

Self-gravitating disks in binary systems: an SPH approach

I. Implementation of the code and reliability tests

L. D. Pinto^{1,2}, R. Capuzzo-Dolcetta¹, and G. Magni²

¹ Dipartimento di Fisica, Sapienza, Università di Roma, P.le Aldo Moro, 5, 00185 Rome, Italy
e-mail: luis.pinto@inaf.it

² INAF-IAPS, Istituto di Astrofisica e Planetologia Spaziali, Area di Ricerca di Tor Vergata, Via del Fosso del Cavaliere, 100, 00133 Rome, Italy

Received 2 April 2018 / Accepted 19 June 2019

ABSTRACT

The study of the stability of massive gaseous disks around a star in a nonisolated context is a difficult task and becomes even more complicated for disks that are hosted by binary systems. The role of self-gravity is thought to be significant when the ratio of the disk-to-star mass is non-negligible. To solve these problems, we implemented, tested, and applied our own smoothed particle hydrodynamics (SPH) algorithm. The code (named GaSPH) passed various quality tests and shows good performances, and it can therefore be reliably applied to the study of disks around stars when self-gravity needs to be accounted for. We here introduce and describe the algorithm, including some performance and stability tests. This paper is the first part of a series of studies in which self-gravitating disks in binary systems are let evolve in larger environments such as open clusters.

Key words. protoplanetary disks – hydrodynamics – binaries: general

1. Introduction

The study of protoplanetary disks in nonisolated system has become a relevant topic in numerical astrophysics because several planetary systems and disks around stars inside open clusters have recently been observed. The recent discovery of a Neptune-sized planet that is hosted by a binary star in the nearby Hyades cluster (Ciardi et al. 2018) has opened new perspectives for the study of the evolution of primordial disks that interact with binary stars in a nonisolated environment. In the context of the study of isolated binary systems, we note that most of the classical models have studied low-mass disks, with the result that even considering their self-gravity, no appreciable change is observed in the time evolution of the stellar orbital parameters. Additionally, low-mass disks give a poor feedback on the hosting stars, which means that the timescale for the variation in their orbital parameters is larger than the other dynamical timescales that are involved.

To investigate such systems, we built our own smoothed particle hydrodynamics (SPH) code to integrate the evolution of the composite star+gas system. Following the scheme described in Capuzzo-Dolcetta & Mocchi (1998), our code treats gravity by means of a classical tree-based scheme (see also Barnes 1986; Barnes & Hut 1986; Mocchi & Capuzzo-Dolcetta 2002) and adds a proper treatment of the close gravitational interactions of the gas particles. The evolution of a limited number of point-mass objects (which may represent either stars or planets) is treated with a high-order explicit method.

This preliminary work provides an instrument that is not only suited to modeling heavy protoplanetary disks that interact with single and binary stars. It can also be used to study these systems not in isolation, but in stellar systems such as open star clusters.

In the next section we describe our code after preliminarily introducing the numerical framework, and in Sect. 3 we present

and discuss some physical and performance tests. In Sect. 4 we describe the disk model we adopted and the application of our code to the study of heavy Keplerian disks. Section 5 is dedicated to the conclusions.

2. Numerical algorithm

In order to study a self-gravitating gas, we developed an SPH code, coupled with a tree-based scheme for the Newtonian force integration. In this section, after a brief recall of some basic theory of gravitational tree codes and of the SPH formalism (Sect. 2.1), we describe the main features of our numerical algorithm (Sect. 2.2). For further technical details, we refer to Appendices A.1 and A.2.

2.1. Basic theory

Introduced for the first time by Lucy (1977) and Gingold & Monaghan (1977), the SPH scheme has been widely adopted to investigate a huge set of astronomical problems that involve fluid systems. An SPH scheme allows integrating the fluid dynamical equations in a Lagrangian approach by representing the system through a set of points, or so-called pseudo-particles. For each particle, a set of fundamental quantities (such as density ρ , pressure P , internal energy u , and velocity \mathbf{v}) are calculated by means of an interpolation with a proper kernel function over a suitable neighbor. For an exhaustive explanation of the method, we refer to various papers in the literature, such as those by Monaghan & Lattanzio (1985), Monaghan (1988, 2005). Here we recall some basic aspects.

Interpolations are performed with a continuous kernel function $W(\mathbf{r}, h)$, whose spread scale is determined by a characteristic length h , called the smoothing length. It can easily be shown

(see, e.g., [Hernquist & Katz 1989](#)) that under some additional constraints, interpolation errors are limited to the order $O(h^2)$. We used as kernel function the cubic spline that was for the first time adopted by [Monaghan & Lattanzio \(1985\)](#), who developed a formalism that had been introduced by [Hockney & Eastwood \(1981\)](#). This kernel function has the following form:

$$W(r, h) = \frac{1}{\pi h^3} \cdot \begin{cases} 1 - \frac{3}{2} \left(\frac{r}{h}\right)^2 + \frac{3}{4} \left(\frac{r}{h}\right)^3, & 0 \leq r < h \\ \frac{1}{4} \left(2 - \frac{r}{h}\right)^3, & h \leq r < 2h \\ 0, & r \geq 2h \end{cases} \quad (1)$$

The SPH interpolation involves only a limited set of N' neighboring particles that are enclosed within the range $2h$, therefore the computational effort is expected to scale linearly with the total particle number N . On the other hand, when long-range interactions such as gravity are considered, the computational effort increases because each particle interacts with the whole system. A classical direct N-body code would therefore require a computational weight scaling as N^2 . However, a suitable gravitational tree-based scheme allows us to efficiently evaluate the Newtonian force by approximating the potential with a harmonic expansion (see [Barnes & Hut 1986](#); [Barnes 1986](#), for a full explanation). For each particle, only the contribution from a local neighborhood is calculated through a direct particle–particle coupling, while the contribution from farther particles is suitably approximated. The following expressions (Eqs. (2) and (3)) represent the approximated potential $\Phi(\mathbf{r})$ and the force (per unit mass) $\mathbf{a}(\mathbf{r}) = -\nabla\Phi(\mathbf{r})$ given by a far cluster of particles,

$$\Phi(\mathbf{r}) = -\frac{GM}{r} - \frac{1}{2} \frac{G \mathbf{r} \bar{\mathbf{Q}} \mathbf{r}}{r^5} \quad (2)$$

$$\mathbf{a}(\mathbf{r}) = -\frac{GM}{r^3} \mathbf{r} + \frac{G \bar{\mathbf{Q}} \cdot \mathbf{r}}{r^5} \mathbf{r} - \frac{5}{2} \frac{G \mathbf{r} \bar{\mathbf{Q}} \mathbf{r}}{r^7} \mathbf{r}. \quad (3)$$

M is the total mass of this ensemble, $r = |\mathbf{r}|$ is the distance of the particle under study to the center of mass of the cluster. The symbol $\bar{\mathbf{Q}}$ represents the so-called quadrupole tensor, which is associated with the specific cluster. In indexed form, it is given by

$$Q_{ij} = \sum_{k=1}^{N_C} \left(3x_i^{(k)} x_j^{(k)} - r_k^2 \delta_{ij} \right) m_k, \quad (4)$$

where $x_i^{(k)}$ and $x_j^{(k)}$ ($i, j = 1, 2, 3$) refer to the Cartesian coordinates of the k th particle of mass m_k . The summation is performed over all the N_C particles that are included in the cluster.

In the following section we describe the main structure and formalism used by GaSPH. Further computational details related to the implementation of the algorithm can be found in Appendices [A](#) and [B](#).

2.2. Main structure of the algorithm

A single step to compute the acceleration contains two preliminary phases. One cycle is dedicated to map the particles into an octal grid domain. A further cycle, linear in N , is needed to evaluate some key parameters such as the density, ρ , the smoothing length, h , and the pressure, P . Then a third set of operations, the most complicated operation, is that of evaluating the gravitational and hydrodynamical forces in addition to the internal energy rate \dot{u} of the gas.

GaSPH can also easily treat a system that consists of a set of point masses. To do this, the part of the SPH computations is turned off and the tree scheme alone is used for gravity interactions. On the other hand, a gas can be treated with pure hydrodynamics by turning off the gravitational field and using the SPH formalism alone.

After the main computations of the acceleration \mathbf{a} and the energy rate \dot{u} , the algorithm updates in time the velocity, position, and internal energy of the gas with a second-order Verlet method. Because of the structure of the second-order technique, the three main computational cycles should be performed twice into a single time iteration to obtain two estimates of \mathbf{a} and \dot{u} .

In addition, the smooth particles may interact with a small number N_{ob} of additional objects, an ensemble of point masses that mimic stars and/or planets. Differently from the other particles, the motion of these few objects is integrated with a 14th-order Runge–Kutta method by direct particle-to-particle N-body interactions without any approximation for the gravitational field. When N_{ob} is sufficiently small, these operations request a little additional computational effort that scales roughly linearly with respect to the total number of points (including both the SPH particles number N and the objects number N_{ob}). For the specific purpose of our investigation, where we have $N_{\text{ob}} \leq 2$, the general efficiency of the code is not affected.

2.2.1. Particle mapping and density computation

For a set of N equal-mass points, we preliminarily need to subdivide the system into a hierarchical series of subgroups of points in order to apply the multipole approximation for the Newtonian field contribution given by a “cluster”. To do this, we use a classical Barnes-Hut tree-code to map the particles into an octal grid space, according to their positions. We follow in particular the technique adopted by [Miochi & Capuzzo-Dolcetta \(2002\)](#) by mapping the points through a 3-bit-based codification (see Sect. [2.2.6](#) for further details).

Before the accelerations are computed, SPH particles need a preliminary stage in which densities and smoothing lengths are computed. To perform a good interpolation, we need to keep a fixed number of neighbors for each point. For inhomogeneous fluids, we must therefore use a smoothing length $h \equiv h(\mathbf{r}, t)$ that varies in space and in time.

Individual smoothing lengths should be chosen in such a way that the higher the local number density $n = \rho/m$, the smaller the interpolation kernel radius: $h \propto n^{-1/3}$, in order to have a roughly constant number of neighbors of the given particle. For this purpose, we adopt a commonly used prescription ([Hernquist & Katz 1989](#); [Monaghan 2005](#)). For each particle, we start from an initial guess for h , then we vary it until the number of particles that lie within the kernel domain reaches a fixed value N_0 . We iterate a process in which each time the number of neighbor points, N' , is counted using a certain smoothing length h_{prev} , then we update this length to a new value h_{new} according to the following formula:

$$h_{\text{new}} = h_{\text{prev}} \frac{1}{2} \left[1 + \left(\frac{N_0}{N'} \right)^{1/3} \right]. \quad (5)$$

If the fluid were homogeneous, $h_{\text{prev}} (N_0/N')^{1/3}$ would immediately provide the correct value of the smoothing length, without any further iteration. The addend 1 lets the program perform an average of the old smoothing length, for which any excessive oscillation error due to non-homogeneities in the spatial distribution of particles is damped. The iteration is stopped when convergence is reached according to the criterion $|N' - N_0| \leq \Delta N$, where

ΔN is a tolerance number. In this regard, [Attwood et al. \(2007\)](#) investigated the acoustic oscillations of some models of polytropic around the equilibrium by imposing a constant neighbor number N' and letting ΔN vary. They found that the fluctuation of $N' \pm \Delta N$ introduced an additional numerical noise that was able to break the stability of this system, giving rise to errors. To prevent errors, the authors found that ΔN should be set to zero, which is the choice we adopt in this paper. Moreover, they showed that the calculation of h according to the iterative process illustrated above and with $\Delta N = 0$ is equivalent to solving for all the particles the $2N$ -equations system described by the following two equations:

$$\begin{cases} \rho_i = \sum_{j=1}^{N'} m_j W(r_{ij}, h_i) \\ h_i = \delta \left(\frac{m_i}{\rho_i} \right)^{1/3} \end{cases}, \quad (6)$$

and to finding the exact solutions of density and smoothing length $\{\rho_i, h_i \mid i \in [1, N]\}$, with $\delta \approx 0.31 N_0^{1/3}$ a suitable constant, and r_{ij} the mutual distance between the i th and the j th particles. We typically use a number of neighbors $N' = 60$, such as $\delta \approx 1.2$.

