rest while the dust is given a uniform velocity $v_x = 1$ in the x-direction. We employ an isothermal equation of state with unit sound speed, 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 2.14.4) for the drag terms, following Laibe & Price (2012a).

The red dashed lines in Figure 43 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 $E_{\text{kin}}(t) = \frac{1}{2}[v_g(t)^2 + v_d(t)^2]$. The close match between the numerical and analytic solu-

tions ($L_2 \sim 3 \times 10^{-4}$) demonstrates that the drag terms are implemented correctly.

The DUSTYBOX test is irrelevant for the one fluid method (Section 2.14.14) since this method implicitly assumes that the drag is strong enough so that the terminal velocity approximation holds — implying that the relative velocities are simply the barycentric values at the end of the DUSTYBOX test.

4.10.2 DUSTYWAVE

Laibe & Price (2011) derived the analytic solution for linear waves in a dust-gas mixture: The ‘DUSTYWAVE’ test. The corresponding dispersion relation is given by

$$(\omega^2 - c_s^2 k^2) - \frac{i}{\omega t_s} (\omega^2 - \tilde{c}_s^2 k^2) = 0, \quad (327)$$

where $\tilde{c}_s = c_s(1 + \rho_d/\rho_g)^{-1/2}$ is the modified sound speed (e.g. Miura & Glass, 1982). This demonstrates the two important limits i) $t_s \rightarrow \infty$, giving undamped sound waves in the gas and ii) $t_s \rightarrow 0$, 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 44.

Two fluid. We perform this test first with the two fluid algorithm, using $64 \times 12 \times 12$ gas particles and $64 \times 12 \times 12$ 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 $v_g = v_d = A \sin(2\pi x)$, $\rho = \rho_0[1 + A \sin(2\pi x)]$ with $\rho_0 = 1$ and $A = 10^{-4}$. The density perturbation is initialised using stretch mapping (Section 3.4; see also Appendix B in Price & Monaghan 2004b). We perform this test using an adiabatic equation of state with $c_{s,0} = 1$. We adopt a simple, constant K drag prescription, choosing $K = 0.5, 5, 50$ and 500 such that the stopping time (Equation 237) is a multiple of the wave period ($t_s = 1, 0.1, 0.01$ and 0.001 , respectively).

The left panels in Figure 44 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 $t_s \gtrsim 0.1$ the numerical solution matches the analytic solution to within 4%. For short stopping times, Laibe & Price (2012a) showed that the resolution criterion $h \lesssim c_s t_s$ 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 $t_s \lesssim 0.016$. This is evident from the lower two panels in Figure 44, where the numerical solution is overdamped compared to the analytic solution.

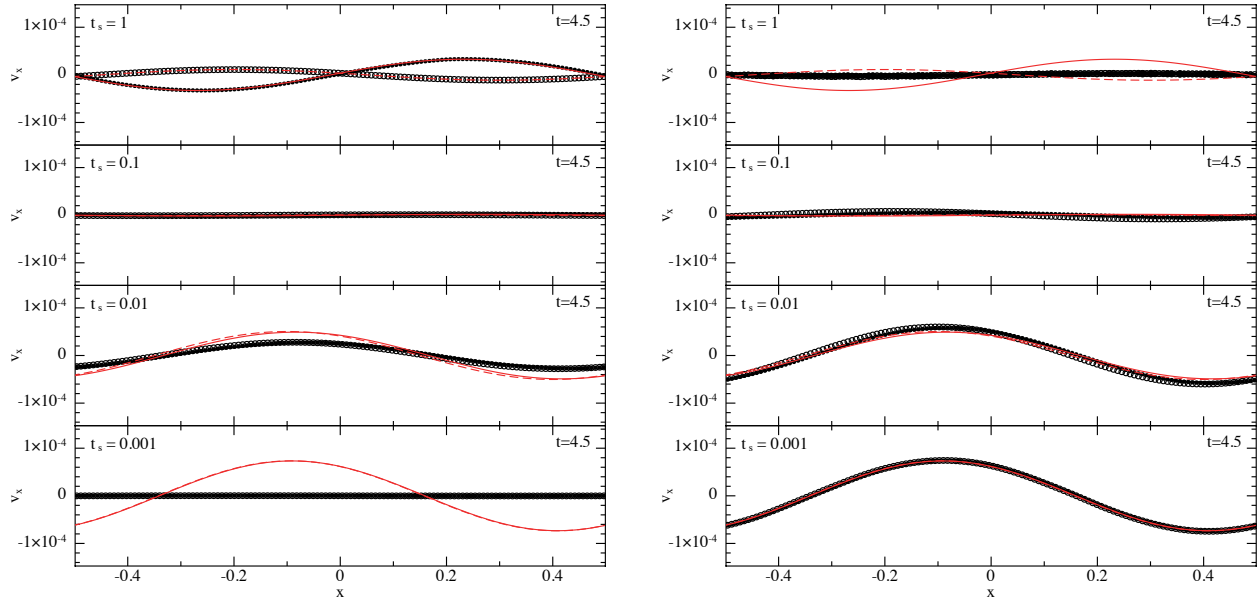


Figure 44. Velocity of gas (solid) and dust (circles) at $t = 4.5$ in the DUSTYWAVE test, using the two fluid method (left; with $64 \times 12 \times 12$ gas particles, $64 \times 12 \times 12$ dust particles) and the one fluid method (right; with $64 \times 12 \times 12$ mixture particles) and a dust-to-gas 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 $h \lesssim c_s t_s$ 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).

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_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 one-fluid formulation in the terminal velocity approximation (Section 2.14.12). To compare and contrast the two methods for simulating dust in PHANTOM, the right half of Figure 44 shows the results of the DUSTYWAVE test computed with the one fluid method. To perform this test we set up a single set of $64 \times 12 \times 12$ ‘mixture’ particles placed on a uniform closepacked lattice, with an initially uniform dust fraction $\epsilon = 0.5$. 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 nu-

merical 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.

