The Lagrangian hydrodynamics code MAGMA2

S. Rosswog

The Oskar Klein Centre, Department of Astronomy, AlbaNova, Stockholm University, SE-106 91 Stockholm, Sweden

Accepted 2020 August 24. Received 2020 August 21; in original form 2019 November 29

ABSTRACT

We present the methodology and performance of the new Lagrangian hydrodynamics code MAGMA2, a smoothed particle hydrodynamics (SPH) code that benefits from a number of non-standard enhancements. By default it uses high-order smoothing kernels and wherever gradients are needed, they are calculated via accurate matrix inversion techniques, but a more conventional formulation with kernel gradients has also been implemented for comparison purposes. We also explore a matrix inversion formulation of SPH with a symmetrization in the particle indices that is not frequently used. We find interesting advantages of this formulation in some of the tests, for example, a substantial reduction of surface tension effects for non-ideal particle setups and more accurate peak densities in Sedov blast waves. MAGMA2 uses artificial viscosity, but enhanced by techniques that are commonly used in finite-volume schemes such as reconstruction and slope limiting. While simple to implement, this approach efficiently suppresses particle noise, but at the same time drastically reduces dissipation in locations where it is not needed and actually unwanted. We demonstrate the performance of the new code in a number of challenging benchmark tests including, for example, multidimensional vorticity creating Schulz–Rinne-type Riemann problems and more astrophysical tests such as a collision between two stars to demonstrate its robustness and excellent conservation properties.

Key words: hydrodynamics – instabilities.

1 INTRODUCTION

A Lagrangian formulation of hydrodynamics is a natural choice for many astrophysical problems. Smoothed particle hydrodynamics (SPH, Lucy 1977; Monaghan 1977) is the most wide-spread Lagrangian method in astrophysics. It is entirely mesh-free and the equations can be symmetrized in a way so that mass, energy, momentum, and angular momentum are conserved by construction. As it is derived, SPH is entirely dissipationless and therefore needs to be augmented by additional measures to produce appropriate amounts of entropy in shocks. This is traditionally done via artificial viscosity, but also Riemann solver approaches have been explored (Inutsuka 2002; Cha & Whitworth 2003; Cha, Inutsuka & Nayakshin 2010; Murante et al. 2011; Puri & Ramachandran 2014).

In SPH one calculates the density via a smooth weighting of nearby particle masses and this smooth density estimate enters the calculation of pressures that drive the motion. The internal energy, in contrast, is evolved via a straightforward discretion of the Lagrangian energy conservation law and does therefore not involve a smoothing process. This potentially different inherent smoothness of both quantities can lead to unintended ‘pressure blips’ when setting up contact discontinuities. These blips cause surface tension effects that can suppress weakly triggered fluid instabilities (Agertz et al. 2007; McNally, Lyra & Passy 2012). Such effects can be counterbalanced by a careful setup of initial conditions with consistent smoothness, by alternative expressions for SPH volume elements (Ritchie & Thomas 2001; Saitoh & Makino 2013; Hopkins 2013; Rosswog 2015a; Cabezon, Garcia-Senz & Figueira 2017) or by adding artificial conductivity terms to smooth out sharp transitions in the internal energy (Price 2008; Valdarnini 2012). These issues have spurred further developments on Lagrangian hydrodynamics, many of which have employed techniques from finite-volume Eulerian hydrodynamics. The AREPO code (Springel 2010a), for example, tessellates space into Voronoi cells and evolves the hydrodynamic equations via a Riemann solver-based finite-volume strategy. Such finite-volume approaches, however, are not bound to Voronoi or other meshes and can actually also be applied to hydrodynamic schemes which use particles. Several such finite-volume particle schemes have been suggested in the applied mathematics literature (Ben Moussa, Lanson & Vila 1999; Vila 1999; Hietel, Steiner & Struckmeier 2000; Junk 2003), but they have only recently found their way into astrophysics (Gaburov & Nitadori 2011; Hopkins 2015; Hubber, Rosotti & Booth 2018) where they have delivered accurate results.

Also on the SPH side, there have been several new developments. Apart from the above-mentioned volume element improvements, substantially more accurate gradient estimates have been implemented (Gabric-Senz, Cabezon & Escartin 2012; Cabezon, Garcia-Senz & Escartin 2012; Rosswog 2015a). The perhaps most advanced SPH scheme to date (Frontiere, Raskin & Owen 2017) uses a reproducing kernel methodology (Liu, Jun & Zhang 1995) together with a sophisticated artificial dissipation scheme. They find good performance in a number of benchmark tests that are generally considered difficult for SPH. While the increasing cross-fertilization between different methods has been very beneficial, the boundaries...
between different numerical schemes and their naming conventions have started to blur.

In this paper, we describe the new Lagrangian hydrodynamics code MAGMA2, an SPH code that benefits from many improvements compared to more traditional SPH methods. It uses, for example, high-order kernels, calculates gradients via matrix-inversion techniques and uses slope-limited velocity reconstructions within an artificial viscosity approach. Many of these techniques have been scrutinized in a both Newtonian and special-relativistic context in Rosswog (2015a). The artificial viscosity approach that we are using is oriented at the recent work by Frontiere et al. (2017) which use fixed dissipation parameters, but reduce the effective viscosity by linearly reconstructing the velocities to interparticle mid-points. In this paper, we further expand on these ideas and throughout this paper we keep the dissipation parameters constant. We have also implemented a new way to steer time-dependent dissipation by monitoring local entropy violations and thus identifying ‘troubled particles’ that need more dissipation. This new approach is discussed in a separate study (Rosswog 2020) and will not be used here. The main purpose of this paper is to document MAGMA2 as a new simulation tool and to demonstrate its performance in a series of benchmark tests.

The paper is structured as follows. In Section 2, we describe the methodology used in MAGMA2 and Section 3 is dedicated to benchmark tests. We begin with smooth advection, then various shock tests, subsequently explore Kelvin–Helmholtz (KH) and Rayleigh–Taylor instabilities and combined, vorticity-creating shocks. We conclude our test series with astrophysical tests such as stellar collisions and tidal disruption events that demonstrate the versatility, robustness, and the excellent conservation properties in practical examples. Section 4 briefly summarizes our study.

2 METHODOLOGY

Many of the design choices in MAGMA2 are informed by our recent work (Rosswog 2015a) where we have carefully explored various SPH ingredients such as the calculation of gradients, dissipation triggers or kernel functions. Methods that have been described in detail elsewhere are only briefly summarized, while we focus here on those elements that are new.

For a general introduction to the SPH method, we refer to review articles (Monaghan 2005; Springel 2010b; Price 2012; Rosswog 2015b), a detailed step-by-step derivation of both Newtonian and relativistic SPH can be found in Rosswog (2009).

2.1 Ideal hydrodynamics

There are many different correctly symmetrized SPH discretizations of the ideal fluid equations. We have implemented three different SPH formulations into MAGMA2, which we briefly summarize below. For all of them we use high-order kernels and a novel dissipation scheme, but the formulations differ in the way gradients are calculated (two use matrix inversions, one kernel gradients), and in the symmetrization of the equations.

Figure 1. The artificial dissipation scheme uses the difference of the velocities reconstructed from particle a and particle b to their interparticle mid-point at r_{ab}.

2.1.1 Matrix inversion formulation I (MI1)

The standard SPH-approach of using direct gradients of (radial) kernel functions is a straightforward way to ensure exact conservation.1 The resulting gradients, however, are of moderate accuracy only (Abel 2011; Garcia-Senz et al. 2012; Cabezon et al. 2012; Rosswog 2015a,b; Cabezon et al. 2017). Improvements of many orders of magnitude in gradient accuracy can be achieved, see fig. 1 in Rosswog (2015a,b) at the (low) price of inverting a 3 x 3-matrix (in 3D).

The first matrix inversion formulation MI1 is the Newtonian limit of a special-relativistic formulation that was derived and extensively tested in Rosswog (2015a).2 It reads explicitly:

\[ \rho_a = \sum_b m_b W_{ab}(h_a), \]  
\[ \frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \left\{ \frac{P_a}{\rho_a^2} \mathbf{G}_a + \frac{P_b}{\rho_b^2} \mathbf{G}_b \right\}, \]  
\[ \left( \frac{d\mathbf{a}_a}{dt} \right) = \frac{P_a}{\rho_a^2} \sum_b m_b \mathbf{v}_{ab} \cdot \mathbf{G}_a, \]  

where \( \rho, \mathbf{v}, \mathbf{u}, h \) denote mass density, velocity, specific internal energy, and smoothing length. The particle mass is denoted by \( m \), \( P \) is the gas pressure, \( \mathbf{v}_{ab} = \mathbf{v}_a - \mathbf{v}_b \), and \( W \) is the chosen SPH kernel function. The gradient functions are given by

\[ (\mathbf{G}_a)^k = \sum_{d=1}^3 C_{kd}(r_a, h_a)(r_b - r_a)^d W_{ab}(h_a), \]  
\[ (\mathbf{G}_b)^k = \sum_{d=1}^3 C_{kd}(r_a, h_b)(r_b - r_a)^d W_{ab}(h_b), \]

where \( W_{ab}(h) = W(|r_a - r_b|, h) \). The ‘correction matrix’ \( C \) accounts for the local particle distribution and is calculated as

\[ C_{ki}(r, h) = \left( \sum_b \frac{m_b}{\rho_b} (r_b - r)^j (r_b - r)^l W(|r - r_b|, h) \right)^{-1}. \]

1See, for example, section 2.4 of Rosswog (2009) for detailed discussion of conservation in SPH. ‘Exact conservation’ means up to potential violations due to finite accuracy due to time integration or approximations of the gravitational forces, for example, due to the use of a tree. These latter terms, however, are usually fully controllable, though at some computational expense.

2The corresponding formulation was called \( \mathcal{F}_3 \) in the original paper.
The functions $G_a$ are antisymmetric with respect to the exchange of the involved position vectors and therefore allow for exact conservation in a similar way as the antisymmetric $\nabla_v W_{ab}$ in standard SPH. For more details, we refer to Rosswog (2015a). The practical conservation will be scrutinized below in a violent, off-centre collision between two stars, see Section 3.7.3, which shows that conservation with matrix inversion gradients is on par with standard SPH.