When the density ρ_i is evaluated, the corresponding pressure P_i can be computed by means of a suitable equation of state. [Appendix B.1](#) illustrates further technical details about the neighbor-search procedure.

2.2.2. Force calculation and softened interactions

For a generic i th particle, the acceleration \mathbf{a}_i is computed by adding both the SPH terms and the Newtonian terms in the same iteration. Together with the acceleration, the internal energy rate of the particle is also computed.

To treat a self-gravitating gas with an SPH scheme, a proper treatment of the gravitational potential is necessary to avoid overestimating the gravity field. Particles can be considered as point sources of the Newtonian field when their mutual distance is larger than $2h$. Otherwise, their Newtonian interaction is, in consistency with the assumed kernel function ([Gingold & Monaghan 1977](#)), such that it vanishes at an interparticle distance approaching zero.

With the cubic spline kernel, we can obtain a different form of the Newtonian interaction between two particles, such that the classical term is softened if the particles approach within a distance of the order of a softening length $\epsilon = 2h$. See the appendix in [Hernquist & Katz \(1989\)](#) for more details, and the appendix in [Price & Monaghan \(2007\)](#) for an explicit expression of the force and the potential. When SPH interaction is turned off, a constant value ϵ is generally used in place of $2h$ for the softening length. In this case, the total energy is conserved within the numerical error. On the other hand, the Hamiltonian becomes time dependent with SPH systems because the softening length varies in time, and so the energy is no longer conserved. To solve this problem, the equations of motion must be rewritten in a conservative form that takes the variation in h into account. We followed the Hamiltonian formalism as adopted for the first time by [Springel & Hernquist \(2002\)](#) for the hydrodynamical interactions, which was further developed by [Price & Monaghan \(2007\)](#) for the gravitational field. The SPH equation assume the form

$$\frac{d\mathbf{v}_i}{dt} = - \sum_j \frac{1}{2} \left(g_{\text{soft}}(r_{ij}, h_i) + g_{\text{soft}}(r_{ij}, h_j) \right) \frac{\mathbf{r}_{ij}}{r_{ij}}$$

$$\begin{aligned} & - \sum_j m_j \frac{G}{2} \left(\frac{\zeta_i}{\Omega_i} \nabla_i W(r_{ij}, h_i) + \frac{\zeta_j}{\Omega_j} \nabla_j W(r_{ij}, h_j) \right) \\ & - \sum_j m_j \left(\frac{P_i}{\rho_i^2 \Omega_i} \nabla_i W(r_{ij}, h_i) + \frac{P_j}{\rho_j^2 \Omega_j} \nabla_j W(r_{ij}, h_j) \right) \\ & - \sum_j m_j \Pi_{ij} \left[\frac{\nabla_i W(r_{ij}, h_i) + \nabla_j W(r_{ij}, h_j)}{2} \right] + \frac{d\mathbf{v}_i^{\text{[stars]}}}{dt} \end{aligned} \quad (7)$$

$$\frac{d\mathbf{u}_i}{dt} = \sum_j m_j \left(\frac{P_i}{\rho_i^2 \Omega_i} + \frac{1}{2} \Pi_{ij} \right) \mathbf{v}_{ij} \cdot \nabla_i W(h_i), \quad (8)$$

where the index i refers to a generic i th particle and the index j in the sums refers to the j th particle that is enclosed within the range $2h_M = 2 \cdot \max(h_i, h_j)$. The term g_{soft} represents the softened gravitational force per unit mass mentioned above: it is a function only of the mutual particle distance r_{ij} and of the smoothing length h . It tends to zero as $r_{ij} \rightarrow 0$ and assumes the classical Newtonian form $m_j G r_{ij}^{-2}$ for $r_{ij} \geq 2h$. The operator ∇_i represents the gradient with respect to the coordinates of the i th particle. The gradient is performed over two different expressions of the kernel W , with two different lengths h_i and h_j . The terms ζ_i and Ω_i are suitable functions that account for the variation in the smoothed Newtonian potential with respect to the softening length and for the non-uniformity of the softening length itself, respectively. They assume this form for a generic particle of index i :

$$\Omega_i = 1 + \frac{h_i}{3\rho_i} \sum_j m_j \nabla_i W(r_{ij}, h_i) \quad (9)$$

$$\zeta_i = - \frac{h_i}{3\rho_i} \sum_j m_j \frac{\partial \phi_{\text{soft}}(r_{ij}, h_i)}{\partial h_i}, \quad (10)$$

where in the same way for the system in [Eq. \(6\)](#), the sum extends over the particles that are enclosed within the range $2h_i$. In [Eq. \(10\)](#), the function ϕ_{soft} represents the softened gravitational potential, such that $\nabla \phi_{\text{soft}} = -g_{\text{soft}} \mathbf{r}_{ij}/r_{ij}$. The potential reaches a constant value as $r_{ij} \rightarrow 0$ and becomes equal to the Newtonian potential for $r_{ij} \geq 2h$ (for an explicit expression, see, e.g., [Price & Monaghan 2007](#)). Terms Ω and ζ are computed in the same neighbor-search iterative loop where ρ and h are worked out.

Only if the gas interacts with stars does the last term in [Eq. \(7\)](#), $d\mathbf{v}_i^{\text{[stars]}}/dt$ (discussed in [Sect. 2.2.4](#)), represent a non-null acceleration that accounts for the Newtonian interaction between i th particle and the point masses. The function Π_{ij} , which we discuss in the following section, characterizes the well-known artificial viscosity. The expression of the equation of motion, [Eq. \(7\)](#), guarantees a symmetric exchange of linear momentum between the particles.

2.2.3. Artificial viscosity

In high-compression regions such as shock wavefronts, the velocity gradient may be so strong that two layers interpenetrate and the hydrodynamical equations may not be integrated correctly and generate unphysical effects. Additional artificial pressure terms are a possible solution for this problem. In our code, we added an artificial term by adopting the same classical schematization as [Monaghan \(1989\)](#), which corresponds to

introducing a suitable artificial viscosity that dampens the velocity gradient when two particles approach. Practically, a viscous-pressure term Π_{ij} is included in Eqs. (7) and (8). It assumes the expression

$$\Pi_{ij} = \begin{cases} \frac{-\alpha\bar{c}\mu_{ij} + \beta\mu_{ij}^2}{\bar{\rho}_{ij}}, & \text{if } \mathbf{v}_{ij} \cdot \mathbf{r}_{ij} < 0, \\ 0, & \text{if } \mathbf{v}_{ij} \cdot \mathbf{r}_{ij} > 0, \end{cases} \quad (11)$$

where $\mu_{ij} = \frac{h\mathbf{v}_{ij} \cdot \mathbf{r}_{ij}}{r_{ij}^2 + \eta^2 h^2}$. The dot product $\mathbf{v}_{ij} \cdot \mathbf{r}_{ij}$ involves the relative velocity and the distance of a pair of particles $i - j$. Only the particles that move in, for which $\mathbf{v}_{ij} \cdot \mathbf{r}_{ij} < 0$, contribute to the artificial viscosity. The parameter η is a suitable term to prevent singularities when two particles approach very closely (we use the typical value of $\eta = 0.1$). The terms \bar{h} , $\bar{\rho}$, and \bar{c} represent the average values of the smoothing length $\frac{1}{2}(h_i + h_j)$, the density $\frac{1}{2}(\rho_i + \rho_j)$, and the speed of sound $\frac{1}{2}(c_{si} + c_{sj})$, respectively. We set $\beta = 2\alpha$. In this simple formulation, the artificial viscosity is activated throughout the fluid; but in two circumstances it should be damped to prevent unphysical effects. Artificial viscosity must be damped in regions where shear dominates, and where the velocity gradient is low.

For two shearing layers of fluid, the relative velocity between the particles leads to an approach that is interpreted by the artificial viscosity (11) as a compression. This incorrect interpretation causes the code to overestimate the strength of the viscous interaction. To prevent false compressions, [Balsara \(1995\)](#) multiplied the term μ_{ij} by a proper switching coefficient:

$$f = \frac{|\nabla \cdot \mathbf{v}|}{|\nabla \cdot \mathbf{v}| + |\nabla \times \mathbf{v}| + 10^{-4}c_s h^{-1}}, \quad (12)$$

in which the divergence of velocity and the velocity curl are evaluated for a particle of index i as

$$\begin{cases} (\nabla \cdot \mathbf{v})_i = \rho_i^{-1} \sum_j m_j \mathbf{v}_{ij} \cdot \nabla_i W(r, h_i) \\ (\nabla \times \mathbf{v})_i = \rho_i^{-1} \sum_j m_j \mathbf{v}_{ij} \times \nabla_i W(r, h_i) \end{cases} \quad (13)$$

We implemented the term f by multiplying μ_{ij} for an average value $\bar{f} = \frac{1}{2}(f_i + f_j)$. Further problems may arise far away from high-compression regions. In the classical formulation of Π_{ij} , $\alpha = 1 = \text{const.}$ (e.g., in [Monaghan 1992](#)). In this scheme, the viscosity acts with the same effectiveness in every region, while we would expect the artificial term to be efficient only where it is needed, that is, close to the shock fronts. To solve this problem, we used the same formalism as was introduced by [Morris & Monaghan \(1997\)](#) and further developed by [Rosswog et al. \(2000\)](#) by considering an individual α_i for each particle that follows the time-variation equation,

$$\frac{d\alpha_i}{dt} = -\frac{(\alpha_i - \alpha_{\min})}{\tau_\alpha} + S_i, \quad (14)$$

where $S_i = \max(-(\nabla \cdot \mathbf{v})_i, 0)$ ($\alpha_{\max} - \alpha_{\min}$) represents a ‘‘source’’ term that increases in the proximity of the shock front; α_{\min} represents a minimum threshold value for α , and α_{\max} represents its maximum. The (increasing) rate of the viscosity coefficient is driven by a characteristic timescale $\tau_\alpha = h_i/bc_s$ that depends on how the fluid allows the perturbations to propagate through the resolution length. The individual viscosity coefficients α_i and α_j , when referred to a generic $i - j$ particle pairing, are averaged in the same way as was done with the other quantities.

For a gas with $\gamma = 5/3$, a good value for the b coefficient can be set such that $5 \leq b^{-1} \leq 10$ ([Morris & Monaghan 1997](#)). For our tests, we set $\alpha_{\max} = 2$, $\alpha_{\min} = 0.1$ and $b^{-1} = 5$. These are the most commonly adopted values in literature for a wide class of problems involving collapse, merging stars or protoplanetary disks (see, e.g., [Rosswog & Price 2007](#); [Stamatellos et al. 2011](#); [Hosono et al. 2016](#)). The implementation of the artificial viscosity term (Eqs. (12)–(14)), together with its form implemented in Eqs. (7) and (8), may affect the accuracy of the code in preserving the total angular momentum. In Sect. 3.1.5 we discuss how this form of viscosity, with different choices of the coefficients α_{\min} and b , guarantees the conservation of the angular momentum.