4.10.3 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 $32 \times 32 \times 32$ particles with an initial dust fraction set according to

$$\epsilon(r, 0) = \begin{cases} \epsilon_0 \left[1 - \left(\frac{r}{r_c} \right)^2 \right], & r < r_c, \\ 0, & \text{elsewhere,} \end{cases} \quad (328)$$

with $\epsilon_0 = 0.1$ and $r_c = 0.25$. We then evolve the dust diffusion equation (Equation 273), discretised according to Equation (280), 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 45 shows the evolution of the dust fraction $\epsilon \equiv \rho_d / \rho$ 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

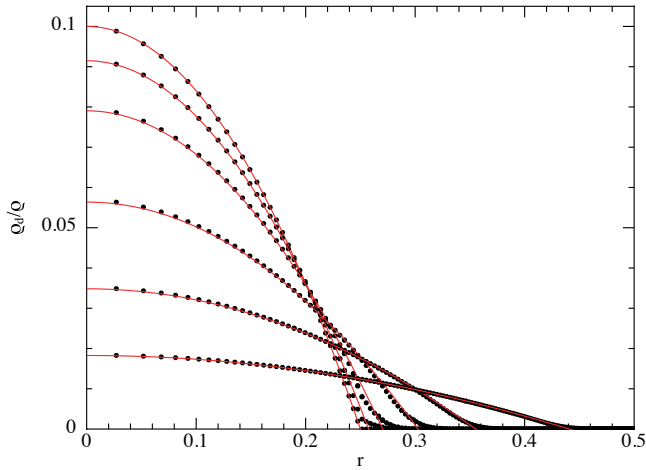


Figure 45. Dust diffusion test from Price & Laibe (2015a), showing the evolution of the dust fraction on the particles (black dots) as a function of radius at 6 different times (top to bottom), which may be compared to the analytic solution given by the red lines.

to the analytic solution, with agreement to within 0.3% of the analytic solution at all times.

4.10.4 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 4.6). 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) with density in the ‘vertical’ direction (y) given by

$$\rho(y) = \rho_0 \exp\left(\frac{-y^2}{2H^2}\right), \quad (329)$$

where we choose $H/R_0 = 0.05$ giving $H = 2.5$ au. We use an isothermal equation of state with sound speed $c_s \equiv H\Omega$ where $\Omega \equiv \sqrt{GM/R_0^3}$, corresponding to an orbital time $t_{\text{orb}} \equiv 2\pi/\Omega \approx 353$ yrs. 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

$$\mathbf{a}_{\text{ext}} = -\frac{GM\mathbf{y}}{\sqrt{R_0^2 + y^2}}, \quad (330)$$

where $G = M = 1$ in code units.

The particles are placed initially on a close-packed lattice using $32 \times 110 \times 24 = 84480$ 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 3.4) to give the density profile according to equation (329). We set the mid-plane density to 10^{-3} in code units, or \approx

$6 \times 10^{-13} \text{g/cm}^3$. 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 running the code for 15 orbits with gas only with damping switched on. We then use `phantommoddump` to either i) add a dust fraction to each particle or ii) duplicate the gas particles to create a corresponding set of dust particles. For the dust we assume 1mm grains with an Epstein drag prescription, such that the stopping time is given by equation (251). Since Δv is not available when computing t_s with the one fluid method we set the factor $f = 1$ in equation (251) when using this method (this is a valid approximation since by definition Δv is small when the one fluid method is applicable). The dimensionless stopping time $t_s\Omega = 8.46 \times 10^{-3}$ initially at the disc midplane. After adding dust we continue the simulation for a further 50 orbits.

Figure 46 shows the dust density at intervals of $10 t_{\text{orb}}$, 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 (Equation 262). 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).

4.11 ISM cooling and chemistry

Figure 47 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 5kpc to 10kpc in radius, initially at 10,000K, with a total gas mass of $2 \times 10^9 M_\odot$. The top panels show a temperature and pressure profile of all gas particles. The temperature plateaus around 10,000K 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 is needed to break apart the cold knots, or the inclusion of a large number of sink particles.

The bottom two panels of Figure 47 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

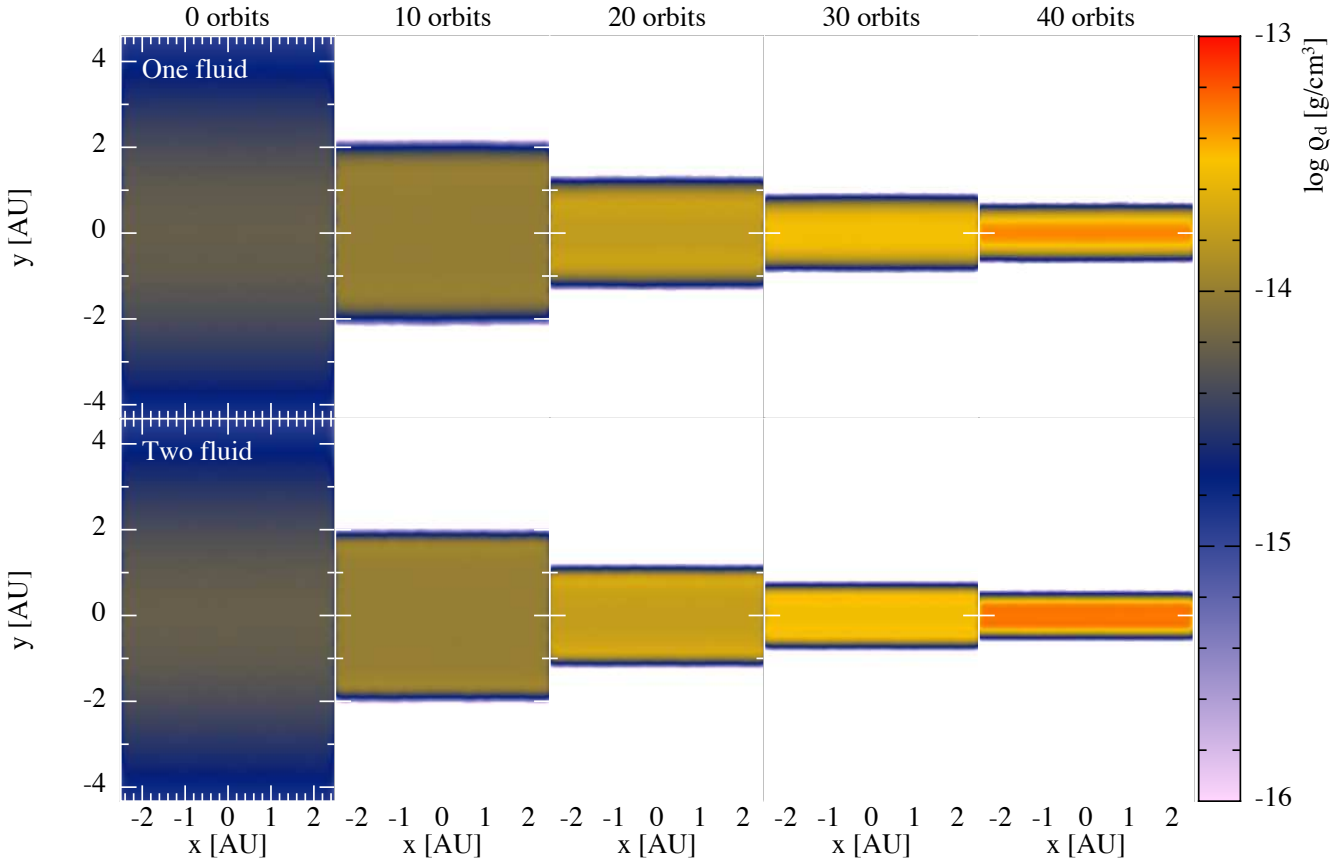


Figure 46. 3D version of the dust settling test from Price & Laibe (2015a), showing the dust density in the ‘r-z’ plane of a protoplanetary disc. We assume mm-sized grains with a 1% initial dust-to-gas ratio in a stratified disc atmosphere with $H/R = 0.05$ with $R_0 = 50$ au in Equation (330).