2.1.2 Matrix inversion formulation 2 (MI2)

The key property of the SPH equations to achieve conservation, is the antisymmetric $\nabla_v W_{ab}$ in standard SPH. For example, section 2.4 in Rosswog (2009) for detailed explanation. This can be achieved in an infinite number of different ways. One can, for example, start from the Lagrangian form of the Euler equations and use (Monaghan 1992)

$$\nabla P = \frac{P}{\rho^2} \nabla \left( \frac{1}{\rho^{1-\sigma}} \right) + \frac{1}{\rho^{2-\sigma}} \nabla \left( \frac{P}{\rho^{\sigma-1}} \right),$$  

where $\sigma$ is a real parameter. This leads to a momentum equation of the form

$$\frac{d\rho}{dt} = - \sum_b m_b \left[ \frac{P_a}{\rho_a^{1-\sigma}} + \frac{P_b}{\rho_b^{1-\sigma}} \right] \nabla_v W_{ab},$$

and

$$\frac{d\rho}{dt} = \frac{P_a}{\rho_a^{1-\sigma}} \sum_b m_b v_{ab} \nabla_v W_{ab},$$

as an energy equation. We implement these equations, as in Section 2.1.1, by replacing the kernel gradients by the functions equations (4) and (5). In the tests shown below we use $\sigma = 1$, so that final equations read

$$\frac{d\rho}{dt} = - \sum_b m_b \left\{ \frac{P_a + P_b}{\rho_a \rho_b} \right\} G_{ab},$$

(10)

$$\left( \frac{d\rho}{dt} \right) = - \sum_b m_b \left\{ \frac{P_a}{\rho_a h_a} \right\} v_{ab} \cdot G_{ab},$$

(11)

where $G_{ab} = 0.5(G_a + G_b)$ and we use the density estimate of equation (1). If one replaces the matrix inversion gradient functions $G$ by the corresponding kernel gradients one recovers the SPH equations that have been successfully used in the GASOLINE2 code (Wadsley, Keller & Quinn 2017).

While both of the matrix inversion formulations MI1 and MI2 yield very similar results on most benchmark tests, they show noticeable differences in some of them, for example in the Sedov explosion and in the ‘square test’, see below. Where the results are practically indistinguishable, we will only show one set of results, only where visible differences occur will we show plots for different formulations.

2.1.3 Standard kernel gradient formulation (stdGrad)

For comparison purposes we have also implemented a more conventional kernel-gradient formulation

$$\frac{d\rho}{dt} = - \sum_b m_b \left\{ \frac{P_a}{\rho_a} \nabla_v W_{ab}(h_a) + \frac{P_b}{\rho_b} \nabla_v W_{ab}(h_b) \right\},$$

(12)

\[ ^3 \text{We had also performed some experiments with } \sigma = 0 \text{ and } 2, \text{ which give good, but slightly worse results in Sedov explosions than } \sigma = 1. \]

The pressures $P$ are separated between two stars, see Section 3.7.3, which shows that conservation is excellent even without these correction terms.

2.2 Dissipation with reconstruction and slope limiting

2.2.1 Artificial viscosity

A common approach to deal with shocks is to enhance the physical pressure $P$ by additional viscous pressure terms $Q$ (von Neumann & Richtmyer 1950), that is, to replace the physical pressure, wherever it occurs, by $P + Q$. The pressures $Q$ are typically the sum two terms, one is proportional to the velocity jump between computational elements (either cells or particles) while the other is proportional to the square of the jump. In SPH, the expression for the viscous pressure at particle position $a$ is often written as (Monaghan & Gingold 1983)

$$Q_a = \mu_a \left( - \alpha c_s a_t \mu_a + \beta \mu_a^2 \right),$$

(14)

where the velocity jump is

$$\mu_a = \min \left( 0, \frac{\sum_b v_{ab}^2 \eta_0^b}{\eta_0^a + \epsilon_0^2} \right).$$

(15)

Here, $\alpha$, $\beta$, and $\epsilon$ are numerical parameters (typical values are 1, 2, and 0.1), $c_s$ is the sound speed and $\eta_0$ are separations between particles, de-dimensionalized via the smoothing length $h_a$. We follow the convention that particles are labelled usually by $a$ and $b$ and we use greek letters to denote summation indices (such as the $\delta$ above).

In SPH, it is common practice to use $\nu_{ab} = v_{ab}^2 - v_{ab}^2$ in equation (15), that is, one applies the velocity difference between the two particles. One can think of artificial dissipation as a simple way of solving an interparticle Riemann problem (Monaghan 1997). Translated into the language of finite-volume methods, the common practice of using straightforwardly the differences between particle velocities corresponds to a zeroth-order, or constant, velocity reconstruction. Zeroth-order reconstruction is known to introduce excessive dissipation in finite-volume methods, but higher order reconstructions yield successively less dissipative numerical schemes. This idea is translated here into an SPH context.

Most modern SPH implementations use time-dependent dissipation parameters to avoid excessive and unnecessary dissipation (Morris & Monaghan 1997; Rosswog et al. 2000; Cullen & Dehnen 2010; Rosswog 2015a; Wadsley et al. 2017; Price et al. 2018; Rosswog 2020). Recently, Frontiere et al. (2017), following ideas from a Eulerian finite-volume context (Christensen 1990), have explored an alternative refinement where the same form of artificial
pressure, equation (14), is used, but the velocity is (in their approach linearly) reconstructed to the particle mid-point. While still being a simple artificial viscosity scheme their approach showed excellent performance even with fixed dissipation parameters \( \alpha \) and \( \beta \).

Instead of using the difference of the two velocities, we quadratically reconstruct the velocities of particle \( a \) and \( b \) to their mid-point at \( r_{ab}^{\text{mid}} = 0.5(r_a + r_b) \), see Fig. 1. The velocities reconstructed from particle \( a \) to the mid-point read
\[
\tilde{v}_i^a = v_i^a + \Phi_{iab} \left[ (\partial_j v_j^b) \delta_j^i + \frac{1}{2} \left( \partial_j \partial_k v_j^b \delta_j^i \delta_k^m \right) \right]_{a},
\]
where \( \Phi_{iab} \) is a slope limiter, see below, the index at the square bracket indicates that the derivatives at the position of particle \( a \) are used and the increments from point \( a \) to the mid-point are \((\delta_j^i)_{a} = \frac{1}{2}(r_b - r_a)^j\). The reconstructed velocities from the \( b \)-side, \( \tilde{v}_i^b \), are calculated correspondingly, but now with derivatives at position \( b \) and increments \( \delta_j^i = -\delta_j^i \). In equation (15), we use the difference in the reconstructed velocities, that is, \( \tilde{v}_i^a - \tilde{v}_i^b \). We calculate the first derivatives as in Rosswog (2015a)
\[
(\partial_j v_j^a)_{a} = C_{jk}^{a} \sum_{b} m_{b} (v_{b} - v_{a}) (r_{b} - r_{a})^k W_{ab}(h_{a}),
\]
where \( C \) is the correction matrix from equation (6). For the second derivatives we proceed in two steps. First, we calculate an auxiliary gradient that does not require the density (Price 2004; Rosswog & Price 2007) and therefore can be conveniently calculated alongside the density loop
\[
(\partial_j v_j^a)^{\text{aux}} = D_{a}^{jk} \sum_{b} m_{b} (v_{b} - v_{a}) \partial_k W_{ab}(h_{a}),
\]
where the corresponding density-independent correction matrix is given by
\[
D_{a}^{jk} = \left\{ \sum_{b} m_{b} (r_{b} - r_{a})^j \partial_k W_{ab}(h_{a}) \right\}^{-1}.
\]
The second derivatives are then calculated by applying equation (18) to the auxiliary first derivatives \((\partial_j v_j^a)^{\text{aux}}\). While this procedure to calculate second derivatives still comes at some cost, it does not require an additional loop over the neighbour particles compared to just linear reconstruction.

We use a modification of van Leer’s slope limiter (van Leer 1974; Frontiere et al. 2017)
\[
\Phi_{iab} = \max \left[ \left( 0, \min \left[ 1, \frac{4 A_{ab}}{(1 + A_{ab})^2} \right] \right) \right], \text{ if } \eta_{ab} > \eta_{\text{crit}},
\]
with
\[
A_{ab} = \sum_{b} \frac{m_{b} (\partial_i v_{j}^b) x_{ab}^{i} x_{ab}^{j}}{\sum_{b} m_{b} (\partial_i v_{j}^b) x_{ab}^{i} x_{ab}^{j}},
\]
where the \( x_{ab}^{i} \) are the components of \( r_a - r_b \) and
\[
\eta_{ab} = \min(\eta_a, \eta_b) = \min \left( \frac{r_{ab}}{h_a}, \frac{r_{ab}}{h_b} \right) \text{ and } \eta_{\text{crit}} = \left( \frac{32 \pi}{3 N_{\text{nei}}} \right)^{1/3},
\]
with \( r_{ab} = |r_a - r_b| \) and \( N_{\text{nei}} \) being the number of neighbours for the chosen kernel, see Section 2.3. This prescription yields excellent, oscillation-free shock results as we demonstrate below.

2.2.2 Artificial conductivity

The analogy with Riemann solvers (Monaghan 1997; Chow & Monaghan 1997; Price 2008) also suggests to include thermal conductivity in the artificial dissipation terms. Such approaches have been found advantageous in certain shock problems (Noh 1987; Rosswog & Price 2007), but – as with all artificial dissipation terms – one has to ensure that no unwanted side effects are introduced. We add an artificial conductivity term to our energy equation
\[
\frac{d \epsilon_{\alpha}}{dt} = -\alpha_{\alpha} \sum_{b} m_{b} v_{\alpha gb}^{\text{sig}} \left( \tilde{u}_{\alpha} - \bar{u}_{\alpha} \right) \left( G_{\alpha} + G_{b} \right),
\]
where for the standard gradient version the average of the \( G \)-terms is replaced by \( (\nabla_{a} W_{ab}(h_{a}) + \nabla_{b} W_{ab}(h_{b}))/2 \). As for the velocities in the artificial viscosity, we use the differences in the quadratically reconstructed internal energies at the interparticle mid-point, \( \tilde{u}_{\alpha} \) and \( \bar{u}_{\alpha} \), which are calculated analogously to the velocities. For the conductivity signal velocity, we use
\[
v_{\alpha gb}^{\text{sig}} = (1 - iG) v_{\text{sig},G}^{ab} + iG v_{\text{sig},G}^{ab},
\]
where \( iG \) is a flag indicating whether gravity is used \((iG = 1)\) or not \((iG = 0)\)
\[
v_{\text{sig},G}^{ab} = \sqrt{\frac{|P_a - P_b|}{\rho_{ab}} \text{ and } v_{\text{sig},G}^{ab} = |\tilde{u}_{a} - \tilde{u}_{b}|.}
\]
These expressions are very similar to those used in the PHANTOM code (Price et al. 2018), but, as with the artificial viscosity terms, we are using here differences in the reconstructed velocities and internal energies (as indicated by tildes) rather that ‘flat’ differences and matrix-inversion-based gradient functions.