2.2.4. Additional stellar objects

We calculated a direct point-to-point interaction for the mutual interaction between stars and to couple stars with SPH particles. The equation of motion of a generic p -th star takes the following form:

$$\begin{aligned} \frac{d\mathbf{v}_p}{dt} = & - \sum_j \frac{1}{2} (g_{\text{soft}}(r_{pj}, \epsilon_p) + g_{\text{soft}}(r_{pj}, h_j)) \frac{\mathbf{r}_{pj}}{r_{pj}} \\ & - \sum_s \frac{1}{2} (g_{\text{soft}}(r_{ps}, \epsilon_p) + g_{\text{soft}}(r_{ps}, \epsilon_s)) \frac{\mathbf{r}_{ps}}{r_{ps}}, \end{aligned} \quad (15)$$

where $g_{\text{soft}}(r, \epsilon)$ represents the Newtonian acceleration, which takes the form discussed above in Sect. 2.2.2. The force softening is accounted for the stars as well according to a constant softening length $\epsilon_s = \text{const.}$ The gravity is thus softened when the mutual distance approaches ϵ_s . The first summation is extended over all SPH particles, while the index s in the second sum refers to the generic stars.

Similarly, the equation of motion, Eq. (7), when referred to a gas i th particle contains the following sum:

$$\frac{d\mathbf{v}_i^{\text{[stars]}}}{dt} = - \sum_s \frac{1}{2} (g_{\text{soft}}(r_{is}, \epsilon_s) + g_{\text{soft}}(r_{is}, h_i)) \frac{\mathbf{r}_{is}}{r_{is}}, \quad (16)$$

where the index s again refers to the stars, and \mathbf{r}_{is} is the distance vector between a gas particle and a star.

2.2.5. Time integration and time-stepping

To evolve the gas system in time, we adopted a second-order integration method that is similar to a classical second-order Runge–Kutta scheme but is at the same time very similar to a leap-frog integrator: the well-known velocity-Verlet method (see [Andersen 1983](#); [Allen & Tildesley 1989](#), for detailed references). The Verlet method is based on a trapezoidal scheme coupled with a predictor-corrector technique for the estimation of \mathbf{v} and \mathbf{u} . The structure of this scheme is very similar to that of classical symplectic leap-frog algorithms, although it requires two computations of the force at every time iteration (see Appendix A.1). Nevertheless, the general velocity-Verlet method applied to gas evolution shows some advantages compared to the symplectic algorithm of the same order. Like a standard Runge–Kutta method, velocity and positions are updated in synchronized steps, without the $\Delta t/2$ shift. This feature provides a good flexibility in problems that are approached with nonuniform time-steps that involve the interaction of the gas component with other components that are integrated with different methods, as in our case. Various applications of velocity-Verlet methods in SPH schemes are found in the literature, for example, in [Hubber et al. \(2013\)](#) or in [Hosono et al. \(2016\)](#).

The additional point masses are ballistic elements, whose equation of motion needs to be integrated with very high precision to avoid secular trends that are typical of few-body gravitational problems. Although the SPH precision is at only second order, we decided to integrate the Newtonian motion of the (few) stars and planets in the system with a 14th-order Runge–Kutta method that was recently developed by Feagin (2012) through the so-called m -symmetry formalism. The method consists of 35 force computations per time-step, and in analogy with the well-known second- and fourth-order RK methods, it updates the velocities and the positions by suitable linear combinations of 35 different Kr and Kv coefficients (see Appendix A.2 for further details).

For the gas, we chose the time-step Δt following a criterion similar to the standard Courant–Friedrichs–Lewy (CFL) criterion that is commonly adopted for SPH systems (see, e.g., Monaghan 1992), together with some additional criteria. A global time-step Δt_{\min} can be determined by taking the minimum between the following two quantities:

$$\Delta t_{\text{term}} = \min_i \left(\frac{C h}{c_{si} + h_i |\nabla \cdot \mathbf{v}_i + \varphi \alpha_i [c_{si} + 2 \max_j(\mu_{ij})]} , C_u \frac{u_i}{\dot{u}_i} \right) \quad (17)$$

$$\Delta t_{\text{dyn}} = \min_i \left(C_a \sqrt{\frac{h_i}{a_i}} , C_d \frac{v_i}{a_i} \right), \quad (18)$$

where c_s is the sound speed, C is a coefficient whose typical value lies between 0.1 and 0.4; we usually choose 0.15. Moreover, C_u , C_a , and C_d are coefficients to be set < 1 . We chose $C_u = 0.04$, $C_a = 0.15$, and $C_d = 0.02$. Finally, the coefficient φ typically ranges from 0.6 to 1.2 (we adopt $\varphi = 1.2$ throughout). Similarly to the control of the variation in kinetic energy, we control the time variation of the thermal energy u/\dot{u} in a single time-step by limiting it to a certain fraction $C_u = 0.04$. The index i refers to an individual time-step Δt_i that is related to a specific particle.

For the point-particle phase in the system (i.e., stars or planets), we chose a characteristic time-step, Δt , that is defined as

$$\Delta t_{\text{stars}} = \min_s \left(C_{\text{ob}} \sqrt{\frac{\epsilon_s}{a}}, C_d \frac{v_s}{a_s} \right), \quad (19)$$

where we use $C_{\text{ob}} = 0.15$. The various quantities with the index s of course characterize a specific star particle.

For a homogeneous medium, the integration can be performed with a global time-step, that is, the lowest value of gas and stars. Generally, the particles have different resolutions h_i and different accelerations, which leads to a wide class of typical evolution timescales. For some particles, the integration might therefore be made with different Δt_i , which avoids the explicit force calculation at every time iteration and saves some computing time. We adopted a technique that was implemented in several N-body algorithms, such as the classical TREESPH (Hernquist & Katz 1989) or in the multi-GPU-parallelized N-body code HiGPUs (Capuzzo-Dolcetta et al. 2013). We assigned to each point a time-step as a negative two-power fraction of a reference time $\Delta t_{\text{max}} = \max_i(\Delta t_i)$ (it can be a fixed quantity, or it may change periodically during the simulation). The particle motion is updated periodically according to their Δt_i in such a way that after an integration time Δt_{max} , all of them are synchronized (further details are explained in Appendix A.1). Particle mapping and sorting are

performed every time for every particle as well, independently of their individual time-step. Thus, the configuration of the tree grid, together with total mass and quadrupole momentum of the boxes, are computed every single step Δt_{\min} . Similarly, at each minimum time-step iteration, the gravitational interactions between gas and star are computed even during “inactive” stages of the gas. Thus, the acceleration of the inactive SPH particles is split into two terms: one is given by a fixed non-updated hydrodynamical and self-gravity term, and the other is given by a constantly updated gas-star gravitational force.

In our scheme, the stars and planets do not follow an individual time-step scheme, and their mutual interactions are computed for every single step Δt_{\min} , even when $\Delta t_{\text{stars}} \neq \Delta t_{\min}$. Furthermore, we force the particles that lie close to the stars within a tolerance distance to be integrated at every time iteration. Practically, we compute the distance for a generic i th particle from the stars, and we furthermore predict this distance at the following time iteration. When these values are lower than a tolerance of $\kappa \epsilon_{\text{ob}}$ (with a constant $\kappa \geq 2$), the particle time-step drops to Δt_{\min} . For our practical purposes, we used a small number of objects (in the current investigation, $N_{\text{ob}} \leq 2$), thus, the 35-stage RK scheme requires a relatively short CPU-time (less than 2% of the total).

In gas problems that involve strong shocks, the use of individual time-steps may lead to strong errors. Even though CFL conditions are satisfied, the strong velocity gradients may determine a great discrepancy in time-steps between close particles. Consequently, close particles may evolve with very strongly different timescales. This may create too many asymmetries in the mutual hydrodynamical interactions, which causes unphysical discontinuities in velocity and pressure. Following the idea of Saitoh & Makino (2009), we limit for each pair of neighboring i th and j th particles the ratio of time-steps $\frac{\Delta t_i}{\Delta t_j} \leq A$. These investigators have shown that a good compromise is reached by the choice of $A = 4$, which gives good results without abruptly affecting the efficiency of the code.

2.2.6. Approximation of the gravitational field: opening criterion

The decomposition of the system into a series of clusters is performed by the tree algorithm through a recursive octal cube eight sub-boxes, each one subdivided into a further eight cubes of order $L = 2$, and so on. A tree structure is thus constituted, made of several nested boxes, each of which contains a group of particles. To calculate the acceleration of an i th particle, the algorithm walks along the tree, starting from the low-order cubes toward the highest order cubes (that contain just one particle), and evaluates the distance between the particle and the center of mass of the boxes. Each time a box is probed, the code decides to open it and probes its internal cubes only if the well-known opening criterion is satisfied,

$$\frac{D_L}{r} > \theta, \quad (20)$$

where θ is the so-called opening angle parameter for which reasonable values range among 0.3 and 1 (see Sect. 3.2, dedicated to performance tests), and $D_L = D_0 \times 2^{-L}$ is the side length of the box. In the opposite case, the algorithm decides to approximate the gravitational field by adding for the acceleration of the i th particle just the contribution of the box (given by Eq. (3)). With this scheme, the net amount of computation scales down to $N \log N$, which is far shorter than N^2 for large N .

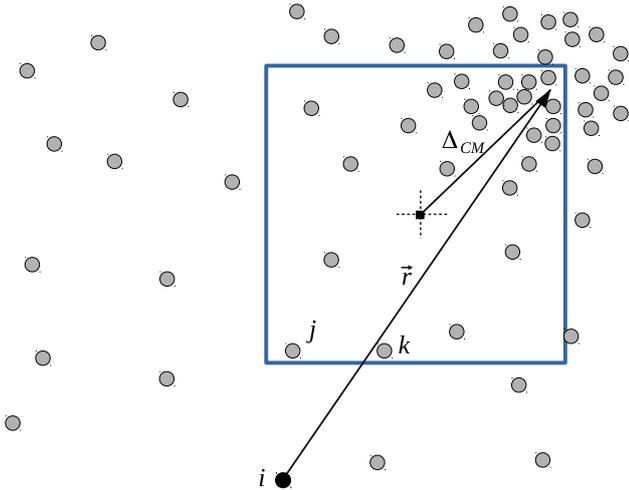


Fig. 1. Schematic 2D example of the lack of accuracy in the field computation caused by a large offset Δ_{CM} . The center of gravity is far away from the i th particle, and the cube is not opened. Nevertheless, some particles that lie near the edge of the box, such as the j th and the k th points, are very close to the i th point, but their direct contribution is missed, which results in a loss of accuracy.

With \mathbf{r}_a the geometrical center of a certain cube and \mathbf{r}_{CM} its center of mass position, we may find a very large offset $\Delta_{CM} \equiv |\mathbf{r}_a - \mathbf{r}_{CM}|$ under specific circumstances. With a center of mass far away from the box center, some errors may arise in the force approximation because a cube can be considered “far enough” from a particle according to the opening criterion even though some of the points enclosed in the box may still be very close to the particle (see Fig. 1). These close particles are therefore ignored, and the whole box gives the multipolar approximated contribution to the particle acceleration. The acceleration is therefore calculated with less accuracy than might be expected. A first key to avoid these errors should be adopted by checking whether the particle lies very close to a box (as was done, e.g., by Springel 2005). If the test particle is inside a cube or close to its borders according to a certain tolerance, the box is always opened, independently of the truthfulness of Eq. (20).

We can furthermore optionally modify the opening criterion in our code by taking the offset term into account. We may use the following rule to open a box:

$$r < \frac{D_L}{\theta} + \Delta_{CM}. \quad (21)$$

This prescription is equivalent to the classic opening criterion, but with an effective opening angle $\theta' < \theta$, to guarantee that every close box is opened. In some peculiar cases in which Δ_{CM} is large (i.e., comparable with the length of the semidiagonal of the box), as in the example of Fig. 1, the effective opening angle is considerably smaller than θ . In Sect. 3.2 we show that for a typical value of $\theta = 0.6$, the adoption of the new criterion does not require too much additional computational effort, especially when many particles are involved.