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. $\times 10$ 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.

5 Software engineering

No code is completely bug-free¹⁰, but 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,
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.

5.1 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 runtime (see Section 4). In PHANTOM, we also *unit test* each module separately. This allows issues to be identified at a much earlier stage in development. The tests are wrapped into the nightly testsuite (Section 5.2) and can also be run by the user (e.g. by typing `./phantom test`). When a bug that escapes the testsuite has been discovered, we have endeavoured to create a unit test to prevent a future recurrence.

¹⁰Experience is the name everyone gives to their mistakes (Wilde, 1892)

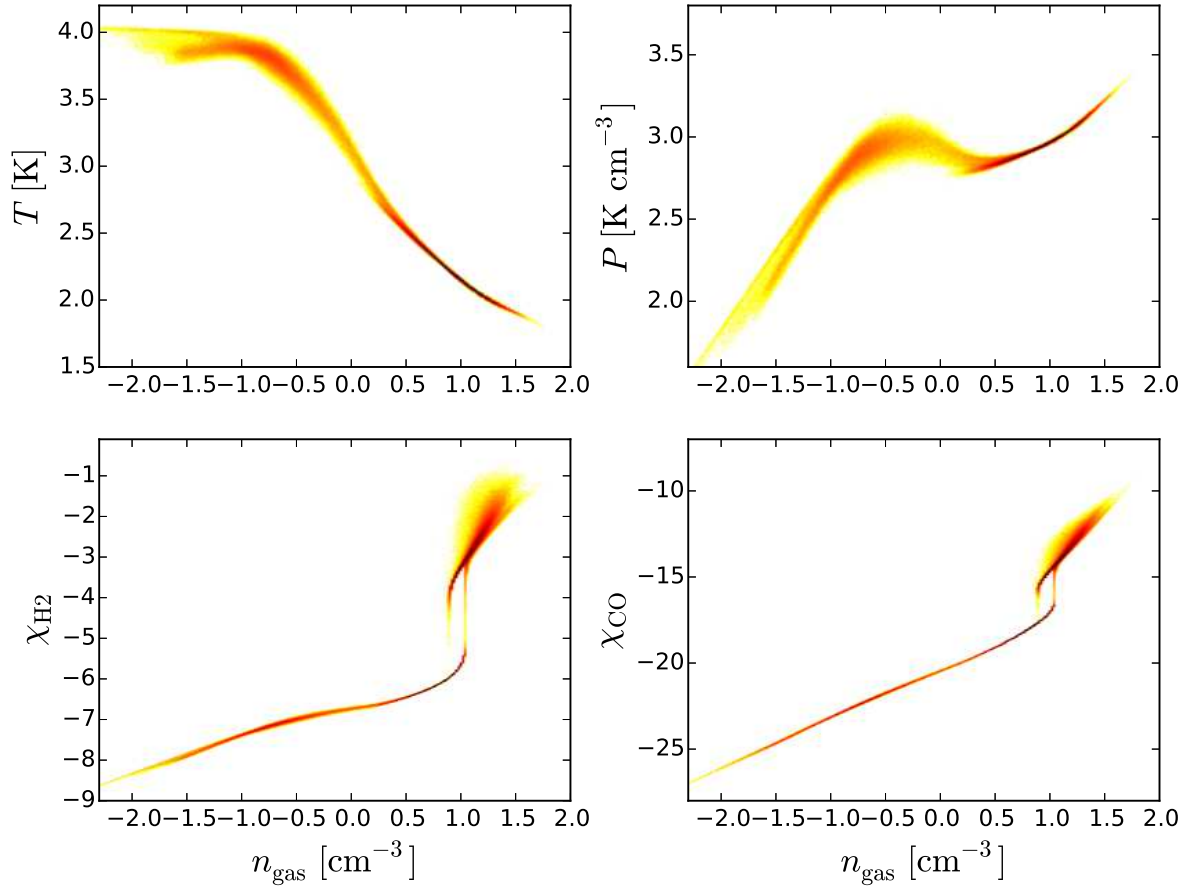


Figure 47. Temperature and pressure profiles resulting from the ISM heating and cooling functions (top) and the abundances of H₂ and CO as a function of gas density (bottom).

5.1.1 Unit tests of derivative evaluations

The unit test of the `derivs` module, `test_derivs`, checks that various derivatives evaluate to within some tolerance of the expected value. To achieve this the test sets up $100 \times 100 \times 100$ 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

$$u_a(x, y, z) = \frac{1}{2\pi} \left[3 + \sin\left(\frac{2\pi x}{L_x}\right) + \cos\left(\frac{2\pi y}{L_y}\right) + \sin\left(\frac{2\pi z}{L_z}\right) \right], \quad (331)$$

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 (35) with the artificial viscosity and other terms switched off, assuming an adiabatic equation of state

(Equation 26) with $\rho = \text{constant}$. We then test that numerical acceleration on all 10^6 particles is within some tolerance of the analytic pressure gradient expected from (331), namely

$$-\frac{\nabla P_a}{\rho_a}(x, y, z) = -(\gamma - 1)\nabla u_a(x, y, z), \quad (332)$$

where typically we use a tolerance of 10^{-3} in the relative error $E(x) = |x - x_{\text{exact}}|/x_{\text{exact}}$.

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 sub-

set 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

$$\frac{dE}{dt} = \sum_a m_a \left(\mathbf{v}_a \cdot \frac{d\mathbf{v}_a}{dt} + \frac{du_a}{dt} \right) = 0, \quad (333)$$

so 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).

5.1.2 Unit tests of sink particles

Unit tests for sink particles include i) integrating a sink particle binary for 1000 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 2.9.2); and v) checking that sink particle creation from a uniform sphere of gas particles (Section 2.9.4) succeeds and that the procedure conserves linear and angular momentum.

5.1.3 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 2.5.5–2.5.7) agree with an iterative solution to the leapfrog corrector step (Equation 66).

5.1.4 Unit tests of neighbour finding routines

In order to unit test the neighbour finding modules, (Section 2.1.8), we set up particles in a uniform random distribution with randomly assigned smoothing lengths. We then check that the neighbour list returned from the call to `get_neighbour_list` agrees with a direct 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, i.e. when all particles lie in a one dimensional line along each of the coordinate axes.

5.1.5 Unit tests of timestepping and periodic boundaries

As a simple unit test of both the timestepping and periodic boundaries we set up $50 \times 50 \times 50$ 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.