Throughout this study, we always use constant dissipation parameters \( \alpha = 1 \) and \( \beta = 2\alpha \). As we will demonstrate below, and consistent with a similar approach in a reproducing kernel context (Frontiere et al. 2017), the described velocity reconstruction produces excellent results and reduces drastically unwanted effects of artificial viscosity. Nevertheless, one may still try to additionally steer the dissipation parameter \( \alpha \) so that – if it safe to do so – it decays towards zero. A novel way of steering \( \alpha \) via violations of exact entropy conservation has recently been explored in a separate study (Rosswog 2020). It has also been implemented in MAGMA2, but all test shown here use fixed parameter values to demonstrate how powerful the artificial viscosity approach with reconstructed quantities is.

Concerning the conductivity, we very much stay on a conservative side with only small conductivity effects: we choose a low value of \( \alpha_{\alpha} = 0.05 \) as default choice, we use the difference of reconstructed \( \alpha \)-values in equation (24) and we use a switch of signal velocities, see equation (26), to protect a hydrostatic equilibrium configuration from being destroyed by conductivity. As shown below, none of the tests is substantially impacted by conductivity, not even the KH tests, see fig. 22 in Section 3.5.1, which were a major motivation tests is substantially impacted by conductivity, not even the KH effects are hardly noticeable in MAGMA2 simulations, we do see some positive effects in the long-term evolution of KH instabilities, where, without conductivity, the flow looks more granular. With our very conservative conductivity implementation we have not encountered any negative side effect, see for example, the test in Section 3.7.1. It is worth noting that Price et al. (2018) actually use a value of unity for \( \alpha_{\alpha} \) and report that they have not encountered artefacts for even this large value.

2.3 Kernel choice

SPH needs a kernel function to estimate densities and gradients, see equations (1)–(13). Cubic spline kernels (Schoenberg 1946;
Monaghan 2005) have traditionally been the standard choice in SPH. In recent years, however, a number of alternatives have been explored, see, for example, Cabezon, Garcia-Senz & Relano (2008), Read, Hayfield & Agertz (2010), Dehnen & Aly (2012), and Rosswog (2015a). In particular the family of Wendland (1995) kernels has received a fair amount of attention, because these kernels avoid the so-called pairing instability (Schüssler & Schmitt 1981) as pointed out by Dehnen & Aly (2012). While these kernels require a large neighbour number for an accurate density and gradient estimate, see figs 4 and 5 in Rosswog (2015a), they are exceptionally good at keeping an ordered particle distribution 4 and in suppressing subresolution noise. We had experimented with other high-order kernels (Rosswog 2015a) and in static density and gradient estimation tests on fixed particle lattices they actually showed better accuracy than the Wendland kernels. In dynamic tests, however, the Wendland kernels delivered more ordered particle distributions and were overall the better choice. To illustrate this, compare the high-order $W_{h,0}$ and a $C^0$ Wendland kernel ($WC6$) in figs 4 and 11 in Rosswog (2015a).

For these reasons, we choose the WC6 kernel as a default

\[ WC6(q) = \sigma \left( 1 - q \right)^6 \left( 32q^3 + 25q^2 + 8q + 1 \right), \]

where $\sigma = 1365/(512\pi)$ and $q = r/(2h)$. Note that for practical purposes/consistency with the other kernels in our module, we have written the kernel in a form so that it vanishes at $r/h = 2$. At every substep in the integration of the ordinary differential equations (ODE), we set the smoothing lengths so that there are exactly 300 contributing neighbours within the 2 $h_{bc}$-support of each particle $a$, see Section 2.5. This neighbour number is motivated by the tests shown in figs 4a and 5 in Rosswog (2015a). All tests in this paper use this combination of kernel and neighbour number.

### 2.4 Neighbour search and gravitational forces

We use a non-recursive tree that is based on ‘recursive coordinate bisection’ (RCB) (Gafton & Rosswog 2011) to search for neighbour particles and to calculate gravitational forces. The main idea is to start from a cuboid that contains all the particles and then one recursively splits the longest side of each cuboid so that it contains (to high accuracy) the same number of particles in each resulting daughter cell. The procedure is repeated until, on the deepest level of the tree, each cell contains no more than $N_h$ particles. Such cells are referred to as lowest level or ll-cells. We use $N_h = 12$, but as shown in our original paper (Gafton & Rosswog 2011), the results are not very sensitive to this choice. Our tree yields very simple integer relations between different nodes in the tree, so that nodes can be addressed via simple integer arithmetics. The tree is not walked down for each particle, but instead only for each ll-cell. This means that the number of required tree walks to find the neighbour particles is reduced by a factor of $\approx N_h$. As a result of a neighbour tree walk, the tree returns a list of potential neighbour particles (‘candidates’). Also for gravity, we only descend the tree for the centre of mass of each ll-cell, the forces at the particle positions within the ll-cell are obtained via a high-order Taylor expansion. As shown in our original paper, this tree build is for $4 \times 10^6$ particles approximately 30 faster than the Press tree (Benz et al. 1990). Neighbour search and gravity are at this particle number about a factor of 6 faster with our RCB with the discrepancy becoming increasingly larger for higher particle numbers $N$. Our RCB tree scales close to $O(N)$, while the Press tree scales like most tree methods proportional to $O(N\log N)$.

Similar to the hydrodynamic equations, one can also derive the gravitational accelerations consistently from a Lagrangian (Price & Monaghan 2007) and the resulting equations contain corrective, ‘gravitational grad-h terms’. Consistent with our treatment of hydrodynamics, we also neglect the grad-h terms here, so that the gravitational acceleration reads

\[ \left( \frac{dv_a}{dt} \right)_{\text{grav}} = - \frac{G}{r_a} \sum_b m_b \left( \frac{\dot{\varphi}_{ab}(h_a) + \dot{\varphi}_{ab}(h_b)}{2} \right) \hat{e}_{ab}, \]

where $\hat{e}_{ab} = (r_a - r_b)/|r_a - r_b|$ and the gravitational potential that is used for monitoring the total energy

\[ \Phi_a = G \sum_b m_b \varphi_{ab}(r_a). \]

The gravitational smoothing kernel for the force, $\varphi'$, and for the gravitational potential, $\varphi$, can be calculated directly from the density kernel $W$ via

\[ \varphi'(r, h) = \frac{4\pi}{r^2} \int_0^r W(r', h) r'^2 dr', \]

\[ \varphi(r, h) = 4\pi \left[ -\frac{1}{r} \int_0^r W(r', h) r'^2 dr' + \int_0^{r_0} W(r', h) r'^2 dr' - \int_0^{r_0} W(r', h) r'^2 dr' \right]. \]

For commonly used density kernels $W$ these integrals can be solved analytically. However, to make it possible to change $W$ with minimal modifications in the code, we calculate $\varphi'$ and $\varphi$ numerically and tabulate all kernels, so that for a different choice of $W$ automatically the consistent kernels $\varphi'$ and $\varphi$ are available for force and potential calculation. As stressed in our original paper, large speed gains have been obtained by mapping the tree variables into a ‘tree vector’ in exactly the order in which they are addressed during the tree walk.

### 2.5 Adoption of smoothing lengths

The so-called grad-h terms (Springel & Hernquist 2002; Monaghan 2002) have been introduced in Newtonian, special- (Rosswog 2010a) and general-relativistic SPH (Rosswog 2010b) to ensure exact conservation. For pragmatic reasons, however, we set them here to unity since a consistent update of smoothing length and density conservation. For pragmatic reasons, however, we set them here to unity since a consistent update of smoothing length and density conservation. For pragmatic reasons, however, we set them here to unity since a consistent update of smoothing length and density conservation. For pragmatic reasons, however, we set them here to unity since a consistent update of smoothing length and density conservation. For pragmatic reasons, however, we set them here to unity since a consistent update of smoothing length and density conservation. For pragmatic reasons, however, we set them here to unity since a consistent update of smoothing length and density conservation. For pragmatic reasons, however, we set them here to unity since a consistent update of smoothing length and density conservation. For pragmatic reasons, however, we set them here to unity since a consistent update of smoothing length and density conservation. For pragmatic reasons, however, we set them here to unity since a consistent update of smoothing length and density conservation. For pragmatic reasons, however, we set them here to unity since a consistent update of smoothing length and density conservation. For pragmatic reasons, however, we set them here to unity since a consistent update of smoothing length and density conservation. For pragmatic reasons, however, we set them here to unity since a consistent update of smoothing length and density conservation. For pragmatic reasons, however, we set them here to unity since a consistent update of smoothing length and density conservation.
is a challenge to assign a good value for the smoothing length $h$ of such front particles since a tiny change in $h$ can easily change the neighbour number by an order of magnitude. This can lead to problems with the size of neighbour lists or – in case counter measures are taken – this can lead to very erratic changes of the smoothing length of this particle and therefore to a substantial amount of numerical noise.

To avoid such problems, we assign to each particle an exact neighbour number before its force is calculated. Consistent with our SPH equations, we consider ‘neighbours’ of particle $a$ all those particles that are in the kernel support of $2h_a$, where $h$ is the smoothing length. We first build our RCB tree with smoothing lengths that are 10 per cent larger than those from the previous time-step. A neighbour tree walk then returns a substantially longer candidate list for each ‘lowest level cell’ than the desired neighbour number $n_{des}$ of each particle. From this candidate list the particle with the $(n_{des} + 1)$th largest distance, $d_{des + 1}^a$, to the particle of interest $a$ is selected via a partitioning algorithm (Press et al. 1992) and this distance sets the smoothing length: $h_a = 0.5d_{des + 1}^a$ (keep in mind that all our kernels are scaled so that they have a support size of $r/h = 2$).

Although this algorithm may seem at first sight very computationally expensive, it is actually not: if we calculate all the derivatives needed for an SPH simulation for the case of a star made of $5 \times 10^6$ particles, the assignment of the smoothing length takes only $\approx 7$ percent of the total time to calculate the derivatives (usually dominated by self-gravity). Apart from being very robust in extreme situations, this procedure has the additional advantage that the smoothing lengths evolve very smoothly and without introducing unnecessary noise.

### 2.6 Time integration

We perform all shown tests with the total variation diminishing (TVD) second-order Runge–Kutta (RK2) method (Gottlieb & Shu 1998)

\[
y^{n+1} = y^n + \Delta tf(y^n)
\]

\[
y^{n+1} = \frac{1}{2} \left[ y^n + y^* + \Delta tf(y^*) \right].
\]

If desired, the derivatives $f(y^*)$ can be ‘recycled’ for the next prediction step, so that only one derivative is effectively calculated.
per time-step. We have implemented this option as a simple switch, so that we can choose whether one or two derivatives are calculated per time-step. We have explored the accuracy of this ‘force recycling’ in practical tests, but have not found noticeable differences in any of them. Nevertheless, we use the TVD RK2 integrator with two derivative calculations in the tests presented below.