3. Code testing

We illustrate here some basic physics tests (Sect. 3.1) and a series of performance tests (Sect. 3.2). In Sect. 3.1 we apply GaSPH to two basic problems: (i) a non-hydrodynamical system, characterized by a cluster of point-mass particles distributed according to a Plummer profile, and (ii) a classical shock-wave

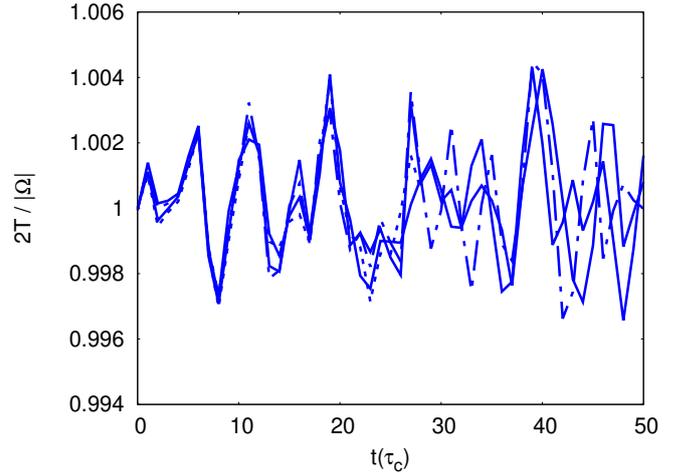


Fig. 2. Oscillation of the virial ratio as a function of time for different choices of code configuration parameters for the simulation of a Plummer distribution of 10^5 equal-mass particles. The continuous line shows $\theta = 0.6$, $\epsilon = 0.2$; the dashed line represents $\theta = 1.0$, $\epsilon = 0.2$; the dotted line shows $\theta = 0.6$, $\epsilon = 0.5$; and the dash-dotted line denotes $\theta = 1.0$, $\epsilon = 0.5$.

problem. These quality tests are followed by some applications to hydrodynamical systems at equilibrium. First, we treat some polytropes with finite radius. Then, we compare our algorithm with a well-known hydrodynamical tree-based code (Gadget-2) in the case of a gaseous Plummer sphere. In Sect. 3.2 we analyze the computational efficiency and the accuracy of our code in different contexts.

3.1. Tests with gas and pressureless systems

3.1.1. Turning off the SPH: the evolution of a pressureless system

In order to test the stability of our numerical method, we performed a series of simulations in which we placed a set of points according to the standard Plummer configuration (Plummer 1911), which is often adopted to study the distribution of stars in globular clusters. The Plummer sphere is pressureless, so that the particles interact only through gravity, and there is no SPH interaction. As units of measurement, we chose the total mass M and the gravitational constant G , and we placed an ensemble of $N = 10^5$ particles in a Plummer distribution with core radius $R = 1$ and cutoff radius $R_{out} = 10R$. The particles had equal masses $m = N^{-1}$ and equal softening length ϵ , chosen as a fraction of the central mean interparticle distance: $\epsilon = \alpha_s \left(\frac{m}{\rho_0}\right)^{1/3} = \alpha_s \left(\frac{4\pi}{3N}\right)^{1/3}$ (with $\alpha_s \in [0.2, 1.0]$). Starting the Plummer distribution at the virial equilibrium, we integrated its time evolution for 50 mean crossing-times τ_c . This parameter is defined as the initial ratio between the half-mass radius and the mean dispersion velocity $\frac{R_{1/2}}{\sqrt{\langle v^2 \rangle}}$. Figure 2 shows the virial ratio $\frac{2T}{|\Omega|}$ as a function of time and compares four runs made with different combinations of the opening angle θ (0.6; 1.0) and ϵ (0.2; 0.5). The four results illustrated in Fig. 2 do not show any relevant difference: the virial ratios oscillate within a small fraction $< 0.5\%$, especially for the configuration with $\theta = 1$ and $\alpha_s = 0.5$, which was expected to be the worst case.

We note that we do not work with a classical high-precision N-body code such as Nbody-6 (for example see Aarseth 1999). The Newtonian force is approximated by means of both the

multipolar expansion, which occurs when particles are sufficiently far, and the softening length damping, which occurs when the particles approach within a distance of about ϵ . Despite these approximations, acceptable results can be obtained in a noncollisional system like our Plummer distribution. The results lie within reasonable errors.

3.1.2. Sedov–Taylor blast wave

To test the code with strong shock waves, we simulated the effects of a point explosion on a homogeneous infinite hydrodynamical medium with constant density ρ_0 and null pressure. If an amount of energy E_0 is injected at a certain point r_0 , an explosion occurs and then a radial symmetric shock wave propagates outward. Sedov (1959) investigated this problem and found a simple analytical law for the time evolution of the shock front:

$$r_s(t) = \left(\frac{E_0}{\rho_0 a} \right)^{1/5} t^{2/5}, \quad (22)$$

where r_s is the radial position of the front relative to the point of the explosion r_0 , while a is a function of the adiabatic constant γ (it is close to 0.5 for $\gamma = 5/3$, and it approaches 1 for $\gamma = 7/5$). Furthermore, the fluid density immediately behind the shock front ($r \leq r_s$) has the following radial profile:

$$\rho(r, t) = \frac{\gamma + 1}{\gamma - 1} \rho_0 G_\gamma \left(\frac{r}{r_s} \right), \quad (23)$$

where G_γ is an analytical function of the relative radial coordinate r/r_s . Similarly as in many previous works (see, e.g., Rosswog & Price 2007; Tasker et al. 2008), we set the initial conditions for a homogeneous and static medium ($\rho_0 = 1$, $v = 0$) by placing 10^6 equal-mass particles in a cubic lattice structure, confined in a box with x , y , and z coordinates each ranging from -1 to 1 . γ was set to $5/3$, and the explosion was simulated by giving an amount of energy $E_0 = 1$ to the origin of the system. We were unable to reproduce a point explosion with an SPH system because its spatial resolution is determined by the kernel support. We therefore needed to inject the energy in a small region with the same scale as $2h$. We thus gave at a time t_* the energy E_0 to the particles that were enclosed in a sphere with radius $R = 2h$. Figure 3 shows three different radial average density profiles $\rho(r, t')$ that correspond to the times $t = 0.05$, $t = 0.1$, and $t = 0.2$. The results are compared with the analytical solution. Although the position of the front follows the expected law of Eq. (22), the peak does not reach the expected value $\frac{\gamma+1}{\gamma-1}\rho_0 = 4\rho_0$.

Intrinsic errors in approximating the physical quantities, given by the smoothing kernel, allow the density to spread out and follow a wider distribution than the true profile. This corresponds to a smoothing of the vertical discontinuity and so to a lower peak of the density. The same figure shows a comparison with results obtained from a further test that was made with the same system, but using a better resolution ($N = 3\,375\,000$). The peak of the curve clearly reaches a higher value.

3.1.3. Polytopes at equilibrium

We tested our code in the case of hydrodynamic self-gravitational systems by building static polytopes with different indexes ($n = 1$, $n = 3/2$, and $n = 2$). A generic polytope of index n constitutes a radially symmetric system whose equation

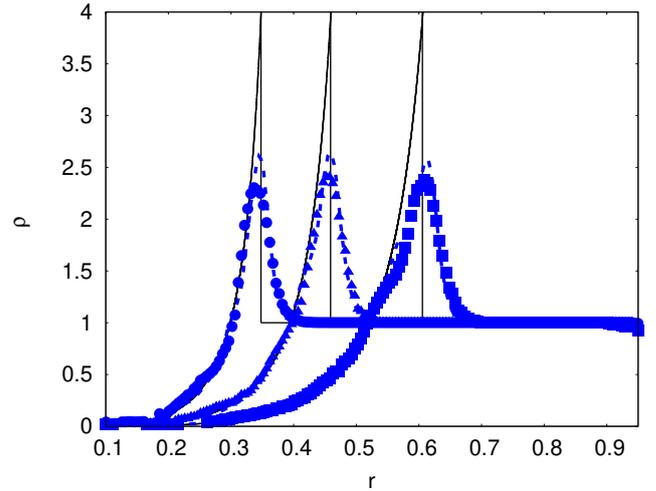


Fig. 3. Radial density profiles of the Sedov–Taylor blast wave at several times (increasing rightward) $t = 0.05$ (circles), $t = 0.1$ (triangles), and $t = 0.2$ (squares) in a simulation with $N = 10^6$. The results obtained using a higher resolution ($N = 3\,375\,000$) are plotted with dotted lines. The full lines represent the classical Sedov–Taylor auto-similar solutions.

of state follows the expression

$$P(r) = K_n \rho(r)^{1+\frac{1}{n}}, \quad (24)$$

where the density is parametrized as $\rho(r)/\rho_0 = \theta^n(r)$, and ρ_0 is the central value. The static radial solution $\theta(r)$ can be found by writing an equilibrium condition between the hydrostatic pressure gradient and the gravitational forces, from which the well-known Lane–Emden equation can be obtained (an exhaustive treatment can be found, e.g., in Chandrasekhar 1958):

$$\frac{\alpha^2}{r^2} \frac{d}{dr} \left(r^2 \frac{d\theta}{dr} \right) = -\theta^n, \quad (25)$$

with $\alpha^2 = (n+1) K_n \rho_0^{\frac{1}{n}-1} / 4\pi G$, and K_n a suitable normalization coefficient. For an index $n \in (0, 5)$, the system has a finite radius and the coefficient K_n depends, through α^2 , both on the radius R and on the total mass M . We set both of them to 1 in our tests, implying $K_1 \approx 0.637$, $K_{3/2} \approx 0.424$ and $K_2 \approx 0.365$.

We tested the ability of our code to let a system spontaneously relax in a polytrope configuration, following the prescription adopted in Price & Monaghan (2007). Starting from a homogeneous sphere of particles placed in a lattice structure, the system was let evolve by forcing the pressure to follow Eq. (24). We forced the SPH system to evolve by damping the velocities with an additional acceleration $\mathbf{a}_{\text{damp}} = -0.05 \mathbf{v}$, until the kinetic energy decreased to a small fraction (1%) of the total energy. A standard nonconstant α was chosen for the artificial SPH viscosity, with $\alpha_0 = 0.1$, and a number of neighbors of 110 was set for the particles.

A correct treatment of self-gravity and hydrodynamic interactions among SPH particles, and the choice of the equation of state (24) allows the system to acquire the density profile $\rho_0 \theta^n$, which is the solution of Eq. (25). Figure 4 shows the three radial density profiles we obtained for the different polytropic indexes. The resolution, related to the particles number, affects the accuracy of the code in correctly sampling the profile $\rho(r)$, especially in the central denser regions. Mainly for the higher index $n = 2$, a higher particle number is needed to let the numerical density

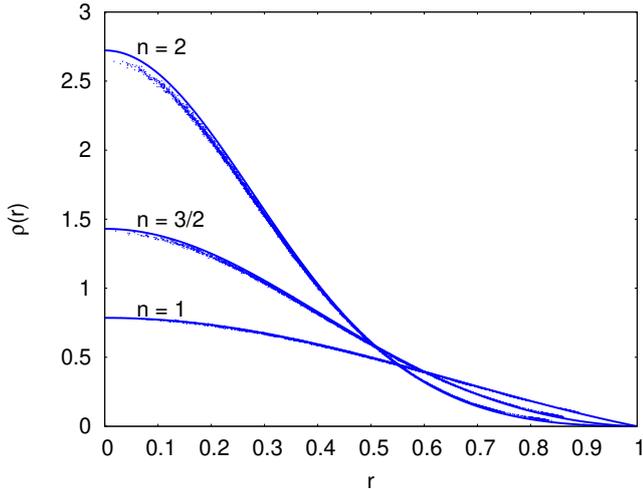


Fig. 4. Equilibrium analytical solutions of the density profiles $\rho(r)$ related to three different models of polytropes with indexes $n = 1$, $n = 3/2$, and $n = 2$, drawn as solid curves. Comparison with the computed profiles (dots).

approach the theoretical expected value at a specific accuracy level. 10 000 particles were used for the models with $n = 1$ and $n = 3/2$, while the polytrope with index $n = 2$ was built with 20 000 particles.