5.1.6 Unit tests of file read/write

The unit test for the `readwrite_dumps` module checks 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.

5.1.7 Unit tests of kernel module

The unit tests of the kernel functions 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.

5.1.8 Unit tests of self-gravity routines

In order to unit test the treecode self-gravity computation (Section 2.13), 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.

5.1.9 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 with-

out error. We then perform a low-resolution DUSTYBOX test (Section 4.10.1), checking the solution matches the analytic solution as in Figure 43. For one-fluid dust we perform a low resolution version of the dust diffusion test (Section 4.10.3), 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.

5.1.10 Other unit tests

Various unit tests are written for many of the other modules in PHANTOM also, including sanity checks of individual timestepping utility routines; checking that the barotropic equation of state is continuous; of the coriolis and centrifugal force routines; checks of conservation in the generalised Newtonian potential (Section 2.5.6); of the `revtree` routine to revise the tree; and of the fast inverse square root routines.

5.1.11 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 4.1.3). 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}).

5.2 The Phantom buildbot

The PHANTOM buildbot runs nightly in to check that all possible configurations of the code compile. Provided commits have been pushed to bitbucket in the last 24 hours, the buildbot script pulls the latest version of the code and attempts to compile `phantom`, `phantomsetup` and `phantomanalysis` for every `SETUP` block listed in the `Makefile` and with two different compilers (`ifort` and `gfortran`). Each setup block specifies the collection of modules and compile-time options needed to run a particular problem (e.g. `make SETUP=sedov`), so covers all possible combinations of options currently being employed by PHANTOM users.

The buildbot checks for build and testsuite failures and new compiler warnings in any of these components, and collects the results in an email that is automatically sent to the email addresses of the developers who were responsible for the last 24 hours worth of changes. In this way users receive rapid feedback on the effect of any changes to the code, and as a result build problems and code regressions are quickly identified and fixed, often without need for intervention from either other users or the main developer. The results of the buildbot are also automatically written to the wiki and the

buildbot tags the GIT repository according to its build state: `buildtestfail`, `buildfail`, `testfail` or `ok` so that it is easy for users to identify safe versions of the development code for checkout.

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 three example 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 2.6. Figure 48 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 48 may be directly compared to the panels in Figure 3 of Price & Federrath (2010). Figure 49 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 (Vázquez-Semadeni, 1994; Passot & Vázquez-Semadeni, 1998; Nordlund & Padoan, 1999; Ostriker et al., 1999; Elmegreen & Scalo, 2004; Kritsuk et al., 2007; Lemaster & Stone, 2008; Padoan & Nordlund, 2011; Molina et al., 2012; Konstandin et al., 2012; Federrath & Klessen, 2013; Federrath & Banerjee, 2015). 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

$$\sigma_s^2 = \ln(1 + b^2 \mathcal{M}^2), \quad (334)$$

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,b) and Monaghan & Lattanzio

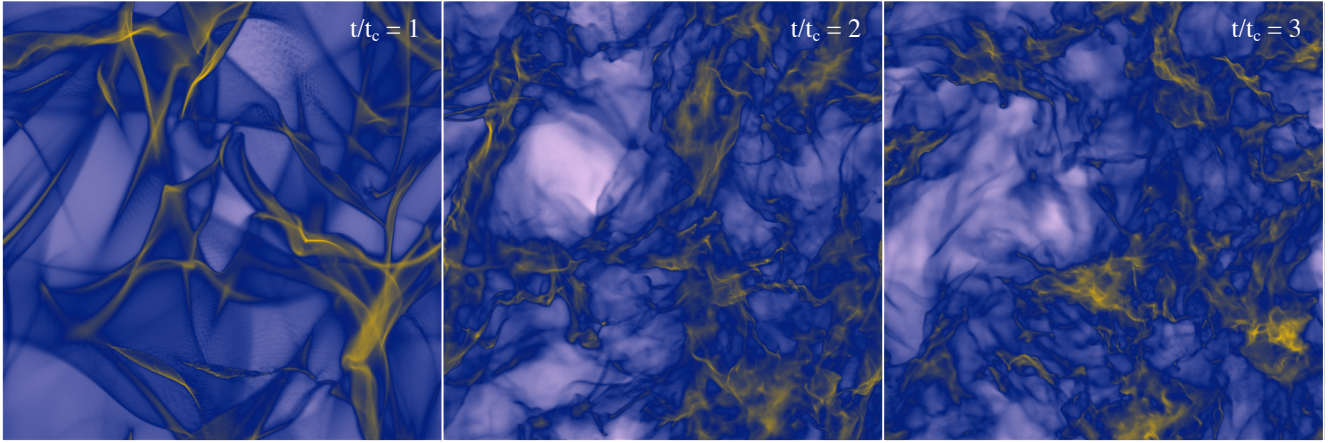


Figure 48. Gas column density in PHANTOM simulations of driven, isothermal, supersonic turbulence at Mach 10, similar to the calculations performed by Price & Federrath (2010). We show the numerical solutions at $t = 1, 2$ and 3 crossing times (left to right, respectively). The colour scale is logarithmic between 10^{-1} and 10 in code units.

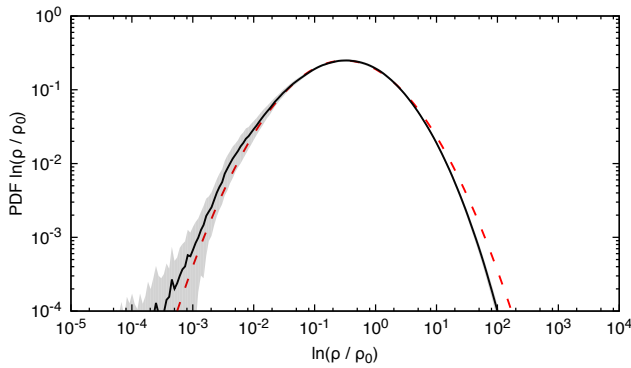


Figure 49. Time-averaged PDF of $s = \ln(\rho/\rho_0)$ for supersonic Mach 10 turbulence, with the shaded region representing the standard deviation of the averaging. The PDF is close to a log-normal distribution, shown by the dashed red line.