**Figure 4.** Advection of a smooth Gaussian pulse. Left: density distribution, and right: $L_1$-error as function of number of points in $x$-direction $n_x$ for the kernel gradient (stdGrad) and the matrix-inversion versions (MI1, MI2). All three converge very close to the expected second order.

**Figure 5.** Results of a surface tension test (density) similar to Saitoh & Makino (2013). The upper row refers to the SPH version with standard gradients, row two to the matrix inversion formulation 1, the lower row to the matrix inversion formulation 2 with a different symmetrization of the SPH equations.

The time-step is chosen as minimum from a ‘force-’ and ‘Courant-criterion’ (Monaghan 1992)

$$
\Delta t = C \min(\Delta t_f, \Delta t_C)
$$

where

$$
\Delta t_f = \min_a \left( \frac{h_a}{|J_a|} \right),
$$

$$
\Delta t_C = \min_a \left( \frac{h_a}{c_a + 0.06c_a (c_a + 2\bar{\mu}_a)} \right),
$$

where $\bar{\mu}_a = \max_b \left[ h_a \rho_a \rho_b (r_{ab}^2 + 0.01) \right]$ and for the pre-factor we choose $C = 0.2$. While being very simple and efficient, this time integration algorithm provides excellent numerical conservation of energy and angular momentum, as we will show below.

We restrict ourselves for now to a global time-step for all particles. Obviously, for problems with a large range of different possible time-steps among the particles, one can substantially reduce the computing time by allowing for individual time-steps. This comes, however, at a price. It makes the code more involved due to the time-step bookkeeping, it deteriorates the conservation properties and if the time-step bins between neighbouring particles are not restricted properly,
MAGMA2

Figure 7. Time evolution of a 3D Sedov blast ($256^3$ particles). Colour coded is the density in $xy$-plane, the black circle at the leading edge of the blast is the analytical shock position. Note the absence of any visible deviations from spherical symmetry.

Figure 8. Sedov blast ($200^3$ particles) for different SPH symmetrizations (matrix inversion formulation MI1 and MI2). The red lines denote the exact solution, and the black dots are the numerical MAGMA2 results.

particles can ‘be surprised’, say, by an approaching blast wave and this can lead to wrong results (Saitoh & Makino 2009). Individual time-steps make it also more cumbersome to remove particles, say, in an accretion process. For all these reasons, we stick for now with a global time-step, but if future problems will require it, we will implement an integration scheme with individual time-steps.

Figure 9. Comparison of the density peak height ($200^3$ particles) for the kernel- (stdGrad) and the two matrix-inversion formulations (MI1, MI2).

2.7 Implementation

MAGMA2 has been written from scratch in clean and modular FORTRAN 95/2003. We have paid particular attention to separate technical infrastructure (such as the tree for neighbour search) from the physical modules. MAGMA2 uses exclusively double precision and much attention has been payed to keep the ‘hot loops’ fast. This is of particular importance for MAGMA2 since due to the large neighbour number the density and hydrodynamic derivative loops become computationally very expensive. The strategy is to analyse in which order different variables are addressed in the expensive loops and to map them exactly in this order in ‘cache arrays’, very similar to how this is done within our RCB tree, see, for example, section 2.2.1 in Gafton & Rosswog (2011). To keep the code clean and (relatively) simple, we have so far only included a global time-step for all particles. At the current stage, MAGMA2 is parallelized
Figure 10. 3D Sedov blast of increasing resolution ($64^3$, $128^3$, and $256^3$ particles; M12 formulation), every single particle is plotted.

Figure 11. Sod shock test performed in 3D with $400 \times 24 \times 24$ particles. The upper two panels shows the particle distribution (density is colour-coded) at $t=0$ and 0.2. The four panels below show (upper left to lower right) density, velocity, pressure, and internal energy of all particles at $t=0.2$ together with the exact solution (red). All particles are plotted.

with OpenMP and is able to perform tests (global time-step, second-order Runge–Kutta and 300 neighbour particles) with $\sim 10^8$ SPH particles when no self-gravity is involved and a few $10^7$ otherwise (e.g. on Intel Skylake Gold compute nodes). Further performance improvements are a subject for the future.
3 TESTS

We begin with an advection test to measure the order of convergence in smooth flows. We then show a number of shocks, a KH and a Rayleigh–Taylor instability test. We further show combined tests where shocks go along with vorticity creation ('Schulz–Rinne tests') and we conclude with astrophysical tests, a stellar collision and a tidal disruption, that demonstrate MAGMA2's robustness for practical applications.

All of the tests shown below are performed with the full 3D code. '2D' tests are performed by simulating a slice that is thick enough for the central plane not to be affected by edge effects.

3.1 Initial conditions: the artificial pressure method

As pointed out earlier (Rosswog 2015a) good initial conditions are crucial for obtaining accurate results in SPH simulations, but unfortunately it is sometimes non-trivial to construct initial conditions with all the desired properties. The particles should be distributed with high regularity so that they guarantee a high interpolation accuracy (see e.g. section 3 of Rosswog 2015a for an accurate interpolation quality indicators). This is, however, not enough, since the particle distribution should also not contain any preferred directions, but simple lattices usually do. This may lead to artefacts such as the piling up of particles in a shock ('particle ringing') along the grid direction, see fig. 17 in Rosswog (2015a) for an illustration.

Setting up SPH-initial conditions for a pre-described non-constant density distribution can become challenging. The simplest approach is to start with some regular lattice, say a cubic lattice (CL) or (better) a 'close-packed' (CP) lattice, and assign particle masses so that the desired density is reproduced. For small density differences this delivers acceptable results, but for large density contrasts this vanishes and it stays at the reached position.

Once a particle has reached an optimal position, the pressure gradient 

\[ \sum b \rho_b \nabla W(\hat{a}) \]

and assign final particle masses 

\[ m_b \rho_b \]

where \( P_0 \) is a so far arbitrary base pressure and the max-function avoids negative pressures as they might otherwise occur for bad initial guesses. This means that particles with too low (too high) density estimates have smaller (larger) pressures and therefore more particles will move into (out of) this region. With these artificial pressures, we construct an SPH-type momentum equation (similar to equation 8)

\[ f_{ap}^p = -m \rho \sum_b \rho_b \nabla W(\hat{a}), \]

and we now choose a Courant-type time-step

\[ \Delta t_{APM} = \frac{\rho \Delta t}{\max a \rho_a \hat{a} \nabla W(\hat{a})}, \]

where we have used the polytropic relation for the sound speed 

\[ c_s = \sqrt{\frac{P}{\rho}}. \]

Note that the base pressure \( P_0 \) has dropped out. The pre-factor \( \xi \) is not critical for the iteration process, but has some impact on how quickly the desired density profile is reached. After some experimenting, we settled on a value of \( \xi = 0.5 \).

During the iteration process, we monitor the maximum and average density error and

\[ \delta P U = 1 - \sum b \rho_b W(\hat{a}), \]

and assign final particle masses \( m_b = \rho_b \Delta \hat{a}/\rho_a \), so that the setup contains particles that are to within per cent level of the same mass.

To illustrate the versatility of this method, we set up a non-trivial density profile with equal mass SPH particles. As density profile we choose

\[ \rho(x, y) = \Delta \rho \sum_{j=1}^4 \exp \left\{ -\frac{(x - x_j)^2 + (y - y_j)^2}{\sigma^2} \right\} + \rho_0, \]

where \( \rho_0 = 1, \Delta \rho = 5, \) and \( \sigma = 0.1 \). We start by placing 20 layers (in z-direction) of 100 × 100 particles in \([-0.2,1.2] \times [-0.2,1.2] \times [0,0,2\Delta] \) and we place six layers of 'frozen' particles (which are not updated in the iteration process) as boundary conditions in the surrounding volume. The centres of the density pulses, \( (x, y, z) \), are offset from the corners by 0.25. The particles are initially distributed on a CL and then randomized to erase the undesired lattice structure. The particle distribution (\( |z| < 0.02 \)) after 1000 APM iterations is shown in Fig. 3 (left-hand panel). The particles have settled into a
As a second example, we set up sharp density profile that the particles try to approximate (within the limits of their finite resolution and uniform mass) as well as possible. For the density profile, we choose
\[
\rho^T(x, y) = \begin{cases} 
\rho_0 + \Delta \rho & \text{inside the outer, but not in inner triangle} \\
\rho_0 & \text{else},
\end{cases}
\]
where the outer triangle refers to the interior of the points \((0.1, 0.1), (0.1, 0.9), (0.5, 0.9)\) and the inner triangle is given by the mid-points of the outer triangle’s sides and we use \(\rho_0 = 1\) and \(\Delta \rho = 5\), as before. The particles are initially placed as in the previous example and their distribution after 1000 APM iterations (\(|z| < 0.02\)) is shown in Fig. 3 (right-hand panel). Note that in all regions the particles have arranged into uniform glasses with sharp transitions between the different regions.

While the method is very flexible and powerful, it costs some computational effort: each iteration requires a density loop with the corresponding neighbour search. We therefore stick to the pragmatic approach that we use simpler particle setups where this delivers good results.

### 3.2 Smooth advection

In this test, we place a Gaussian density pulse of initial shape
\[
\rho_{\text{in}}(x, y) = (\rho_2 - \rho_1) \exp \left[ -\frac{(x - 0.5)^2 + (y - 0.5)^2}{\sigma^2} \right] + \rho_1,
\]
and uniform mass. The particles are initially placed as in the previous example and their distribution after 1000 APM iterations (\(|z| < 0.02\)) is shown in Fig. 3 (right-hand panel). Note that in all regions the particles have arranged into uniform glasses with sharp transitions between the different regions.

While the method is very flexible and powerful, it costs some computational effort: each iteration requires a density loop with the corresponding neighbour search. We therefore stick to the pragmatic approach that we use simpler particle setups where this delivers good results.