3.1.4. Gaseous Plummer distribution

We tested the equilibrium of a static gas density distribution according to the Plummer function:

$$\rho(r) = \rho_0 \left[1 + \left(\frac{r}{a} \right)^2 \right]^{-5/2}, \quad (26)$$

with the central density defined by $\rho_0 = 3M/4\pi a^3$, where M and a are the total mass of the system and a characteristic length, respectively. We set $M = 1$ and $a = 1$ ($G = 1$ in internal code units) so that the half-mass radius of the system is $r(50\%) \approx 1.3$. In a static configuration with a null velocity field, the gas SPH particles compensate for the mutual self-gravity with a pressure gradient that results from a temperature distribution $T(r) = \kappa \rho(r)^{1/5}$. κ represents a constant that is calibrated by taking the equation of state of a perfect gas $P = (\gamma - 1)\rho u$ into account and by imposing the virial equilibrium between gravitational energy W and total thermal energy $U = \sum_i u_i m_i$, that is, $|W| = 2U$. We placed 50 000 particles according to a Monte Carlo sampling of the distribution in Eq. (26). A realistic distribution has an infinite radius, therefore we used a cutoff at a proper radial distance $r \approx 22$, such that the distribution contained 99.8 % of the mass of a realistic infinitely extended Plummer sphere. Figure 5 shows the time variation of some Lagrangian radii that contain 5%, 15%, 30%, 50%, 75%, and 90% of the total system mass for an integration time of 90 central free-fall timescales ($\tau_0 = (3\pi / 32G\rho_0)^{1/2} \approx 1$ in our code units).

We compared the results with the well-known gravitational SPH code Gadget-2 (Springel 2005). Figures 6 and 7 show a comparison between the radial density profiles obtained by the two algorithms with the same choice of the main parameters. The α viscosity coefficient was set constant and equal to 1. The reported density was computed at $t = 90 \tau_0$, even though the system reached an acceptable equilibrium state already within a few units of τ_0 after several slight oscillations.

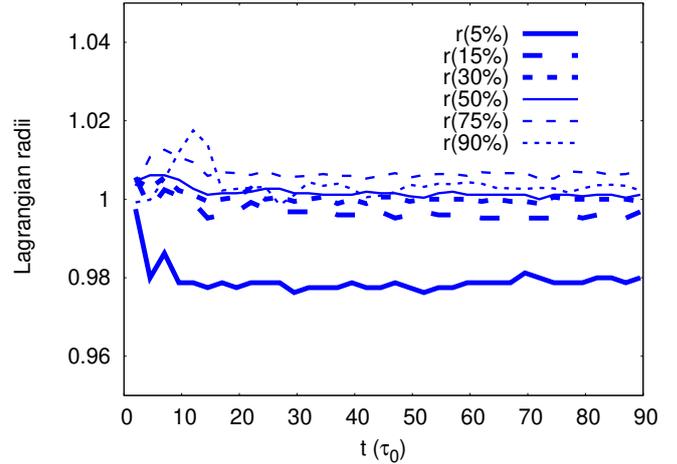


Fig. 5. Lagrangian radii as a function of time for a hydro-Plummer distribution at equilibrium (see Sect. 3.1.4). The radii are normalized to their respective initial values.

In Fig. 7 we can distinguish three main radial zones for $r \leq 0.3$, $0.3 < r < 2$, and $r \geq 2$. In the middle zone, the codes are in good agreement and provide a density profile with an accuracy lower than 2% with respect to the analytical model. For $r < 0.3$, Gadget-2 describes a density that deviates by up to the 6% from the expected value, while our program has a maximum deviation of 11%. For both models, these higher errors can be ascribed to the fact that the system contains only about 2% of the total mass (and thus 2% of the total particles) within the radial distance $r = 0.3$, which causes a poor sampling of the potential inside the sphere. Consequently, the system tends to shrink slightly. In the outward zone, the deviations can reach significantly higher values with both codes because the density values are far lower than in the central zone. We therefore conclude that in the context of a standard physical environment, the two codes show a satisfactory agreement overall.

3.1.5. Artificial viscosity and angular momentum conservation

A nonconstant artificial viscosity may lead to nonconservation of the angular momentum, L . The actual conservation of this quantity was tested with different settings of the artificial viscosity parameters in Eq. (14), integrating the time evolution of a system that was very similar to the one described in the previous section. We used the same Plummer distribution (see Eq. (26)), with $M = 1$ and $a = 1$, made of 50 000 SPH particles. The same thermal energy profile was adopted, but scaled down by a factor 1/2, so that $T(r) = \frac{\kappa}{2} \rho(r)^{1/5}$. We converted the (subtracted) thermal energy into kinetic energy by assigning to each i th particle a clockwise azimuthal velocity, with absolute value $v_i = \sqrt{u_i}$ (where u_i was the original specific thermal energy characterizing the Plummer system used in Sect. 3.1.4), and a direction parallel to the X, Y plane. The system thus acquired a nonzero vertical component of the angular momentum, $L_z = \sum_i m_i (x_i v_{yi} - y_i v_{xi})$.

The virial equilibrium was still formally preserved because gravitational potential energy and thermokinetic energy were the same such that $|W| = 2(K + U)$, but the (new) angular rotation triggered changes in the density distribution. We integrated in time for about 100 initial central free-fall timescales τ_0 (which is on the same order as the azimuthal dynamical timescale, taken as the ratio $r_c/v(r_c) \approx 1.4$, where r_c is the initial radius at which

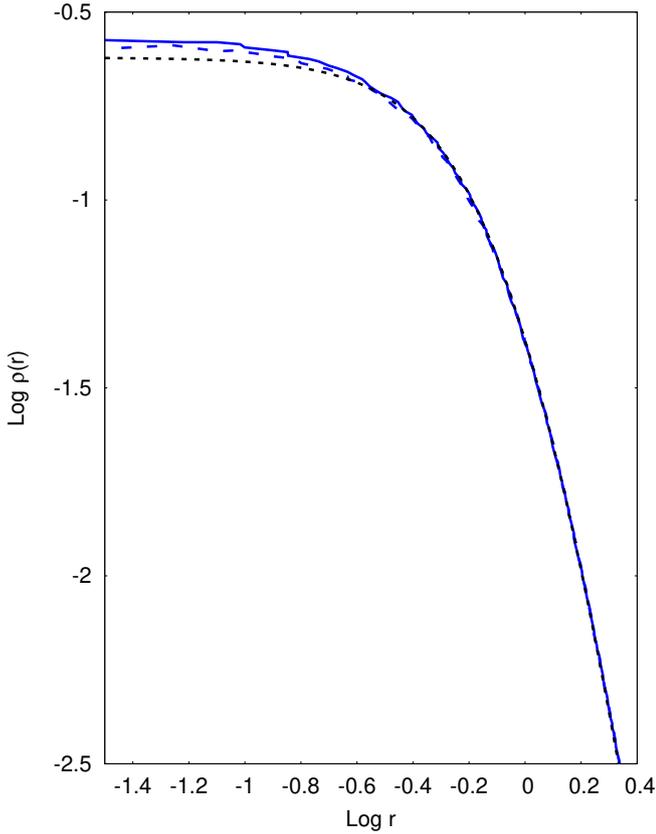


Fig. 6. Radial density profile for a Plummer distribution with 50 000 SPH particles (solid line). The results obtained with *Gadget-2* are also shown (dashed line). The analytical Plummer profile is plotted with a dotted line. The density is in units such that $\rho_0 = 3/(4\pi)$.

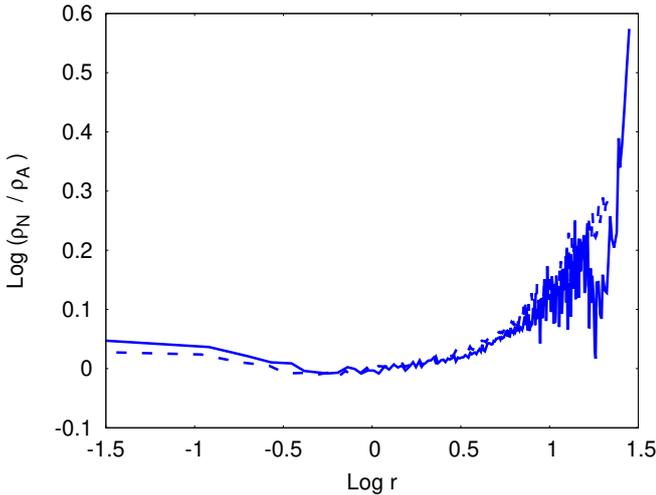


Fig. 7. Logarithmic ratio ρ_N/ρ_A of the numerical to analytical radial density profile for a Plummer distribution with 50 000 SPH particles (solid line). The results obtained with *Gadget-2* are also shown (dashed line).

the density drops by a factor 1/2). We performed three different simulations by varying in the α rate Eq. (14) the parameters α_{\min} and b . We set $\alpha_{\min} = 0.1$, $b^{-1} = 5$, $\alpha_{\min} = 0.02$, $b^{-1} = 5$, and $\alpha_{\min} = 0.1$, $b^{-1} = 7$. The angular rotation changes the configuration of the system, which causes the initial Plummer density distribution to become flatter perpendicularly to the z -axis,

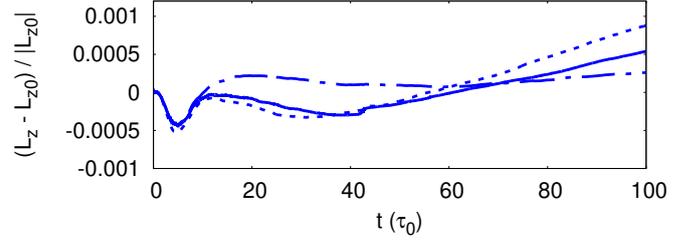


Fig. 8. Fractional variation in the z -component of the angular momentum for our simulated rotating Plummer model (see Sect. 3.1.5). We plot the results obtained by varying some configuration parameters: $\alpha_{\min} = 0.1$, $b^{-1} = 5$ (full line), $\alpha_{\min} = 0.02$, $b^{-1} = 5$ (dotted line), and $\alpha_{\min} = 0.1$, $b^{-1} = 7$ (dash-dotted line).

while the whole system expands. During an initial phase of about $20\tau_0$, the distribution underwent some rapid variation followed by a slow secular evolution. Figure 8 shows the quantity $(L_z - L_{z0})/|L_{z0}|$, which represents the variation as a function of time of the component L_z compared to its initial value L_{z0} . The three lines refer to the different choices of the parameters α_{\min} and b . The curves show a conservation of the angular momentum within 10^{-3} up to 100 evolution timescales. In particular, the choice of $\alpha_{\min} = 0.1$, $b^{-1} = 7$, compared with the other configurations, gives a stronger variation of L_z during the first phases, while it shows a lower change rate during the secular evolution of the system. On the other hand, a low value $\alpha_{\min} = 0.02$ gives rise to a better conservation in the initial phases and a higher deviation during later stages. We observed in all three simulations that the two components L_x and L_y maintain values compared to L_z within a relative error of 10^{-3} .