(1986, 1991), even motivating the original development of MHD 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 50 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 0.375pc in diameter with a mass of $50M_{\odot}$. 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 equation of state $P = K\rho^{\gamma}$ in the form

$$\frac{P}{\rho} = \begin{cases} c_{s,0}^2, & \rho < \rho_1, \\ c_{s,0}^2 \left(\frac{\rho}{\rho_1}\right)^{(\gamma_1-1)}, & \rho_1 \leq \rho < \rho_2, \\ c_{s,0}^2 \left(\frac{\rho_2}{\rho_1}\right)^{(\gamma_1-1)} \left(\frac{\rho}{\rho_1}\right)^{(\gamma_2-1)}, & \rho_2 \leq \rho < \rho_3, \\ c_{s,0}^2 \left(\frac{\rho_2}{\rho_1}\right)^{(\gamma_1-1)} \left(\frac{\rho_3}{\rho_2}\right)^{(\gamma_2-1)} \left(\frac{\rho}{\rho_3}\right)^{(\gamma_3-1)}, & \rho \geq \rho_3, \end{cases} \quad (335)$$

where we set the initial sound speed $c_{s,0} = 2 \times 10^4 \text{ cm s}^{-1}$ and set $[\rho_1, \rho_2, \rho_3] = [10^{-13}, 10^{-10}, 10^{-3}] \text{ g cm}^{-3}$ 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 $10^{-10} \text{ g cm}^{-3}$, 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 50 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 7 different realisa-

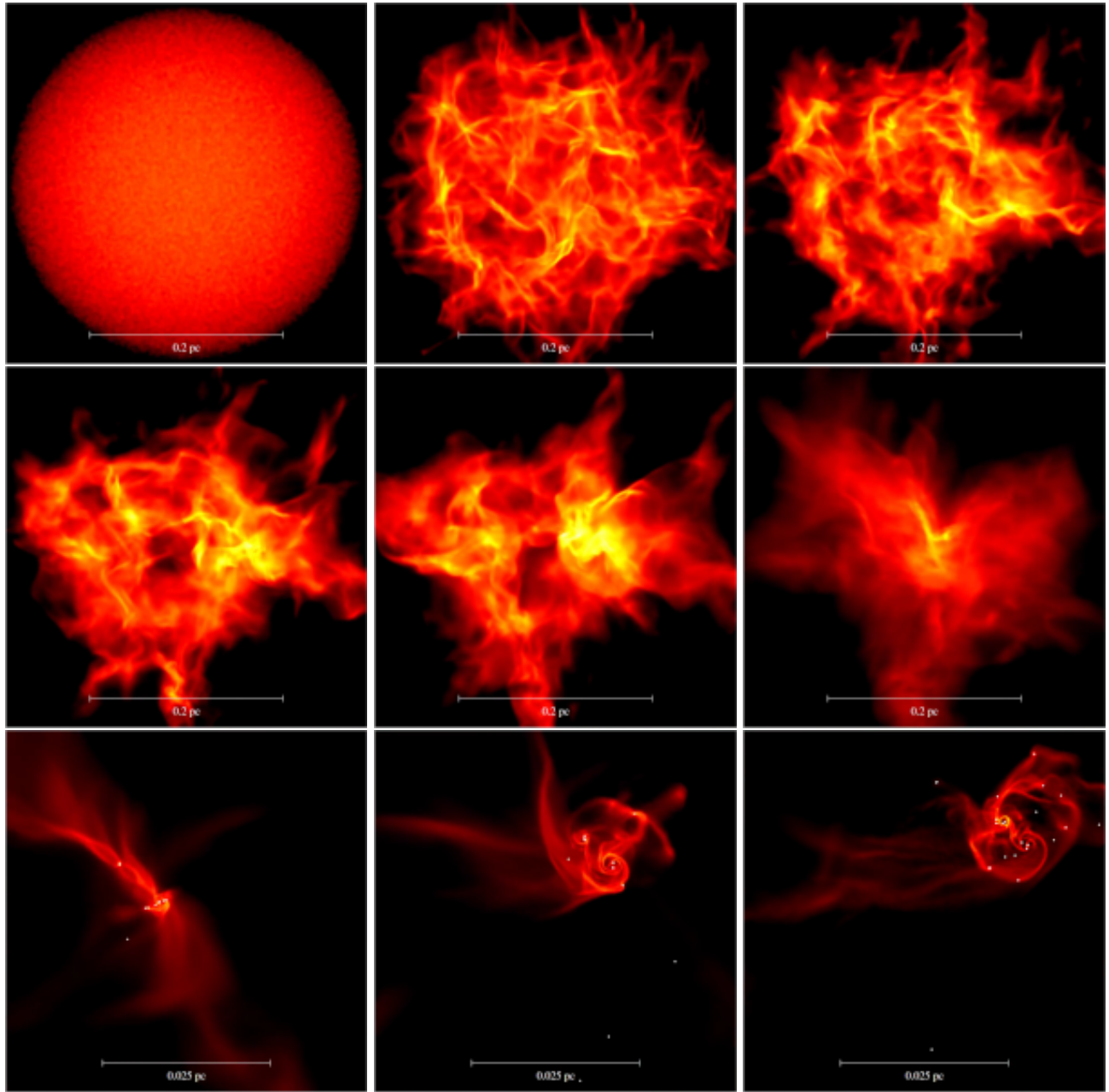


Figure 50. Star cluster formation with PHANTOM, showing snapshots of gas column density during the gravitational collapse of a $50M_{\odot}$ molecular cloud core, following [Bate et al. \(2003\)](#). Snapshots are shown every $0.2 t_{\text{ff}}$ (left to right, top to bottom), with the panels after $t > t_{\text{ff}}$ 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.

tions 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 ($500M_{\odot}$) calculation. The IMF produced with a barotropic equation of state 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. This is a known artefact of the barotropic equation of state ([Matzner & Levin,](#)

[2005](#); [Krumholz, 2006](#); [Bate, 2009b, 2012](#); [Offner et al., 2009](#); [Krumholz et al., 2010](#)), 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 et al., 2005](#); [Whitehouse & Bate, 2006](#)). This is not yet implemented in PHANTOM but it is high on the agenda.