**Figure 12.** 3D strong blast wave test at \(t = 0.01\) (800 \(\times\) 24 \(\times\) 24 particles).

**Figure 13.** Circular blast wave problem 1 in 3D at \(t = 0.2\) (2003 particles). Upper row: MAGMA2 result for density, \(|v|\) and pressure. Lower row: corresponding quantities from MAGMA2 results compared with a reference solution (red line) obtained by the Eulerian-weighted average flux method (4003 grid cells, Toro 1999).
where $\rho_1 = 10^{-3}$, $\rho_2 = 1$, and $\sigma = 0.1$ in a (quasi-)2D box with $[0, 1] \times [0, 1]$, see Fig. 4 (left-hand panel). The gas with polytropic index $\Gamma = 5/3$ is represented by 20 CL-layers of $n_x \times n_y$ particles. The density profile is advected with uniform pressure $P_0 = 10^{-6}$ and velocity $v_x = 1$ through the box with periodic boundaries. After crossing the box once the $L_1$-error, $L_1 = \sum_{b=0}^{N} |\rho_l(b) - \rho_b|/N$, is measured. The $L_1$-results for different resolutions $n_x$ are plotted in the right-hand panel of Fig. 4 for all of our three variants. The measured...
using volume elements that are different from a sharp mass transition). Here, further progress could be obtained by not entirely free of surface tension effects (at least for this setup with both the standard gradient version (top row) and the MI1 formulation (2013), we use constant pressure energies according to \( \rho = \left[ \text{e} \right] \left( 2013 \right) \), and compare the performance of our three different SPH formulations. We place 40\(^3\) particles on a \( x \) with spacing \( \Delta \) within \([ -0.5, 0.5]\)\(^3\) and assign them masses \( m_a = \tilde{\rho}(r_a)\Delta^3 \) with a sharp transition according to \( \tilde{\rho}(x, y, z) = \begin{cases} 4 & \text{for } -0.25 < x, y, z < 0.25 \\ 1 & \text{otherwise} \end{cases} \)
to test for the presence of spurious forces. Note, however, that this is not how we would usually set up reliable simulation. We then calculate the density according to equation (1) and assign the internal energies according to \( u_a = P_0(\Gamma - 1)\rho_a \). Following Saitoh & Makino (2013), we use constant pressure \( P_0 = 2.5 \) everywhere.

The results for the different SPH formulations are shown in Fig. 5. Both the standard gradient version (top row) and the MI1 formulation show (with this somewhat pathological test setup) clear signs of surface tension. The MI2 formulation performs by far best, but is still not entirely free of surface tension effects (at least for this setup with a sharp mass transition). Here, further progress could be obtained by using volume elements that are different from \( V_b = m_b/\rho_b \) (Saitoh & Makino 2013; Hopkins 2013; Rosswog 2015a; Cabezon et al. 2017). We also show the impact of varying the conductivity signal speed, see Fig. 6. Clearly, \( v_{\text{sig, nG}} \) shows the better result, which argues for using the switch in equation (26). We note, however, that despite small possible surface tension effects both MI1 and MI2 formulations perform very well in the instability tests, see below.

### 3.4 Shocks

#### 3.4.1 3D Sedov–Taylor explosion

The Sedov–Taylor explosion test, a strong, initially point-like blast into a low-density environment, has an analytic self-similarity solution (Taylor 1950; Sedov 1959). For an explosion energy \( E \) and a density of the ambient medium \( \rho \), the blast wave propagates after a time \( t \) to the radius \( r(t) = \beta(ET\rho)^{1/5} \), where \( \beta \) depends on the adiabatic exponent of the gas \((\approx 1.15 \) in 3D for the \( \Gamma = 5/3 \) that we are using). Directly at the shock front, the density jumps by the blast explosion limit factor of \( \rho_2/\rho_1 = (\Gamma + 1)/(\Gamma - 1) = 4 \), where the numerical value refers to our chosen \( \Gamma \). Behind the shock the density drops quickly and finally vanishes at the centre of the explosion.

To set up the test numerically, we distribute a given number of SPH particles according to a CVT (Du, Faber & Gunzburger 1999) in the computational volume \([-0.5,0.5] \times [-0.5,0.5] \times [-0.5,0.5]\). While this produces already nearly perfect initial conditions, they can be further improved by additional sweeps according to equation (40). Even if the differences in the particle distributions are hard to see by eye, they still improve the outcome of the Sedov test. We use here 500 of such sweeps. Once the particle distribution is settled, we assign masses so that their density is \( \rho = 1 \). This is done in an iterative way where we first assign a guess value for the masses, then measure the resulting density via equation (1) and subsequently correct the particle masses. The iteration is stopped once the density agrees everywhere to better than 0.5 per cent with the desired value. The energy \( E = 1 \) is spread across a very small initial radius \( R \) and it is entirely distributed as internal energy, the specific internal energy \( u \) of the particles outside of \( R \) is entirely negligible \((10^{-10}\) of the central \( u \)). For the initial radius \( R \) we choose twice the interaction radius of the innermost SPH particle. Boundaries play no role in this test as long as the blast does not interact with them. We therefore place ‘frozen’ particles around the computational volume as boundary particles.

Fig. 7 shows the time evolution of the density for 256\(^3\) particles for our MI2-formulation (the other formulations look virtually identical). The overall agreement with the exact solution (shock position is indicated by a black circle at shock front) is excellent and there are no noticeable deviations from spherical symmetry. We show in Fig. 8 comparisons of the solutions (pressure, velocity, and density) obtained with the MI1 and MI2 formulations, the particle solutions are shown as black dots (downsampled by a factor of 10), the red lines indicates the exact solutions. Note in particular the absence of density and velocity oscillations in the wake of the shock. Such oscillations plague practically all (even modern) SPH simulations of this test, see, for example, Rosswog & Price (2007), Hu et al. (2014), Cabezon et al. (2017), Wadsley et al. (2017), and Frontiere et al. (2017), but they are virtually absent in our tests.\(^6\) We attribute this to our carefully constructed initial conditions together with the use of a Wendland kernel with 300 neighbour particles.

Despite this overall very good agreement, a closer inspection shows some interesting differences that are due to different SPH

\(^6\)The PHANTOM code paper (Price et al. 2018) does not show velocities, but some oscillations are visible in the densities.
Figure 18. Density evolution for weakly triggered KH instability at \((128 \times 128 \times 8), (256 \times 256 \times 8),\) and \((512 \times 512 \times 8)\) particles resolution.

Figure 19. Resolution dependence of the KH mode amplitude growth (MI1). As reference solution (solid, red) we use a simulation with the PENCIL code at a resolution of \(4096^2\). Note that all cases show a growth close to the reference solution, even at the lowest resolution \((128^2)\) particles. For comparison, traditional SPH simulations struggle with this only weakly triggered test, even at substantially higher resolution of \(512^2\) (McNally et al. 2012).

Symmetrizations (gradient accuracy plays a minor role in this test) see Fig. 8. The MI2-formulation seems to be substantially more sensitive to density variations which can be seen in a larger post-shock pressure variance, probably picking up on small density variations that are residuals of our iterative approach to set up the initial conditions. More importantly, MI2 reaches a noticeably larger density peak (apart from the symmetrization everything else is exactly the same), see Fig. 9. Note also that the SPH peak for the more commonly used symmetrizations (stdGrad and MI1) occurs at the shock-front while the MI2-case peaks between both lines indicating the exact solution. The MI2-symmetrization result also captures the flat central pressure profile in a better way.

Obviously, the peak height could easily be raised by using a lower order kernel with fewer neighbours, if one was willing to accept more post-shock noise (which we are not). For completeness, we also show Sedov tests (with MI2) with increasing resolution of \(64^3, 128^3,\) and \(256^3\) SPH particles within \([-0.5, 0.5]^3\), see Fig. 10.

3.4.2 3D Sod shock

The ‘Sod shock tube’ (Sod 1978) is a classic code test to ensure the correctness of a hydrodynamics code implementation. As initial conditions we use

\[
\begin{cases}
(\rho, v, P) = (1.000, 0, 0, 1.0) & \text{for } x < 0, \\
(0.125, 0, 0, 0.1) & \text{else}
\end{cases}
\]

and \(\Gamma = 5/3\). In most SPH papers, this test is only shown in 1D. Here, we perform a quasi-1D test with the 3D code, employing \(400 \times 24 \times 24\) particles placed in \([-0.5, 0.5] \times [-0.03, 0.03] \times [-0.03, 0.03]\). Here, even the simplest approach with particles placed on a uniform CL gives good results, see Fig. 11. The numerical results (all particles are shown) agree well with the exact solution, there is only a small velocity overshoot at the shock front and small over-/undershoots at one of the edges of the rarefaction region \((x \approx -0.05)\). The plot
shows the results of the MI1-formulation, there are no noteworthy differences between the different formulations for this test.

### 3.4.3 3D strong blast wave

A substantially more challenging version of the Sod test, the so-called strong blast wave test, is set up with initial conditions

\[
(r, v, P) = \begin{cases} 
(1.0, 0, 0, 0, 1000.0) & \text{for } x < 0 \\
(1.0, 0, 0, 0, 0.1) & \text{else}
\end{cases}
\]  

and with \( \Gamma = 1.4 \). Again we set up this test as quasi-1D with our 3D code in an analogous way to the Sod test with 800 \( \times \) 24 \( \times \) 24 particles between \([-0.5, 0.5]\) in \( x \)-direction. The result (obtained with MI1; no noteworthy differences for MI2 and stdGrad) at time \( t = 0.01 \) is shown in Fig. 12. Overall, the numerical result is in very good agreement with the exact solution, but there are small overshoot in density and dips in pressure and velocity at the contact discontinuity.
Figure 21. Growth of the KH mode amplitude for the 256×2-case where we vary methodological choices as in Fig. 20. The results for the M1 default choices are shown as black circles, the results with only linear reconstruction (blue square) grow at a similar rate, but slightly slower. Both the case with standard kernel gradients (green triangles) and the one using the cubic spline kernel (orange triangles) grow somewhat slower. The largest impact in this test, however, has the velocity reconstruction: if it is not applied and the standard SPH prescription is applied instead, the instability is suppressed (magenta diamonds). The symmetrization in the matrix inversion formulation does make a small difference in the growth rate, with M12 growing slightly faster.

3.4.4 Spherical blast wave 1

As another benchmark we use a 3D shock-tube problem. We follow Toro (1999) in the parameter choice (apart from a shift of the origin): the computational domain is $[-1, 1]^3$ and the initial conditions are chosen as:

$$
(\rho, v, P) = \begin{cases} 
(1.000, 0, 0, 0, 1.0) & \text{for } r < 0.5 \\
(0.125, 0, 0, 0, 0.1) & \text{else.}
\end{cases}
$$

The solution exhibits a spherical shock wave, a spherical contact surface travelling in the same direction and a spherical rarefaction wave travelling towards the origin. We show the Magma2 solution (2003 particles, CL; shown is M11, other formulations nearly identical) at time $t = 0.2$ in Fig. 13. We show in the upper row the density, $|v|$ and pressure at $t = 0.2$ in the XY-plane. In the lower row, we compare the SPH result ($|v| < 0.018, |z| < 0.018$) with a reference solution obtained by the Eulerian weighted average flux method with 4003 grid cells (Toro 1999). The SPH solution is essentially oscillation-free and in very close agreement with the reference solution. Only the sharp edges (e.g. in $|v|$ near $|x| \approx 0.3$) are somewhat smoothed out and there is a small oscillation in the pressure at the contact discontinuity.