3.2. Code performance

In order to analyze the computational efficiency of our algorithm as a function of the particle number, we performed several tests by measuring the average CPU time that was spent for a single run by the main routines. We therefore studied the performances of the *GaSPH* code in three different contexts:

- (1) A system with pure self-gravity and zero pressure, adding a comparison with the results of *Gadget-2*.
- (2) A system with self-gravity and SPH pressure.
- (3) A system similar to that of case 2, but with the addition of 20 point star-like external objects.

We performed the tests by placing a set of N particles with the same Plummer density profile distribution as adopted in Sect. 3.1.1. The program was tested on an Intel®Core™ i7-4710HQ architecture with 6MB of cache memory and with 16GB of RAM memory DDR3L with a data-transferring speed of 1600 MHz.

For a standard tree code without SPH, the computational time per particle is expected to be linear in $\log N$ because the overall time scales as $N \log N$. To increase the efficiency and save considerable memory resources, we can also use a simple formalism made by considering only the first “monopole” term $-MGr^{-3}\mathbf{r}$ that appears in the right member of Eq. (3). This is a technique that was also adopted in *Gadget-2* and simplifies the complexity of the algorithm by neglecting the efforts for the quadrupole tensor computation. The suppression of the quadrupole term decreases the computational time, with a minor cost in terms of accuracy. For a pure self-gravitating system, Fig. 9 shows the CPU time needed for a single particle force calculation as a function of the ten-based logarithm of the particle

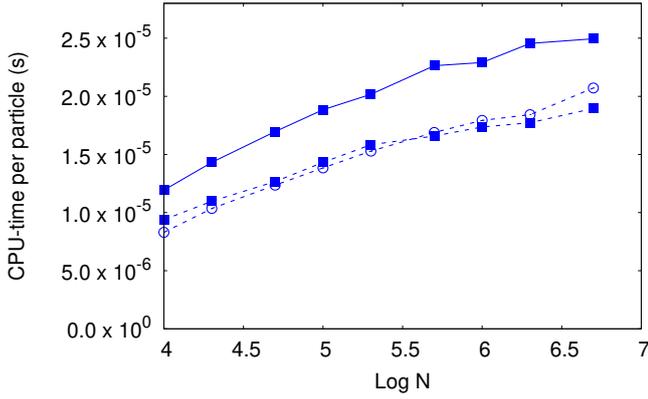


Fig. 9. CPU time per particle for the pure gravitational force calculation in monopole and quadrupole approximations at different N . Gadget-2 results (empty circles) are compared to our code results (squares connected with dashed lines) in the same monopole approximation. The continuous line refers to the performance of GaSPH, and the quadrupole term is included in the field.

number N (ranging from 10^4 to 5×10^6). Choosing an opening angle $\theta = 0.6$, we performed a series of force evaluation by considering a simple pressureless system, with particles interacting only with the Newtonian field. The computational times measured using our code (averaged over a reasonable number ≥ 30 of equal tests) are comparable with the average CPU times measured using Gadget-2. The figure also shows the results based on a second series of runs with GaSPH performed with the quadrupole term included in the gravitational field. Including this term, an additional CPU time of about 30% is requested. Figure 10 compares the previous CPU times with the times needed by GaSPH for a full self-gravitating SPH system, with the quadrupole term included in the computation and the same value of $\theta = 0.6$. The times per particle for the density computation routine are also shown. In computing the acceleration, the additional time per particle is fairly independent of N , as can be seen in the figure, because the close SPH interactions are always made over a fixed number of neighboring points, which we set to 60 in this example. For the same reasons, the average time per particle needed to calculate the density is also expected to be constant, as Fig. 10 shows. The calculus of ρ and h requests an iterative process in which the routine for each particle is called several times. The CPU times illustrated in the figure are the average values per single iteration. Determining the optimum value of h the code typically requires no more than two iterations.

The optional introduction of the offset Δ_{CM} term in the opening criterion (as discussed in Sect. 2.2.6) causes in some cases a considerable reduction of the effective angle θ . Consequently, the number of direct particle-to-particle interactions increases, which decreases the code performance. Figure 11 illustrates the code efficiency in terms of number of particles processed in a second. The results, related to the two acceleration routines (pure self-gravity and self-gravity with SPH) shown in the previous graph, are compared with other result obtained by including the offset term Δ_{CM} in the opening criterion of Eq. (21). A substantial but not drastic worsening in performance can be observed. For instance, using 5×10^6 SPH particles and including Δ_{CM} , the code computes the accelerations at a rate of $\approx 24\,000$ particles per second (about 17% slower than the case without Δ_{CM}). Computations were made with $\theta = 0.6$ and including the quadrupole terms. The performance of the same two force subroutines (pure gravity and gravity plus hydrodynamics) were also studied at

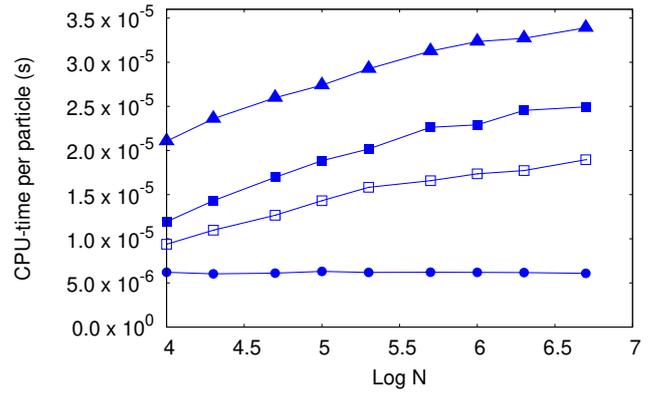


Fig. 10. GaSPH average CPU times per particle as a function of N . Comparison of different routines: SPH neighbor-search routine averaged for a single iteration (circles), pure gravity computation with monopole term (empty squares) and with quadrupole term (filled squares), self-gravity computation up to quadrupole and including the SPH terms (triangles). The units are the same as in Fig. 9.

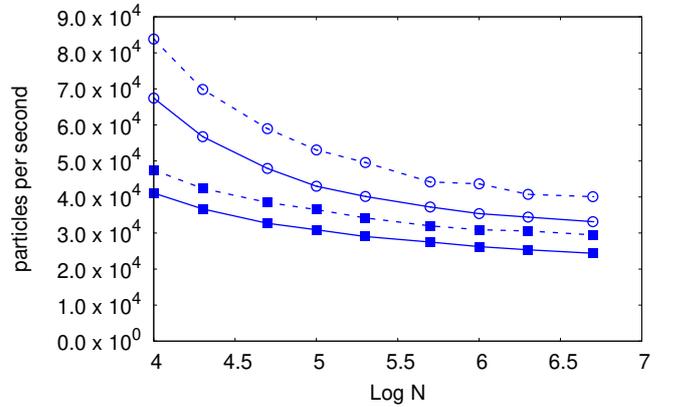


Fig. 11. Number of processed particles per second. The continuous lines and the dashed lines refer to results with and without the correction term Δ_{CM} , respectively. We show the simple gravity field calculation (empty circles) and the full self-gravity routine with hydrodynamics (squares). The quadrupole term is considered for the gravity field. $\theta = 0.6$.

different values of θ (CPU times per particle as a function of N are shown in Fig. 12).

Smaller angles should provide higher precision at the cost of a longer computational time. On the other hand, with larger angles there are fewer direct point-to-point interactions and we gain in efficiency, but we expect a lower accuracy. We evaluated the accuracy of our tree code by measuring a “mean relative error” in computing the accelerations according to the prescription suggested by Hernquist (1987), with different conditions of particle number N and opening angle θ . The prescription consists of a comparison of the three components of the acceleration vector as computed by means of the tree scheme, a_k^{TREE} , $k = 1, 2, 3$, with the “exact” value a_k^{NBODY} computed by direct summation. A mean error $\langle \delta a_k \rangle = \frac{1}{N} \sum_i (a_k^{(i)\text{TREE}} - a_k^{(i)\text{NBODY}})$ is computed by averaging over all the N particles. Then, the relative error is computed as follows:

$$\text{Err}(a_k) = \frac{\sum_{i=1}^N |a_k^{(i)\text{TREE}} - a_k^{(i)\text{NBODY}} - \langle \delta a_k \rangle|}{\sum_i |a_k^{(i)\text{NBODY}}|}. \quad (27)$$

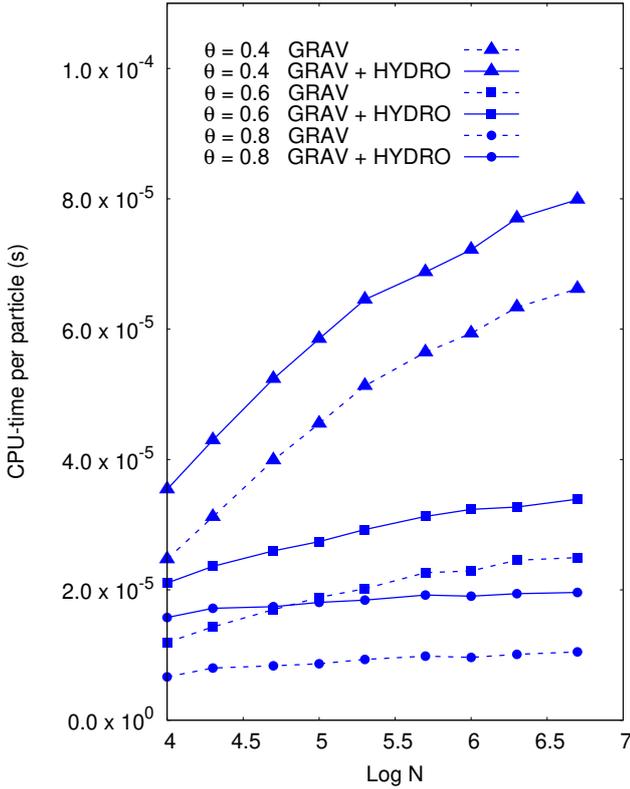


Fig. 12. Computational time per particle vs. $\log N$ for various values of the opening angle θ . The pure tree gravitational algorithm (dashed line) and full SPH+gravity algorithm are shown as dashed and solid lines, respectively. The quadrupole term is included in the force evaluation. The Δr_{CM} offset term is not considered by the opening criterion.

Figure 13 shows these relative errors, obtained with GaSPH, as a function of the CPU time. The figure does not show the relative errors for each single component, but shows the mean values,

computed by the simple average $\frac{1}{3} \sum_{k=1}^3 \text{Err}(a_k)$. Results for several setup configurations are illustrated. The figure shows the results in three different panels, according to the value of N ($N = 10^4$, $N = 10^5$, and $N = 10^6$). For each value of N , we used different combinations of the parameter θ (0.4, 0.6, 0.8) with different opening criteria (Eq. (20) or Eq. (21)) and different multipole approximations (only the monopole term, or also including the quadrupole term). As the data show, the approximation of the field with the quadrupole moment always represents the best choice in terms of performance because at the same error, it requests a shorter CPU time than the monopole approximation. On the other hand, the choice of the new opening criterion gives a smaller improvement of the error with respect to the benefits that are obtained by switching from monopole to quadrupole term.