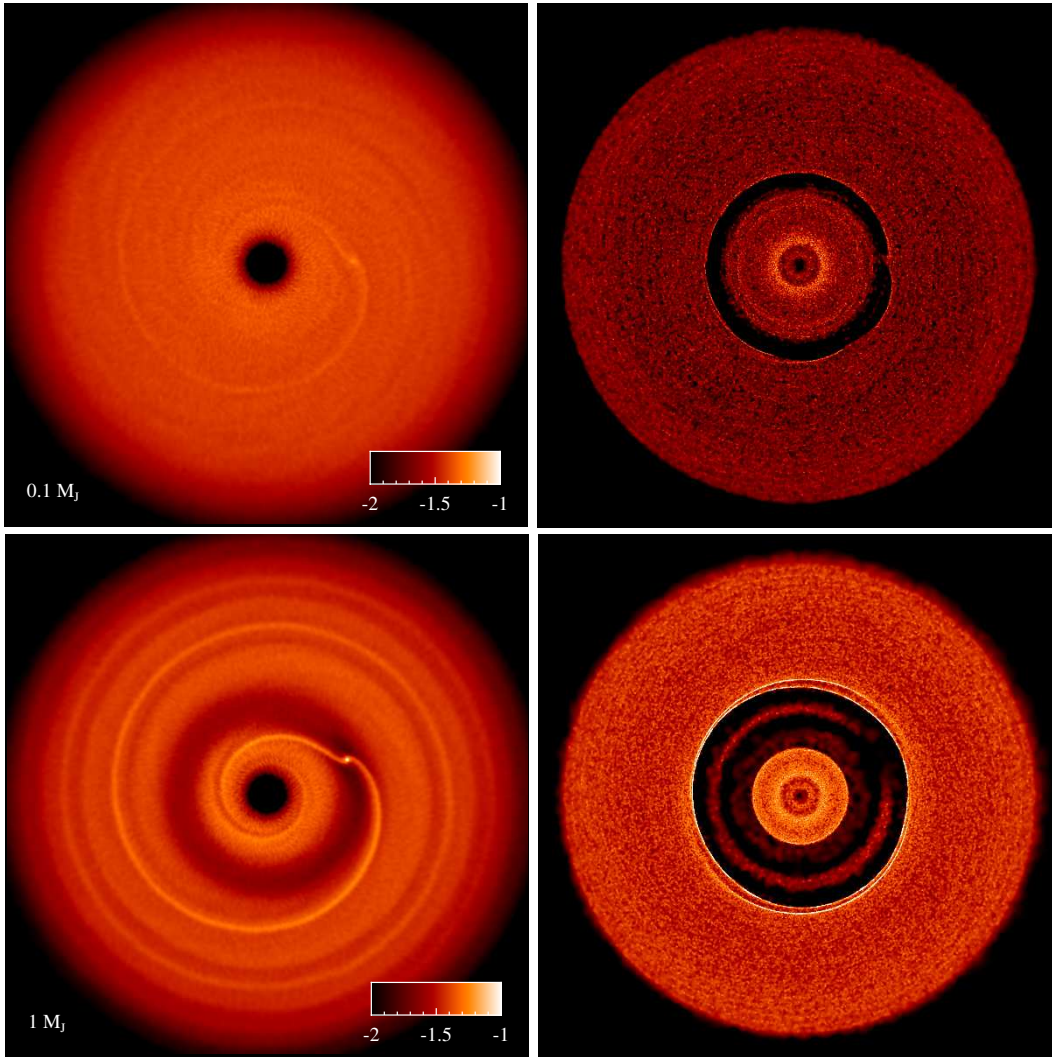


Figure 51. Gap opening in dusty protoplanetary discs with PHANTOM (from [Dipierro et al. 2016](#)), showing surface density in gas (left) and mm dust grains (right) in two simulations of planet-disc interaction with planet masses of $0.1 M_{\text{Jupiter}}$ (top) and $1 M_{\text{Jupiter}}$ (bottom) in 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.

6.3 Gap opening in dusty discs

Our third and 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](#)). Figure 51 shows the results of two calculations employing planets of mass $0.1 M_{\text{Jupiter}}$ (top row) and $1.0 M_{\text{Jupiter}}$ (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](#); [Papaloizou & Lin, 1984](#); [Lin & Papaloizou, 1986](#)), 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 51 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.

For the calculations we use 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 $2 \times 10^{-4} M_{\odot}$. 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 vertically isothermal

equation of state with $c_s \propto r^{-0.35}$ normalised such that $H/R = 0.05$ at 1 au. We use the minimum disc viscosity possible, setting $\alpha_{AV} = 0.1$.

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 very small dust grains (van Boekel et al. 2016; tracing the gas) to ALMA images of the mm dust emission (Andrews et al., 2016).

7 Summary

In this paper we have outlined the algorithms and physics currently implemented in the PHANTOM smoothed particle hydrodynamics and magnetohydrodynamics 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 hydrodynamics, continuing development of the dust algorithms, and an implementation of relativistic hydrodynamics. Rumours of the death of smoothed particle hydrodynamics in astrophysics have been greatly exaggerated.

Acknowledgments

PHANTOM is the result of many interactions over the years with numerous talented and interesting people. Particular mentions go to Joe Monaghan and Matthew Bate with whom DP has learned and discussed a great deal about SPH. DP would also like to thank Walter Dehnen, 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 4-year Future Fellowship (FT130100034) to DJP from the Australian Research Council (ARC), as well as funding via Discovery Projects DP130102078 (funding James Wurster) 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). 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 Supercom-

puting 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 Mark Twain for inspiration.

REFERENCES

- Abel T., 2011, *MNRAS*, **413**, 271
- Abel T., Anninos P., Zhang Y., Norman M. L., 1997, *New A*, **2**, 181
- Agertz O., et al., 2007, *MNRAS*, **380**, 963
- Alexiades V., Amiez G., Gremaud P.-A., 1996, *Communications in Numerical Methods in Engineering*, **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
- Balsara D. S., 1995, *J. Comp. Phys.*, **121**, 357
- Balsara D. S., 1996, *ApJ*, **465**, 775
- Balsara D. S., 1998, *ApJS*, **116**, 133
- Balsara D. S., Spicer D., 1999, *J. Comp. Phys.*, **148**, 133
- Barmin A., Kulikovskiy A. G., Pogorelov N. V., 1996, *J. Comp. Phys.*, **126**, 77
- Barnes D. J., Kawata D., Wu K., 2012, *MNRAS*, **420**, 3195
- Barranco J. A., 2009, *ApJ*, **691**, 907
- Bartosch L., 2001, *International Journal of Modern Physics C*, **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., 2012, *MNRAS*, **419**, 3115
- 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
- Bate M. R., Tricco T. S., Price D. J., 2014, *MNRAS*, **437**, 77
- Bauer A., Springel V., 2012, *MNRAS*, **423**, 2558
- Begeman K. G., Broeils A. H., Sanders R. H., 1991, *MNRAS*, **249**, 523
- Benz W., 1990, in Buchler J. R., ed., *Numerical Modelling of Nonlinear Stellar Pulsations Problems and Prospects*. p. 269