3.4.5 Spherical blast wave 2

As a second spherical blast wave problem (Toro 1999), we start from

$$
(\rho, v, P) = \begin{cases} 
(1.0, 0, 0, 0, 2.0) & \text{for } r < 0.5 \\
(1.0, 0, 0, 0, 1.0) & \text{else.}
\end{cases}
$$

We show the numerical solution (2003 particles, CL; M11) at time $t = 0.2$ in Fig. 14. Again, the SPH result is in very good agreement with the (higher resolved) reference solution, only the inner density plateau is somewhat smeared out, and there is a small wiggle at the contact discontinuity. This could be cured, for example, by applying a larger amount of conductivity (keep in mind we only use a small value, $\alpha_u = 0.05$). In this test, all three the SPH formulations perform well, the matrix-inversion versions capture the central density plateau better and show a smaller overshoot than the kernel-gradient version, see Fig. 15. Note that in both circular blast wave problems the spherical symmetry is very well preserved, despite the initial particle setup on a CL.

3.4.6 Noh test

Next we consider the very challenging 3D Noh implosion test (Noh 1987) which has the reputation as a ‘code breaker’, since a number of established methods are not able to handle this test without breaking...
or simply producing wrong results (Liska & Wendroff 2003). The test is performed with a polytropic exponent of $\Gamma = 5/3$ and with initial conditions $[\rho_0, v_{i0}, P_0] = [1, -r_{ia}^i, 10^{-10}]$, where $r_{ia}^i$ is the $i$-component of the radial unit vector of particle $a$. This corresponds to the spherical inflow to a point and results in a self-similar shock moving outwards at a velocity $v_s = 1/3$. The shocked region ($r < v_s t$) has density $\rho_s = \rho_0[(\Gamma + 1)/(\Gamma - 1)]^{3/2}$. The pre-shock flow ($r > v_s t$) undergoes shockless compression according to $\rho(r, t) = \rho_0(1 - v_0 t/r)^2$.

The test is challenging for two reasons. First, the initial conditions contain an unresolved point of convergence that gives rise to the well-known ‘wall heating problem’ where the thermal energy overshoots at the origin while the density undershoots in order to strive for the correct pressure. This phenomenon lead to the original suggestion to apply an artificial conductivity (Noh 1987) to alleviate the problem. The second difficulty is that the adiabatic pre-shock compression should not produce entropy which is a serious challenge for most artificial viscosity schemes.

We setup this challenging test similar to Sedov test case: we distribute $2.3 \times 10^7$ particles according to a CVT (Du et al. 1999) and subsequently perform 1000 correction sweeps according to equation (40). We show in Fig. 16 the density for this test (for the MI1 equation set; the other equation sets give very similar results), $\rho$, $v$, and $P$ are shown in Fig. 17 compared with the exact solution. At the centre the ‘wall heating problem’ with a substantial density undershoot is encountered and the pressure is about 4 per cent below the theoretical value. Nevertheless, compared to most existing methods MAGMA2 performs rather well in this challenging test. The wall heating effect could be alleviated by employing larger $\alpha_u$-values. The PHANTOM code (Price et al. 2018), for example, uses $\alpha_u = 1$ and the authors report to have not found serious artefacts from this large value. However, given the somewhat pathological initial conditions with its unresolved point of convergence, we are not excessively worried about the encountered wall-heating problem (that is shared by most other methods) and do not see this as a strong incentive to increase the dissipation parameter $\alpha_u$ beyond the chosen, low value.

### 3.5 Instabilities

#### 3.5.1 Kelvin–Helmholtz instability
Table 1. Initial data for the Schulz–Rinne-type 2D Riemann problems.

| Variable | NW    | NE    | SW    | SE    |
|----------|-------|-------|-------|-------|
| \(\rho\)  | 0.5323 | 1.5000 | 0.1380 | 0.5323 |
| \(v_x\)   | 1.2060 | 0.0000 | 1.2060 | 0.0000 |
| \(v_y\)   | 0.0000 | 0.0000 | 1.2060 | 0.0000 |
| \(P\)     | 0.3000 | 1.5000 | 0.0290 | 0.3000 |

SR2; contact point: \((- 0.15, - 0.15)\)

| Variable | NW    | NE    | SW    | SE    |
|----------|-------|-------|-------|-------|
| \(\rho\)  | 0.5065 | 1.1000 | 1.1000 | 0.5065 |
| \(v_x\)   | 0.8939 | 0.0000 | 0.8939 | 0.0000 |
| \(v_y\)   | 0.0000 | 0.0000 | 0.8939 | 0.0000 |
| \(P\)     | 0.3500 | 1.1000 | 1.1000 | 0.3500 |

SR3; contact point: \((0, 0, 0)\)

| Variable | NW    | NE    | SW    | SE    |
|----------|-------|-------|-------|-------|
| \(\rho\)  | 2.0000 | 1.0000 | 1.0000 | 3.0000 |
| \(v_x\)   | -0.7500 | -0.7500 | 0.7500 | 0.7500 |
| \(v_y\)   | 0.5000 | -0.5000 | 0.5000 | -0.5000 |
| \(P\)     | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

SR4; contact point: \((0, 0, 0)\)

| Variable | NW    | NE    | SW    | SE    |
|----------|-------|-------|-------|-------|
| \(\rho\)  | 0.5313 | 1.0000 | 0.8000 | 0.5313 |
| \(v_x\)   | 0.8276 | 0.1000 | 0.1000 | 0.1000 |
| \(v_y\)   | 0.0000 | 0.0000 | 0.0000 | 0.7276 |
| \(P\)     | 0.4000 | 1.0000 | 0.4000 | 0.4000 |

SR5; contact point: \((0, 0, 0)\)

| Variable | NW    | NE    | SW    | SE    |
|----------|-------|-------|-------|-------|
| \(\rho\)  | 1.0000 | 0.5313 | 0.8000 | 1.000 |
| \(v_x\)   | 0.7276 | 0.0000 | 0.0000 | 0.0000 |
| \(v_y\)   | 0.0000 | 0.0000 | 0.0000 | 0.7262 |
| \(P\)     | 1.0000 | 0.4000 | 1.0000 | 1.0000 |

SR6; contact point: \((0, 0, 0)\)

where \(\rho_1 = 1, \rho_2 = 2, \rho_m = (\rho_1 - \rho_2)/2\), and \(\Delta = 0.025\). The velocity is set up as

\[
v(y) = \begin{cases} v_1 - v_m e^{(y - 0.25)/\Delta} & \text{for } 0.00 \leq y < 0.25 \\ v_2 + v_m e^{(0.25 - y)/\Delta} & \text{for } 0.25 \leq y < 0.50 \\ v_2 + v_m e^{(y - 0.75)/\Delta} & \text{for } 0.50 \leq y < 0.75 \\ v_1 - v_m e^{(0.75 - y)/\Delta} & \text{for } 0.75 \leq y < 1.00 \end{cases}
\]

(51)

with \(v_1 = 0.5, v_2 = -0.5, v_m = (v_1 - v_2)/2\), and a small velocity perturbation in \(y\)-direction is introduced as \(v_i = 0.01 \sin (2\pi u/\lambda)\) with the perturbation wavelength \(\lambda = 0.5\). In the linear regime, the instability grows on a characteristic time-scale of

\[
\tau_{KH} = \frac{\rho_1 + \rho_2}{\sqrt{\rho_1 \rho_2 [v_1 - v_2]^2}},
\]

(52)

with \(\tau_{KH} \approx 1.06\) for the chosen parameters. The test is performed with a polytropic equation of state with exponent \(n = 5/3\).

KH instabilities occur in shear flows with a perturbed interface. They play an important role in astrophysics and occur in a broad range of environments, for example, in mixing processes in novae (Casanova et al. 2011), amplification of magnetic fields in neutron star mergers (Price & Rosswog 2006; Giacomazzo et al. 2015; Kiuchi et al. 2015), or planetary atmospheres (Johnson, Wing & Delamere 2014), to name just a few. Traditional versions of SPH, however, have been shown to struggle with weakly triggered KH instabilities (Agerz et al. 2007; McNally et al. 2012). We focus here on a test setup in which traditional SPH has been shown to fail, even at rather high resolution in 2D, see McNally et al. (2012). We follow the latter paper in setting up the test with the only difference that we use our full 3D code to perform the test in quasi-2D. To this end, we set up a thin 3D slice with \(N \times N \times 20\) particles (referred to as ‘\(N^2\)’), for simplicity initially placed on a CL. Periodic boundary conditions are obtained by placing appropriate particle copies outside of the ‘core’ volume. The test is initialized as:

\[
\rho(y) = \begin{cases} \rho_1 - \rho_m e^{(y - 0.25)/\Delta} & \text{for } 0.00 \leq y < 0.25 \\ \rho_2 + \rho_m e^{(0.25 - y)/\Delta} & \text{for } 0.25 \leq y < 0.50 \\ \rho_2 + \rho_m e^{(y - 0.75)/\Delta} & \text{for } 0.50 \leq y < 0.75 \\ \rho_1 - \rho_m e^{(0.75 - y)/\Delta} & \text{for } 0.75 \leq y < 1.00 \end{cases}
\]

(50)

Downloaded from https://academic.oup.com/mnras/article/498/3/4230/5879700 by university of winnipeg user on 15 March 2022
inversion formulations, but MI2 is even closer to the high-resolution reference solution (open circles versus filled circles in Fig. 21).

Obviously, the growth rates are in good agreement with the reference solution, even at very low resolution and one may wonder how important the thermal conductivity is for this result. To find out, we perform the following experiment. We set up $128^2$ particles with the APM,\(^8\) Section 3.1, and run for each simulation the test once with default parameters and once with default parameters, but without conductivity. As can be seen in Fig. 22, the impact of conductivity is rather small and it seems that (apart from the reconstruction) the largest impact is made by the gradient accuracy:

\(^8\)We hardly see a difference for a grid setup.

both matrix inversion formulations (columns 3–6) show even at this low resolution a healthy growth, rather independent of conductivity, while the standard gradient version struggles and only grows with some noticeable delay, both with and without conductivity (columns 1 and 2).

3.5.2 Rayleigh–Taylor instability

The Rayleigh–Taylor instability is a standard probe of the subsonic growth of a small perturbation. In its simplest form, a layer of density $\rho_t$ rests on top of a layer with density $\rho_b < \rho_t$ in a constant acceleration field, for example, due to gravity. While the denser fluid sinks down, it develops a characteristic, ‘mushroom-like’ pattern. Simulations with traditional SPH implementations have shown only
retarded growth or even a complete suppression of the instability (Abel 2011; Saitoh & Makino 2013).