For lower particle numbers, a better computational performance without loss of accuracy can be obtained by including the quadrupole approximation and the (more expensive) opening criterion given in Eq. (21), together with a suitable change of the θ angle. We focus, for example, on the simulation setups characterized by $\theta = 0.6$ with any possible opening criterion, monopole approximation, and $N = 10^4$ or $N = 10^5$. The change $\theta = 0.6 \rightarrow \theta = 0.8$, together with the use of criterion (21), represents a good choice and provides better performing simulations without degrading the precision of the algorithm. We obtain the same advantage when we pass, similarly, from $\theta = 0.4$ to $\theta = 0.6$.

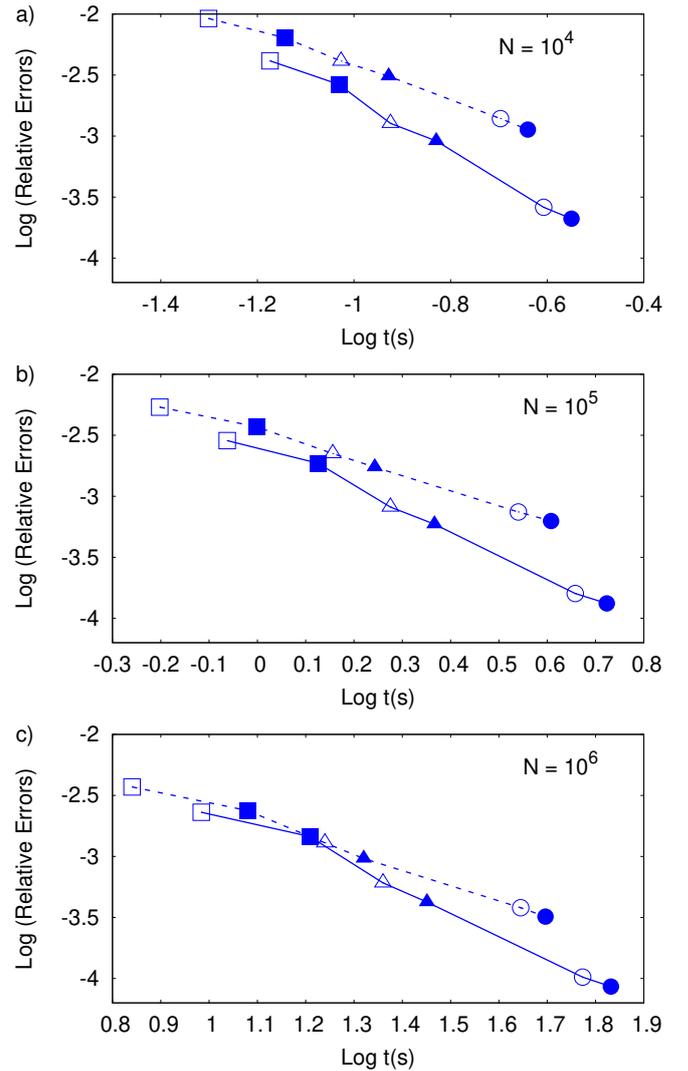


Fig. 13. Tree code relative errors $\text{Err}(a_k)$ (averaged over all the three Cartesian coordinates) for the gravitational field computation as a function of the CPU time. The data are illustrated for different particle numbers ($N = 10^4$, $N = 10^5$, and $N = 10^6$) in panels *a*, *b*, and *c*. In each panel, the full line connects the points related to a computation of the gravitational field made with the quadrupole approximation, while the dashed line refers to computations made by using just the monopole term. The shape of the “void” markers distinguishes different choices of θ with the “standard” criterion (Eq. (20)): $\theta = 0.8$ (void squares), $\theta = 0.6$ (void triangles), and $\theta = 0.4$ (void circles). Results for different opening angles with the “opening law” (Eq. (21)) criterion are also marked: $\theta = 0.8$ (solid squares), $\theta = 0.6$ (solid triangles), and $\theta = 0.4$ (solid circles).

On the other hand, for $N = 10^6$ (panel *c* in Fig. 13), the results related to the approximation with monopole and with quadrupole have smaller differences than the other cases with different N . The choice of a larger opening angle (passing from $\theta = 0.6$ to $\theta = 0.8$ or passing from $\theta = 0.4$ to $\theta = 0.6$) together with the use of the new opening criterion can give a better performance even though the accuracy slightly decreases.

If we wish to preserve the high efficiency of the tree code by keeping the CPU time to scale as $N \log N$, an angle $\theta \geq 0.3$ must be chosen (Hernquist 1987). The choice of $\theta = 0.6$ in quadrupole approximation or the choice of $\theta = 0.4$ in monopole approximation together with criterion (21) represents a satisfying option because it provides relative errors of at most about 10^{-3} .

Table 1. Work profiling (in percentage with respect to the total) of GaSPH, tested on a Plummer gas distribution with 20 stars and different numbers of SPH (first column).

N	Tree build.	Neigh. search	GRAV + HYDRO acc.	Stars & gas inter.	Other oper.
10^4	8.3	19.7	66.9	2.5	2.5
2×10^4	6.0	18.2	71.1	2.4	2.3
5×10^4	6.3	16.9	71.9	2.2	2.8
10^5	6.0	16.7	72.7	2.0	2.5
2×10^5	6.0	15.7	74.2	2.2	2.0
5×10^5	5.9	15.0	75.2	1.9	2.0
10^6	6.0	14.5	75.9	2.0	1.7
2×10^6	6.2	14.3	75.8	1.8	1.9
5×10^6	6.3	13.7	76.2	1.7	2.1

Notes. The opening angle is $\theta = 0.6$, and we chose $N' = 60$ neighbor particles.

We now added $N_{\text{ob}} = 20$ stars to the SPH distribution. As explained in the previous section, a 14th-order explicit method was applied to calculate the evolution of these objects, and giving their low number in comparison to N , very little additional CPU time is expected. Table 1 reports the percentage of workload related to the main relevant subroutines for the new gas+stars system: tree-building + particle-sorting routine, density computation routine, acceleration routine, and the star evolution routine. In addition, we report the rest of the time needed for the basic operations (such as \mathbf{v} , \mathbf{u} , and \mathbf{r} updating, energy computation, and time-step computation). Different work balances are shown for several values of N . The percentage of workload related to the tree-building routine is stable to the order of 6% at different N . In contrast, the work needed by the density routine becomes less and less relevant as N increases, while the gravity+SPH computation becomes increasingly essential. The computational effort to treat the evolution of stars, together with their interaction with the gas, is due both to the pure N -body RK coupling, which is expected to scale as N_{ob}^2 , and to the time for coupling each star with each SPH particle, which is expected to be linear in N . Nevertheless, Table 1 shows that 20 stars contribute very little to the total CPU time. For the specific purposes of our current work, we used fewer than two stars or two stars, and their contribution to the code effort is therefore far lower than the 2% ÷ 3%.

4. Protoplanetary disks

This section is dedicated to the description of some tests performed on two main problems involving protoplanetary disks. The first test is to compare the numerical integration of a disk around one star with the analytical prediction (Sect. 4.1). The second test involves a self-gravitating disk that interacts with a binary star (Sect. 4.2).

4.1. Protoplanetary disks around one star

4.1.1. Disk model

Here we illustrate the general setup we used to model a protoplanetary disk in equilibrium around a star of mass $M_s = 1 M_\odot$.

According to the classical flared-disk model (see, e.g., Garcia 2011; Armitage 2011), we let the disk revolve around the central object with a roughly Keplerian frequency $\Omega_k \approx \sqrt{M_s G/R^3}$ (where R is the cylindrical coordinate $R = \sqrt{x^2 + y^2}$ in the reference frame centered on the central object). The disk evolution is essentially driven by secular viscous dissipation. According to the well-known α -disk model (Shakura & Sunyaev 1973), the turbulence in the internal disk is schematized by means of a pseudo-viscosity of the following form:

$$\nu = \alpha_{SS} c_s H. \quad (28)$$

This kinematic viscosity perturbs the fluid equations by leading to a net transport of matter inward and an outward flux of angular momentum. α_{SS} represents a characteristic efficiency coefficient for the momentum transport, while $H = c_s/\Omega_k$ represents a characteristic vertical pressure of the disk scale height. The viscous evolution is usually much slower than the dynamical evolution (the characteristic secular timescale is $\propto r^2/\nu$, which typically is 2 or 3 orders of magnitude larger than Ω_k^{-1}). This modeling of the turbulence is basically dimensional and is made by mainly taking dynamical turbulence processes into account. Thus, α_{SS} extends over a wide range of variability (typically 10^{-4} and 10^{-2}). When the disk self-gravity is strong enough, another important effect arises that is due to the gravitational perturbations. Several works (see, e.g., Mayer et al. 2002; Boss 1998, 2003) have numerically estimated the gravitational timescales in a protoplanetary disk, which is on the same order as its dynamical time. They showed that under certain conditions, matter can undergo instabilities and eventually condense to form clumps in $10^3 \div 10^4$ yr, which may give rise to gaseous planets. It can be shown that the disks maintain their equilibrium state against collapse according to the Toomre criterion,

$$Q = \frac{c_s \Omega_e}{\pi G \Sigma} > 1.5, \quad (29)$$

where Ω_e represents the epicyclic frequency, which is approximately equivalent to Ω_k for Keplerian disks (see Binney & Tremaine 1987; Toomre 1964, for a detailed study). The Toomre factor is a general coefficient that quantifies the predominance of the gravitational processes over the typical thermal and dynamical actions.

We initially let our disk revolve with an azimuthal velocity $v_k \equiv v_\phi(R) = \sqrt{\frac{G(M_s + M(R))}{R}}$ that depends both on the mass of the central star M_s and on the internal mass of the disk itself $M(R) = \int_0^R \Sigma(R) 2\pi R dR$. The cumulative mass $M(R)$ can be neglected only for low disk masses $M_D \ll M_s$. The shape of the disk in the direction perpendicular to the revolving midplane depends on the vertical pressure scale height H , such that pressure and density scale with a Gaussian profile $\exp(-z^2/2H^2)$. Here a local vertically isothermal approximation was used because we assumed that any radiative input energy from the star is efficiently dissipated away: the cooling times are far shorter than the dynamical timescales. The disk is thus vertically isothermal, and the temperature depends only on the radial distance from the central star.

We set the thermal disk profile according to the well-known flared-disk model, for which the ratio H/R increases with R (see Garcia 2011; Dullemond et al. 2007, for a full clarification). The disk temperature thus follows the profile

$$T = T_0 \left(\frac{R}{R_0} \right)^{-q}, \quad (30)$$

which is commonly used by setting $q = 1/2$, while R_0 represents a scale length. We used a slightly different slope $q = 3/7$, adopted by [D'Alessio et al. \(1999\)](#) by assuming that the thermal processes in the inner layers of the disk do not affect its dynamical stability.

With this temperature profile (independent of t and z), the gas pressure follows a barotropic equation of state $P = c_s^2 \rho$. This choice represents a rough approximation of the cooling processes and allows us to model self-gravitating disks in equilibrium only when $Q > 2$, which excludes disk models with a state of marginal stability ($Q \approx 1$). In a realistic model of a disk without the isothermal approximation, when the Toomre parameter approaches unity, the loss of thermal energy due to radiative cooling processes leads to matter aggregation, which in turn causes shock waves that heat the gas again. If the disk is capable of retaining a sufficient amount of the additional thermal energy that is generated, the collapse only causes some spiral instabilities that do not increase exponentially. The collapse process is thus arrested and the disk reaches a meta-stable state. Every time that a gravitational instability occurs, it is further dissipated by the heat back-production in this meta-state. For a good treatment, see for example [Kratter & Lodato \(2016\)](#). Conversely, the isothermal equation adopted by our model forces the system to cool down at an infinitely high efficiency rate, and to expel all the additional thermal energy that is generated by the compression of matter. Thus, in regions where $1 \leq Q < 2$, the density increases and the collapse process is not halted by production of heat. Our model of a disk in equilibrium is thus limited to masses M_D for which the self-gravity guarantees the condition $Q \geq 2$.