- 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
- Binney J., Tremaine S., 1987, *Galactic Dynamics*. Princeton, NJ, Princeton University Press
- Boily C. M., Kroupa P., Peñarrubia-Garrido J., 2001, *New A*, **6**, 27
- Bonet J., Lok T-S. L., 1999, *Computer methods in applied mechanics and engineering*, **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, preprint, ([arXiv:1611.09853v1](https://arxiv.org/abs/1611.09853v1))
- 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, *J. Comp. Phys.*, **75**, 400
- Brookshaw L., 1985, *PASA*, **6**, 207
- Brookshaw L., 1994, *Mem. Soc. Astron. Italiana*, **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, *J. Comp. Phys.*, **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
- Cerioni A., Lodato G., Price D. J., 2016, *MNRAS*, **457**, 939
- Cha S.-H., Whitworth A. P., 2003, *MNRAS*, **340**, 73
- Chabrier G., 2005, in Corbelli E., Palla F., Zinnecker H., eds, *Astrophys. Space Sci. Libr. Vol. 327, The Initial Mass Function 50 Years Later*. p. 41 ([arXiv:astro-ph/0409465](https://arxiv.org/abs/astro-ph/0409465))
- 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
- Chiang E., 2008, *ApJ*, **675**, 1549
- Chin S. A., Chen C. R., 2005, *Celestial Mechanics and Dynamical Astronomy*, **91**, 301
- Choi E., Kim J., Wiita P. J., 2009, *ApJS*, **181**, 413
- Chow E., Monaghan J. J., 1997, *J. Comp. Phys.*, **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., 2016a, preprint, ([arXiv:1608.05711](https://arxiv.org/abs/1608.05711))
- Coughlin E. R., Nixon C., Begelman M. C., Armitage P. J., 2016b, *MNRAS*, **459**, 3089
- Cox D. P., Gómez G. C., 2002, *ApJS*, **142**, 261
- Cullen L., Dehnen W., 2010, *MNRAS*, **408**, 669
- Cummins S. J., Rudman M., 1999, *J. Comp. Phys.*, **152**, 584
- Dahlburg R. B., Picone J. M., 1989, *Physics of Fluids B*, **1**, 2153
- Dai W., Woodward P. R., 1994a, *J. Comp. Phys.*, **111**, 354
- Dai W., Woodward P. R., 1994b, *J. Comp. Phys.*, **115**, 485
- Dai W., Woodward P. R., 1998, *ApJ*, **494**, 317
- Dedner A., Kemm F., Kröner D., Munz C.-D., Schnitzer T., Wesenberg M., 2002, *J. Comp. Phys.*, **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, *European Physical Journal Plus*, **126**, 55
- Diehl S., Rockefeller G., Fryer C. L., Riethmiller D., Statler T. S., 2015, *PASA*, **32**, 48
- Dipierro G., Price D., Laibe G., Hirsh K., Cerioni 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
- 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
- Elmegreen B. G., Scalo J., 2004, *ARA&A*, **42**, 211
- Epstein P. S., 1924, *Physical Review*, **23**, 710
- Escala A., Larson R. B., Coppi P. S., Mardones D., 2004, *ApJ*, **607**, 765
- Español P., Revenga M., 2003, *Phys. Rev. E*, **67**, 026705
- Eswaran V., Pope S. B., 1988, *Computers and Fluids*, **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, *Journal of Plasma Physics*, **65**, 29
- Fan L.-S., Zhu C., 1998, *Principles of Gas-Solid Flows*. Cambridge University Press
- Farr W. M., Bertschinger E., 2007, *ApJ*, **663**, 1420
- Federrath C., Banerjee S., 2015, *MNRAS*, **448**, 3297

- Federrath C., Klessen R. S., 2013, *ApJ*, **763**, 51
- 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, *Journal of Geophysical Research (Space Physics)*, **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, *International Journal of Modern Physics D*, **16**, 941
- Fulk D. A., Quinn D. W., 1996, *J. Comp. Phys.*, **126**, 165
- Furth H. P., Killeen J., Rosenbluth M. N., 1963, *Physics of Fluids*, **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, *J. Comp. Phys.*, **205**, 509
- Gardiner T. A., Stone J. M., 2008, *J. Comp. Phys.*, **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, *J. Comp. Phys.*, **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 T., et al., 2016, preprint, ([arXiv:1610.10053](https://arxiv.org/abs/1610.10053))
- 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, *Computer methods in applied mechanics and engineering*, **190**, 6641
- Hairer E., Lubich C., Wanner G., 2003, *Acta Numerica*, pp 399–450
- Haworth T. J., et al., 2016, *PASA*, **33**, e053
- Herant M., 1994, *Mem. Soc. Astron. Italiana*, **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, *Physics of Fluids*, **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, *J. Comp. Phys.*, **179**, 238
- Jacquet E., Balbus S., Latter H., 2011, *MNRAS*, **415**, 3591
- Jubelgas M., Springel V., Dolag K., 2004, *MNRAS*, **351**, 423
- Kaufmann T., Mayer L., Wadsley J., Stadel J., Moore B., 2007, *MNRAS*, **375**, 53
- Khoperskov S. A., Vasiliev E. O., Sobolev A. M., Khoperskov A. V., 2013, *MNRAS*, **428**, 2311
- Kitsionas S., et al., 2009, *A&A*, **508**, 541
- Konstandin L., Girichidis P., Federrath C., Klessen R. S., 2012, *ApJ*, **761**, 149
- Kritsuk A. G., Norman M. L., Padoan P., Wagner R., 2007, *ApJ*, **665**, 416
- Krumholz M. R., 2006, *ApJ*, **641**, L45
- Krumholz M. R., Cunningham A. J., Klein R. I., McKee C. F., 2010, *ApJ*, **713**, 1120
- 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 Journal on Scientific and Statistical Computing*, **7**, 591
- Le Bourlot J., Pineau des Forêts G., Flower D. R., 1999, *MNRAS*, **305**, 802

- Lecoanet D., et al., 2016, *MNRAS*, 455, 4274
- Lee A. T., Chiang E., Asay-Davis X., Barranco J., 2010, *ApJ*, 718, 1367
- Lemaster M. N., Stone J. M., 2008, *ApJ*, 682, L97
- Lense J., Thirring H., 1918, *Physikalische Zeitschrift*, 19, 156
- Lewis B. T., Bate M. R., Price D. J., 2015, *MNRAS*, 451, 288
- Lin D. N. C., Papaloizou J., 1986, *ApJ*, 307, 395
- 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, *J. Comp. Phys.*, 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
- 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., Saygin H., 2006, *New A*, 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., 2010, *MNRAS*, 406, 2279
- 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, *J. Comp. Phys.*, 229, 3916
- Miura H., Glass I. I., 1982, *Royal Society of London Proceedings Series A*, 382, 373
- Miyamoto M., Nagai R., 1975, *PASJ*, 27, 533
- Molina F. Z., Glover S. C. O., Federrath C., Klessen R. S., 2012, *MNRAS*, 423, 2680
- Monaghan J. J., 1989, *J. Comp. Phys.*, 82, 1
- Monaghan J. J., 1992, *ARA&A*, 30, 543
- Monaghan J. J., 1997, *J. Comp. Phys.*, 136, 298
- Monaghan J. J., 2000, *J. Comp. Phys.*, 159, 290
- Monaghan J. J., 2002, *MNRAS*, 335, 843
- Monaghan J. J., 2005, *Reports on Progress in Physics*, 68, 1703
- Monaghan J. J., 2012, *Annual Review of Fluid Mechanics*, 44, 323
- Monaghan J. J., Kocharyan A., 1995, *Computer Physics Communications*, 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, *J. Comp. Phys.*, 136, 41
- Müller K., Fedosov D. A., Gompper G., 2015, *J. Comp. Phys.*, 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
- Nixon C. J., 2012, *MNRAS*, 423, 2597
- Nixon C., Lubow S. H., 2015, *MNRAS*, 448, 3472
- 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 Franco J., Carraminana A., eds, *Interstellar Turbulence*. Cambridge: CUP, p. 218 ([arXiv:astro-ph/9810074](https://arxiv.org/abs/astro-ph/9810074))
- O'Sullivan S., Downes T. P., 2006, *MNRAS*, 366, 1329
- Offner S. S. R., Klein R. I., McKee C. F., Krumholz M. R., 2009, *ApJ*, 703, 131
- 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, *Computer Physics Communications*, 146, 188
- Orszag S. A., Tang C.-M., 1979, *J. Fluid Mech.*, 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
- Padoan P., Nordlund Å., 2011, *ApJ*, 730, 40
- Pandey B. P., Wardle M., 2008, *MNRAS*, 385, 2269