As before, we adopt a quasi-2D setup and use the full 3D code for the evolution. We place the particles on a CL in the XY-domain \([-0.25, 0.25]\) \([0, 1]\) and we use eight layers of particles in the Z-direction. Similar to Fontier et al. (2017), we use \(\rho_{\text{f}} = 2\), \(\rho_{\text{b}} = 1\), a constant acceleration \(g = -0.5\delta_t\), and

\[
\rho(y) = \rho_{\text{b}} + \frac{\rho_{\text{f}} - \rho_{\text{b}}}{1 + \exp[-(y - y_t)/\Delta]} \tag{53}
\]

with transition width \(\Delta = 0.025\) and transition coordinate \(y_t = 0.5\). We apply a small velocity perturbation to the interface

\[
v_y(x, y) = \delta v_y,0 [1 + \cos(8\pi x)][1 + \cos(5\pi(y - y_t))] \tag{54}
\]

for \(y\) in \([0.3, 0.7]\) with an initial amplitude \(\delta v_{y,0} = 0.025\), and use a polytropic equation of state with exponent \(\Gamma = 1.4\). The equilibrium pressure profile is given by

\[
P(y) = P_0 - g\rho(y)[y - y_t] \tag{55}
\]

with \(P_0 = \rho_0\Gamma\), so that the sound speed is near unity in the transition region. To enforce boundary conditions we add 10 rows of extra particles (\(y > 1\) and \(y < 0\)) that we ‘freeze’ at the initial conditions and we use periodic boundary conditions elsewhere.

We show in Fig. 23 the results of several simulations at a time of \(t = 4\). The first panel shows the result for a simulation that uses the standard SPH approach with kernel gradients and a XY-resolution of 128 \(\times\) 256 particles which overall performs reasonable well. The second to fourth panels show the results for everything else being the same (MI1 formulation), but using matrix-inversion kernels instead. This version delivers finer resolved/less diffusive density structures and a larger plunge depth. With increasing resolution the density transitions become sharper and more substructure appears, but the plunge depth remains the same.

A comparison between the MI1 and MI2 formulation, see Fig. 24, demonstrates that the latter shows a larger amount of mixing, consistent with the results from Section 3.3.

3.6 Complex shocks with vorticity creation

A set of challenging 2D benchmark tests has been suggested by Schulz-Rinne (1993). They are constructed in such a way that four constant states meet at one corner and the initial values are chosen so that one elementary wave, either a shock, a rarefaction or a contact discontinuity appears at each interface. During the subsequent evolution complex wave patterns emerge for which no exact solutions are known. These tests are considered challenging benchmarks for multidimensional hydrodynamics codes (Schulz-Rinne 1993; Lax & Liu 1998; Kurganov & Tadmor 2002; Liska & Wendroff 2003). Such tests are rarely shown for SPH codes, in fact, we are only aware of the work by Puri & Ramachandran (2014) who show results for one such shock test in a study of Godunov SPH with approximate Riemann solvers.

Here, we investigate six such configurations. Since our code is intrinsically 3D, we simulate, as before, a slice thick enough so that the mid-plane is unaffected by edge effects (we use 10 particle layers in Z-direction). We use 660 \(\times\) 660 particles in the XY-plane arranged on a hexagonal lattice between \([x_c - 0.5, x_c + 0.5] \times [y_c - 0.5, y_c + 0.5]\), \((x_c, y_c)\) being the contact point of the quadrants, and we use a polytropic exponent \(\Gamma = 1.4\) in all of the tests. We refer to these Schulz–Rinne-type problems as SR1–SR6 and give their initial parameters for each quadrant in Table 1. These test problems correspond to configuration 3, 4, 5, 6, 11, and 12 in the labelling convention of Kurganov & Tadmor (2002). Our results (MI1) are shown in Fig. 25.

Test SR1 is the only one of the Schulz–Rinne tests that has to our knowledge been tackled with (Godunov-)SPH (Puri & Ramachandran 2014). Their results show that existing SPH implementations struggle to resolve the mushroom-like structure along the diagonal (roughly at \(x \approx y \approx 0.2\) in our figure, upper left panel) and depending on the chosen approximate Riemann solver serious artefacts appear. Our results, in contrast, show crisp transitions between the different regions, well developed ‘mushrooms’ and they look overall similar to those found with established Eulerian methods, see, for example, Liska & Wendroff (2003), their fig. 4.1.

In test SR2 straight 1D shocks separating constant states and two curved shocks bordering a lens-shaped high-density/-pressure region occur. The result should be symmetric with respect to the lens axis and the MAGMA2 results do not show any noticeable deviation from perfect symmetry. Test SR3 yields a lense-shaped central region with two vortex structures occurring at the upper left and lower right part of the shown domain. Again, our result at \(t = 0.23\) closely resembles those in the literature, for example, Lax & Liu (1998), their fig. 5.

This is also true for the remaining tests, SR 4 can be compared, for example, with fig. 4.1 in Liska & Wendroff (2003), SR 5 with fig. 11 in Lax & Liu (1998), and SR6 with fig. 4.4 in Liska & Wendroff (2003). Note that all tests show a high (though not perfect) degree of symmetry which – with freely moving particles – is not actively enforced. Overall, the tests are in very good agreement with the Eulerian results found in the literature (Lax & Liu 1998; Liska & Wendroff 2003) and they look crisp and noise free. Note in particular the appearance of mushroom-like structures (in panels 1, 5, and 6) which are usually considered a challenge for SPH methods. Some of the weak straight lines in the panels, however, are considered spurious. But these artefacts are shared by the majority of methods found in the literature.
3.7 Astrophysical applications

In this last section, we show tests that are close to astrophysical applications. The purpose of these tests is to demonstrate the performance of the dissipation scheme, measure the numerical conservation in a relevant example and to show robustness and geometric flexibility. We did not find noteworthy differences between different SPH formulations in these tests and, unless explicitly stated otherwise, we show the MI1-results.

3.7.1 Oscillating white dwarf

As another experiment, we take a relaxed $\Gamma = 5/3$-polytropic star that represents a model for a 0.3 M$_\odot$ white dwarf (WD). We use only 10K SPH particles and provide them with a radial velocity $v_a = v_0 r_a$, where we choose $v_0 = 0.02$. As before, all tests use constant dissipation parameters of $\alpha = 1$. To avoid dissipation from other sources, we run these tests with low tolerance for the tree accuracy ($\Theta = 0.1$) and a time integration prefactor $C = 0.1$. The evolution of the oscillations in the gravitational energy are shown Fig. 26. As blue line we show the standard SPH approach, that is, without velocity reconstruction, the red line shows our default choice of methods and parameters and the black, open circles show the result for the default choices, but with $\alpha_u = 0$. Consistent with the experiments in the KH instability test, see Fig. 20, we find a massive suppression of unwanted dissipation when velocity reconstruction is employed. As intended, artificial conductivity does not switch on in this test problem, the results with $\alpha_u = 0$ lie nearly exactly on top of the $\alpha_u = 0.05$ (default).

3.7.2 Collapse of isothermal sphere

The collapse of an initially isothermal cloud is another frequently performed complex code test (Evrard 1988; Hernquist & Katz 1989; Steinmetz & Müller 1993; Dave, Dubinski & Hernquist 1997; Springel, Yoshida & White 2001; Wadsley, Stadel & Quinn 2004; Cabezon et al. 2017; Price et al. 2018) that tests for the coupling of gravity and hydrodynamics. The test starts with a gas cloud at rest that collapses under its own gravitational, then forms a shock, bounces back with a shock wave moving outward until the system settles into a virial equilibrium. This benchmark tests the transformation between different forms of energy: initially mostly gravitational, then kinetic and finally thermal.

We prepare the setup according to the parameters of Evrard (1988) with

$$\rho(r) = \frac{M(R)}{2\pi R^2 r},$$

(56)

where the initial cloud radius is $R = 1$ and the mass $M = 1$. Similar to other tests, we set up $10^6$ SPH particles according to a CVT (Du et al. 1999) and subsequently perform 5000 sweeps according to equation (40) to further improve the initial particle distribution. We
then assign masses so that the initial density profile is reproduced, set the internal energy to $u = 0.05\frac{GM}{R}$ and use a polytropic exponent of $\Gamma = 5/3$.

As a first step, we compare the MAGMA2 result with `standard SPH choices', specifically (i) a cubic spline kernel with 50 neighbour particles, (ii) constant dissipation ($\alpha = 1, \beta = 2$) without reconstruction, and (iii) standard kernel gradients to calculate derivatives. Note that this is different from what we had abbreviated before as `stdGrad': the latter uses kernel gradients, but all the other benefits of MAGMA2. The results of the MI2 formulation (with all default choices) and the `standard SPH choices' for a low resolution case with $10^5$ SPH particles is shown in Fig. 27 at $t = 0.77$. Compared to the reference solution (Steinmetz & Müller 1993), our low-resolution MI2 result shows a velocity overshoot at the shock (left-hand panel), but otherwise agrees well. The `standard SPH choice' version, in contrast, shows a fair amount of spurious entropy production (middle panel), so that the shock is broadly smeared out (left- and right-hand panels) and actually sitting at too large a radius.

Thus, the MAGMA2 results are a major improvement over traditional SPH choices. Among our different SPH variants we only find very minor differences in this test. We show a case with $10^6$ SPH particles, prepared as before, in Fig. 28 together with the reference solution (Steinmetz & Müller 1993). Note that all particles are plotted in the figure. All three formulations are nearly perfectly spherically symmetric and agree very well with the reference solution. Only at the shock front there is some velocity overshoot.

3.7.3 Collision between two main-sequence stars

In this test, we simulate a collision between two main-sequence stars. The aim is, on the one hand, to demonstrate the robustness and usefulness for the simulation of violent astrophysical events and, on the other hand, to measure how accurately matrix-inversion formulations numerically conserve physically conserved quantities.
for typical simulation parameters (such as the tree opening criterion and time integration prefactor.) For the simulation we choose two identical stars, each with a mass $M_\ast = 1 M_\odot$ and a radius $R_\ast = R_\odot$, modelled as a $\Gamma = 5/3$ polytrope with $2 \times 10^6$ equal mass SPH particles. The stars approach each other on a parabolic orbit with an impact strength

$$\beta = \frac{2 R_\ast}{R_{\text{pen}}} = 3.$$  

(57)

We perform this simulation twice, once with the evolution equation set that uses kernel gradients (stdGrad; first row in Fig. 29) and once with the equation set that uses matrix-inversion based gradients (MII; second row). During the first collision, the stars become heavily shocked (see panels 1 and 4), vorticity is created and the stellar cores are substantially spun up (panels 2 and 5). During the subsequent evolution the stars fall back towards each other, thereby creating further shocks and vorticity (panels 3 and 6). For this test, we find very similar results. We have performed these simulations with parameters that we would use for a practical simulation: $C = 0.2$ in equation (34) and a (rather tolerant) tree opening criterion $\Theta = 0.9$, see Gafton & Rosswog (2011) for a detailed description of the used RCB tree. The latter guarantees a fast evaluation of the gravitational forces, though at the price of sacrificing some accuracy. Despite this seemingly tolerant opening criterion, the conservation of both energy and angular momentum for both approaches are better than 0.4 per cent, see Fig. 30, and could be easily further improved by choosing a stricter force criterion.