- Papaloizou J., Lin D. N. C., 1984, *ApJ*, **285**, 818
- Papaloizou J. C. B., Pringle J. E., 1983, *MNRAS*, **202**, 1181
- Passot T., Vázquez-Semadeni E., 1998, *Phys. Rev. E*, **58**, 4501
- 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, *Physics of Fluids B*, **3**, 29
- Plummer H. C., 1911, *MNRAS*, **71**, 460
- Politano H., Pouquet A., Sulem P. L., 1989, *Physics of Fluids B*, **1**, 2330
- Potekhin A. Y., Chabrier G., 2010, *Contributions to Plasma Physics*, **50**, 82
- Powell K. G., Roe P. L., Linde T. J., Gombosi T. I., de Zeeuw D. L., 1999, *J. Comp. Phys.*, **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: 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, *J. Comp. Phys.*, **227**, 10040
- Price D. J., 2010, *MNRAS*, **401**, 1475
- Price D. J., 2012a, *J. Comp. Phys.*, **231**, 759
- Price D. J., 2012b, *MNRAS*, **420**, L33
- Price D. J., Bate M. R., 2007, *MNRAS*, **377**, 77
- Price D. J., Bate M. R., 2008, *MNRAS*, **385**, 1820
- 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, *Reports on Progress in Physics*, **48**, 955
- Pringle J. E., 1981, *ARA&A*, **19**, 137
- Pringle J. E., 1992, *MNRAS*, **258**, 811
- Puri K., Ramachandran P., 2014, *J. Comp. Phys.*, **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, *New A Rev.*, **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, *Computers & Fluids*, **35**, 353
- Schmidt W., Federrath C., Hupp M., Kern S., Niemeyer J. C., 2009, *A&A*, **494**, 127
- Schoenberg I. J., 1946, *Q. Appl. Math.*, **4**, 45
- Schwarzschild K., 1916, *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin)*, 1916, Seite 189-196,
- Sedov L. I., 1959, *Similarity and Dimensional Methods in Mechanics*. New York; Academic Press
- Shakura N. I., Sunyaev R. A., 1973, *A&A*, **24**, 337
- Shu F. H., Galli D., Lizano S., Cai M., 2006, *ApJ*, **647**, 382
- Sod G. A., 1978, *J. Comp. Phys.*, **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, *New A*, **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., Norman M. L., 1992, *ApJS*, **80**, 791
- 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
- Syrovatskii S. I., 1981, *ARA&A*, **19**, 163
- Takahashi K., Yamada S., Yamada 2013, *Journal of Plasma Physics*, **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, *Proceedings of the Royal Society of London Series A*, **201**, 159
- Taylor G., 1950b, *Proceedings of the Royal Society of London Series A*, **201**, 175
- Taylor G. I., Green A. E., 1937, *Proceedings of the Royal Society of London Series A*, **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, *Computer Physics Communications*, **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., Swesty F. D., 2000, *ApJS*, **126**, 501
- Tóth G., 2000, *J. Comp. Phys.*, **161**, 605
- Toupin S., Braun K., Siess L., Jorissen A., Gail H.-P., Price D., 2015a, in *EAS Publications Series*. pp 173–174, doi:10.1051/eas/1571038
- Toupin S., Braun K., Siess L., Jorissen A., Price D., 2015b, in *Kerschbaum F., Wing R. F., Hron J., eds, Astronomical Society of the Pacific Conference Series Vol. 497, Why Galaxies Care about AGB Stars III: A Closer Look in Space and Time*. p. 225
- Tricco T. S., Price D. J., 2012, *J. Comp. Phys.*, **231**, 7214
- Tricco T. S., Price D. J., 2013, *MNRAS*, **436**, 2810
- Tricco T. S., Price D. J., Bate M. R., 2016a, *J. Comp. Phys.*, **322**, 326
- Tricco T. S., Price D. J., Federrath C., 2016b, *MNRAS*, **461**, 1260
- 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, *J. Chem. Phys.*, **97**, 1990
- Uhlenbeck G. E., Ornstein L. S., 1930, *Physical Review*, **36**, 823
- Valdarnini R., 2012, *A&A*, **546**, A45
- Vanaverbeke S., Keppens R., Poedts S., Boffin H., 2009, *Computer Physics Communications*, **180**, 1164
- Vanaverbeke S., Keppens R., Poedts S., 2014, *Computer Physics Communications*, **185**, 1053
- Vazquez-Semadeni E., 1994, *ApJ*, **423**, 681
- Verlet L., 1967, *Physical Review*, **159**, 98
- Violeau D., Leroy A., 2014, *J. Comp. Phys.*, **256**, 388
- Vogt D., Letelier P. S., 2011, *Monthly Notices of the Royal Astronomical Society*, **411**, 2371
- Wada K., Koda J., 2001, *PASJ*, **53**, 1163
- Wadsley J. W., Stadel J., Quinn T., 2004, *New A*, **9**, 137
- Wadsley J. W., Veeravalli G., Couchman H. M. P., 2008, *MNRAS*, **387**, 427
- 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
- Weingartner J. C., Draine B. T., 2001, *ApJ*, **563**, 842
- Wendland H., 1995, *Advances in Computational Mathematics*, **4**, 389
- 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., 1892, *Lady Windermere's Fan*, Act III
- 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
- Youdin A. N., Goodman J., 2005, *ApJ*, **620**, 459
- Yurin D., Springel V., 2014, *MNRAS*, **444**, 62
- Zurek W. H., Benz W., 1986, *ApJ*, **308**, 123
- de Val-Borro M., et al., 2006, *MNRAS*, **370**, 529
- van Boekel R., et al., 2016, preprint, (arXiv:1610.08939)
- von Neumann J., Richtmyer R. D., 1950, *J. Appl. Phys.*, **21**, 232