3.7.4 Tidal disruption of a white dwarf star

As another astrophysical test case, we show a tidal disruption of a $0.5 M_\odot$ WD star by a $1000 M_\odot$ black hole (BH). Such encounters can lead to a tidal ignition and explosion of the WD (Luminet & Pichon 1989; Rosswog, Ramirez-Ruiz & Hix 2008; Rosswog et al. 2009) provided that the BH is of ‘intermediate’ mass (below $\approx 10^5 M_\odot$). We show a weak encounter with a ‘penetration factor’ $\beta = R_\ast/R_\beta = 0.9$, where $R_\ast = R_{\text{wd}}(M_{\text{wd}}/M_\ast)^{1/3}$ is the tidal radius inside of which a star is disrupted by the BH’s tidal forces. The quantity $R_\beta$ is the distance of closest approach (‘pericentre distance’). We chose a value of $\beta < 1$, since this results in a partial disruption where the star is ‘nearly disrupted’, but while receding from the BH its self-gravity overcomes the tidal pull again and a part of the tidal debris re-collapses into a self-gravitating core. Due to the highly elongated geometry and the small self-gravitating core such partial disruptions pose particular computational challenges.

We model the initial WD as a polytrope with exponent $\Gamma = 5/3$ with $5 \times 10^6$ equal-mass SPH particles. Since for the chosen parameters the pericentre distance is $R_\ast > 90 GM_{\text{wd}}/c^2$, we can treat the BH to excellent accuracy as a Newtonian point mass. Initially the WD is placed at a distance $r_0 = 6 R_\odot$ which guarantees that the tidal acceleration is only a tiny perturbation ($\ll 1$ per cent) compared to self-gravity. In Fig. 31 (left-hand panel), we show the density in the orbital plane at $t = 9$ code units (1 code unit $= 2.745$ s), when the star is approaching the BH, at $t = 20$, after the star has just passed it, and at $t = 47$ when the central region has re- contracted into a self-gravitating core.

3.7.5 Double TDE

As a last astrophysical test, we show the disruption of a stellar binary system of two main-sequance stars by a supermassive BH. Here, the main challenge comes from the widely varying geometry and the involved scales. Mandel & Levin (2015) studied tidal interactions of stellar binary systems with massive BHs and found that in a substantial fraction of cases both stars become disrupted. According to their estimate, close to 10 per cent of all stellar tidal disruptions may be double disruption events.

We simulate here the disruption of a binary consisting of two massive stars of 67.01 and 36.8 $M_\odot$ by an $M_{\text{bh}} = 10^6 M_\odot$ BH (initial conditions kindly provided by Ilya Mandel). Fig. 32 shows four snapshots of this disruption (colour coded is column density). The first shows the stage ($t = 1.21$ h) when the stars are approaching the BH and the leading, more massive star is about to be disrupted. The second snapshot ($t = 2.77$ h) shows the leading star being disrupted.

9For relativistic encounters one can use MAGMA2 with either the accurate pseudo-potential of Tejeda & Rosswog (2013), or, even better, within the approach suggested in Tejeda et al. (2017).
while the companion is approximately at pericentre. Snapshot three shows both disrupted stars receding from the BH, while in the last snapshot debris is fed in two narrow streams to the hole and an accretion disc is being assembled.

4 SUMMARY

In this paper, we have presented the new Lagrangian hydrodynamics code MAGMA2, which benefits from a number enhancements compared to traditional SPH codes.

(i) MAGMA2 uses consistently high-order kernels which substantially reduce noise, but this comes at the price of large neighbour numbers. Our default choice is a Wendland $C^6$ kernel together with 500 neighbours in the kernel support of each particle.

(ii) To produce entropy in shocks, our code employs artificial viscosity, but enhanced by techniques that are borrowed from finite-volume methods. Instead of employing the velocity difference between two particles in the artificial viscosity terms (which is the common SPH practice), we use the difference of the slope-limited, quadratically reconstructed velocities at the interparticle mid-point. All tests shown in this paper are performed with constant dissipation parameters ($\alpha = 1$ and $\beta = 2$) and even with such large parameters we find excellent results in benchmark tests. We have also implemented a new way to steer time-dependent dissipation by monitoring for each particle how well entropy is conserved. This allows to identify ‘troubled particles’ that need their dissipation increased. This approach is discussed in detail in a separate publication (Rosswog 2020).
(iii) Apart from a conventional SPH formulation (‘stdGrad’) that calculates derivatives via kernel gradients, MAGMA2 also offers two additional SPH formulations that use much more accurate gradient estimates that are based on matrix inversion techniques (‘MI1’ and ‘MI2’), see Section 2.1. These two formulations only differ in the way the SPH equations are symmetrized. All three SPH versions are implemented with the above-described kernels and artificial dissipation techniques.

(iv) Self-gravity and neighbour search are implemented via a fast tree that tessellates space by means of an RCB and that is described in detail in Gaftron & Rosswog (2011).

(v) In Section 3.1, we suggest a new way to set up SPH initial conditions. SPH is known to perform best when equal-mass particles are used, but setting up geometrically complicated initial conditions with equal-mass particles is non-trivial. To address this problem, we have introduced the artificial pressure method (APM), which is very much in the spirit of SPH. It starts from an initial distribution of equal mass SPH particles and compares the currently measured density with a desired density profile. Based on the local density errors it calculates an artificial pressure force that steers the SPH particles into positions where the deviations from the theoretical density profile are minimal.

We have scrutinized MAGMA2 in a large number of benchmark tests including smooth advection, a variety of shock and instability tests, vorticity creating Schulz–Rinne shocks (which are rarely shown in SPH publications) and a number of more astrophysical tests that demonstrate its robustness, versatility, and excellent conservation properties. We find very good results in these benchmarks, also in tests that are traditionally considered a challenge for SPH codes.

As expected, MAGMA2 is second-order accurate in smooth flows, see Section 3.2 and it yields good results in shocks. The techniques borrowed from finite-volume approaches (slope-limited reconstruction) in the artificial dissipation are a major improvement compared to the standard approach. For example, with reconstructed velocities, but large and constant artificial viscosity values MAGMA2 performs excellently in a KH test where SPH approaches without such a reconstruction fail completely. Even the low-resolution cases grow with rates very close to the (much higher resolved) reference solution. The effect of the quadratic reconstruction (as compared to a linear one) is small, though welcome. But it may be a valid choice to restrict oneself to just linear reconstruction and to avoid the need of calculating second derivatives.

We have also introduced a set of matrix inversion SPH equations (MI2) with a rarely used symmetrization in the particle indices. This symmetrization (though different gradients, dissipation strategy, kernels, etc.) has been successfully used the GASOLINE2 code (Wadsley et al. 2017). While delivering in most tests very similar results to the other matrix inversion formulation (MI1), MI2 has some distinct advantages:

(i) it is substantially more sensitive to density variations and – with everything else being the same – achieves substantially better results in the Sedov explosion test than the other two formulations.

(ii) in addition, it substantially reduces surface tension effects and therefore also has an advantage in instability tests compared to the other two SPH formulations.

The only slight disadvantage that we have noticed is that it is less robust against non-ideal particle setups. For example, in shock tests with particles placed on a CL it leads easier to particle ‘ringing effects’. However, such problems can be easily cured by a more sophisticated particle setup such as via the above-described APM.

We have further performed a number of the challenging, vorticity-creating shocks suggested by Schulz-Rinne (1993). Also here, MAGMA2 yields crisp results that are comparable with those from established Eulerian methods. We have also performed a number of more astrophysical simulations (stellar collision and tidal disruptions by BHs) to demonstrate the robustness of MAGMA2 and its accurate numerical conservation.

In none of the comparisons did we find any disadvantage of the matrix-inversion gradient prescription. But in many tests they showed clearly superior performance. Since the computationally expensive ingredients (long neighbour loops, matrix inversions, self-gravity) are shared by all three SPH formulations, we do not find substantial differences in their run times. As practically demonstrated in the stellar collision example, the matrix inversion formulations perform equally well in terms of numerical conservation as standard kernel gradients.

In its current version, MAGMA2 has implemented only self-gravitating gas dynamics with polytropic equations of state, but this framework will be enriched in the near future by more physics.

ACKNOWLEDGEMENTS

It is a great pleasure to thank Ilya Mandel for sharing the initial conditions of the double-TDE and Stephen Justham for providing stellar profiles of the corresponding masses. Thank you also to Davide Gizzi and Christoffer Lundman for their careful reading of an earlier draft. Some of the figures of this article were produced with the visualization software SPLASH (Price & Monaghan 2007). This work has been supported by the Swedish Research Council (VR) under grant number 2016-036573, by the Swedish National Space Agency under grant number Dnr. 107/16, by the research environment grant ‘Gravitational Radiation and Electromagnetic Astrophysical Transients (GREAT)’ funded by the Swedish Research Council (VR) under Dnr 2016-06012, and by the Knut and Alice Wallenberg Foundation under Dnr. KAW 2019.0112. We gratefully acknowledge support from COST Action CA16104 ‘Gravitational waves, black holes and fundamental physics’ (GWverse) and from COST Action CA16214 ‘The multi-messenger physics and astrophysics of neutron stars’ (PHAROS). The simulations for this paper were performed on the facilities of the North-German Supercomputing Alliance (HLRN) in both Göttingen and Berlin, and on the SNIC resources Tetralith and Beskow.

DATA AVAILABILITY

The data underlying this article will be shared on reasonable request to the corresponding author.

REFERENCES

Abel T., 2011, MNRAS, 413, 271
Ageritz O. et al., 2007, MNRAS, 380, 963
Ben Moussa B., Lanson N., Vila J., 1999, Int. Ser. Numer. Math., 129, 31
Benz W., Bowers R., Cameron A., Press W., 1990, ApJ, 348, 647
Brandenburg A., Dobler W., 2002, Comput. Phys. Commun., 147, 471
Cabezon R. M., Garcia-Senz D., Relano A., 2008, J. Comput. Phys., 227, 8523
Cabezon R. M., Garcia-Senz D., Escartín J. A., 2012, A&A, 545, A112
Cabezon R. M., Garcia-Senz D., Figueira J., 2017, A&A, 606, A78
Casanova J., Jose J., Garcia-Berro E., Shore S. N., Calder A. C., 2011, Nature, 478, 490
Cha S.-H., Whitworth A. P., 2003, MNRAS, 340, 73
Cha S.-H., Inutsuka S.-I., Nayakshin S., 2010, MNRAS, 403, 1165