We used $\bar{\mu} = 2.33$ as the mean molecular weight for the gas. This parameter is commonly adopted to model protoplanetary disks (see, e.g., [Kratter et al. 2008](#); [Liu et al. 2017](#)) because it is an average mean weight for a gas composed of H_2 and He, based on the observed cosmic abundance of the elements.

The effects of the Shakura-Sunyaev viscosity associated with Keplerian disks can be emulated by means of the SPH artificial viscosity. [Meglicki et al. \(1993\)](#) found that the SPH viscosity coefficient α provides a viscous acceleration with an effective kinematic viscosity that contains a similar form with a shear component plus a bulk viscosity. When a cubic spline function is used for the kernel function, the authors showed that ν assumes the following form:

$$\nu \propto \alpha c_s h. \quad (31)$$

In several works (e.g., [Artymowicz & Lubow 1994](#); [Nelson et al. 1998](#)) the viscosity term in Eq. (11) is used under peculiar conditions: it acts not only for approaching particles, but also for points that move out (with $\mathbf{r}_{ij} \cdot \mathbf{v}_{ij} > 0$). It turns out that $\nu = 0.1 \alpha c_s h$ (for a comprehensive explanation, see [Meru & Bate 2012](#)). We thus have the following law that connects the Shakura-Sunyaev viscosity coefficient to the α parameter used in SPH:

$$\alpha_{SS} = \frac{1}{10} \alpha \frac{h}{H}. \quad (32)$$

This modification of the SPH formalism provides a more realistic prediction of the effect given by a kinematic viscosity because it acts under compression and under gas expansion. This prescription is reliable except for strong velocity gradients, that is, shock waves due to strong compressions. In this case, the classical [Morris & Monaghan \(1997\)](#) amplification law (Eq. (14) in this paper) would generate high dissipative forces even in expansion regions. Protoplanetary disks are usually modeled as quiet

systems and are not expected to undergo such huge compressions to let strong shock waves arise.

Using Eq. (11) for the viscosity, and consequently, activating dissipation only for particles approaching each other, the law in Eq. (32) can be modified and improved by also considering the effects of the β coefficient on the kinematic viscosity ([Meru & Bate 2012](#); [Picogna & Marzari 2013](#)). It follows that for Keplerian disks,

$$\alpha_{SS} = \frac{31}{525} \alpha \frac{h}{H} + \frac{9}{70\pi} \beta \frac{h^2}{H^2}. \quad (33)$$

Equations (32) and (33) formally do not contain any effect of the Balsara switch to compensate for the false sharing attenuation (Eq. (12)). As was done in [Picogna & Marzari \(2013\)](#), we used the artificial viscosity term of the disk in Eq. (11), and multiplied the factor μ_{ij} by the term of [Balsara \(1995\)](#) \tilde{f} .

4.1.2. Evolution of the viscous disk

We tested the response of our model with respect to dissipative processes with long timescales as are characteristic of viscous turbulent disks. We started from the well-known disk model developed by [Lynden-Bell & Pringle \(1974\)](#) (see also [Pringle 1981](#); [Hartmann et al. 1998](#)). It consists of a thin ($H/R \ll 1$) non-self-gravitating disk that is subjected to a power-law dissipative turbulent viscosity. The surface density evolution is described by the following equation (see [Pringle 1981](#); [Hartmann et al. 1998](#)):

$$\frac{\partial \Sigma(R, t)}{\partial t} = \frac{3}{R} \frac{\partial}{\partial R} \left[R^{1/2} \frac{\partial}{\partial R} \left(R^{1/2} \nu \Sigma(R, t) \right) \right]. \quad (34)$$

It has been shown that when the disk is perturbed by a radial power-scaling kinematic viscosity $\nu \propto R^\phi$, the differential equation admits the following similarity solution (see [Lynden-Bell & Pringle 1974](#); [Hartmann et al. 1998](#)):

$$\Sigma(R, t) = \Sigma_0 \left(\frac{R}{R_1} \right)^{-\phi} \left(\frac{t}{\tau_\nu} + 1 \right)^{-\gamma} \exp \left[- \left(\frac{t}{\tau_\nu} + 1 \right)^{-1} \left(\frac{R}{R_1} \right)^{2-\phi} \right], \quad (35)$$

where $\gamma = \frac{5/2-\phi}{2-\phi}$. R_1 is a characteristic radial scale that contains about the 68% of the total disk mass, while Σ_0 is the normalization scale density. In Eq. (35), $\tau_\nu = \frac{R_1^2}{3(2-\phi)^2 \nu_1}$ represents the characteristic viscous timescale of the disk, and it is proportional to the inverse of the viscosity evaluated in correspondence to the scale radius ($\nu_1 = \nu(R_1)$).

We sampled a disk, made of 20 000 particles, which followed the thermal law of Eq. (30) described in the previous section, where we assumed $R_0 = 10$ AU for scaling and $T_0 = 25$ K. The SPH particle distribution was achieved by sampling the initial ($t = 0$) radial density profile as from Eq. (35),

$$\Sigma(R, 0) = \Sigma_0 \left(\frac{R}{R_1} \right)^{-\phi} \exp \left[- \left(\frac{R}{R_1} \right)^{2-\phi} \right], \quad (36)$$

where R_1 was set here to 50 AU. We used a constant value for the viscosity coefficient, $\alpha_{SS} = 10^{-2}$. The kinematic viscosity is proportional to the sound speed and the vertical scale height (Eq. (28)), and when we assume that the exponent $q = 3/7$ in the power-law Eq. (30), ν follows a radial power law with a positive exponent $\phi = 15/14 \approx 1.07$. In order to reproduce the effects of such a radial viscosity law, we imposed a specific prescription

for the artificial viscosity in the code. Taking the law of Eq. (33) into account and considering that we used $\beta = 2\alpha$, we obtain the following expression for the coefficient α :

$$\alpha = \alpha_{SS} \frac{H}{h} \frac{1}{31/525 + (9/35\pi)(h/H)}, \quad (37)$$

with constant $\alpha_{SS} = 10^{-2}$. To apply this form, we inserted the expression above in the artificial viscosity term of Eq. (11), which we used in all our disk simulations, without considering the Morris & Monaghan (1997) variation law.

The (infinite) disk has been truncated at $R_{\text{out}} = 8R_1 = 400$ AU. The distribution was also truncated at an inner cutoff $R_{\text{in}} = R_1/5 = 10$ AU. The gravitational softening radius of the central star together with its sink radius were set equal to R_{in} . The inner border condition we set is that the gas particles that cross the R_{in} radius are absorbed by the star and are therefore excluded from time integration. The mass of the sunk SPH particles was considered to increase the mass of the central star.

The ratio $R_{\text{out}}/R_{\text{in}} \approx 40$ is high enough to match the infinite extension of the analytical density profile as accurately as possible. This reduces the external radial boundary discontinuity. Hartmann et al. (1998) indeed pointed out that the inward flux of matter (which constitutes the most important process in guiding the evolution of an α disk) considerably depends on the outward angular momentum transportation and thus on the disk expansion through the external shells.

The boundary conditions at $R_{\text{in}} > 0$ may affect the disk evolution throughout the whole spatial extension. As pointed out in Lynden-Bell & Pringle (1974) and remarked by Hartmann et al. (1998), a viscous disk may formally extend to $R \rightarrow 0$, but at a critical radius that is on the same order as the radius of the star, both the torque and the viscosity ν go to zero. Computational efficiency purposes prevent us from modeling a disk by introducing a very small cutoff radius. We imposed zero viscosity ν by varying the coefficient α within $R = 3R_{\text{in}}$ down to zero, assumed at $R = R_{\text{in}}$.

Parameter α in Eq. (37) varies with time because the disk evolution processes lead to a time variation of the ratio between h and the height scale H . For $R = R_1/2$ the disk initially has an average $h/H \approx 2$, while this ratio reaches a minimum value of about 1.5, in correspondence to $R = 2R_1$, which provides a value of $\alpha \approx 0.02$ and $\alpha \approx 0.04$, respectively. As the gas is captured by the central star, we expect the density to decrease and the h -to- H ratio to increase. We show below that the surface density varies quickly during an initial phase and later evolves at a relatively lower rate; this causes h/H to follow the same cadence. At later stages (after about 1.6 Myr), the disk is indeed integrated with lower values of α that range from approximately 0.01 to 0.03.

Our disk is virtually non-selfgravitating because $Q \gg 2$ throughout its surface (the minimum value is about 20), even though we formally took the disk mass into account in setting the azimuthal velocity $v_\phi(R)$. For $R = R_1$, it has a vertical aspect ratio $H/R \approx 0.05$. With this setup, the disk should have a viscous evolution timescale $\tau_\nu \approx 840\,000$ yr according to the analytical model.

During the earlier integration phases, the disk experiences a fast relaxation caused by the inner cutoff. In this stage the internal density discontinuity is smoothed out and fades out near the inner border. We therefore allowed the system to relax for a time of about 50 000 yr (about 900 Keplerian orbits at $R = R_1$), which is far shorter than τ_ν and sufficient to obtain a steady state. From this time, the only expected changes in the disk density profile are those that are due to the secular viscous evolution.

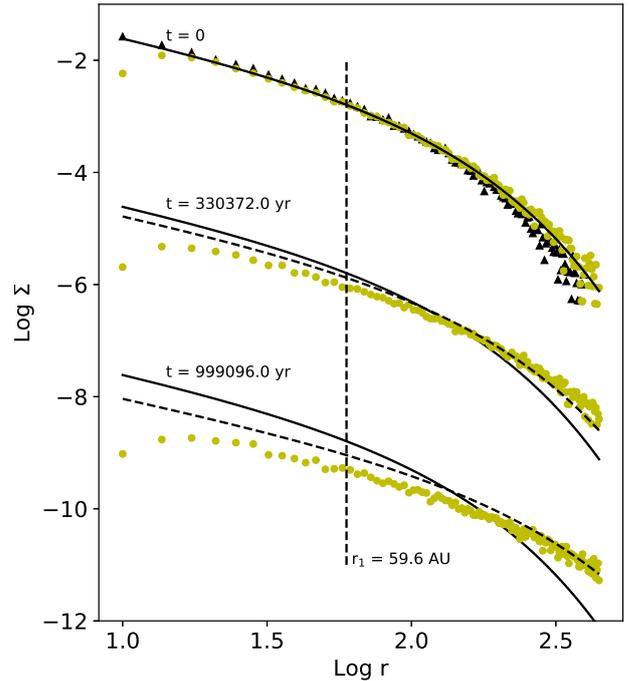


Fig. 14. Surface density profiles $\Sigma(R)$ of the disk at different times ($t = 0$ corresponds to the disk state after it has been relaxed up to a time of 50 000 yr). The dots represent the numerical results, while the lines refer to the analytical model (Eq. (35)). The actual initial disk setting (black triangles) is reported for the sake of comparison with the relaxed state. The abscissa is in AU and the density is expressed in $10^{-2} M_\odot \text{AU}^{-2}$. The evolved density profiles at $t > 0$ ($t = 330\,000$ yr and $t = 1\,000\,000$ yr) are shifted down by -3 and -6 in logarithm, respectively, for clarity. The fitting curve of the density profile at $t = 0$ is also plotted (full lines).

Thus, after 50 000 yr, the disk is quickly arranged in a configuration that is slightly different from the initial one: it is characterized by a density distribution that follows the density law of Eq. (36), but now with a larger radius $R_1 \approx 60$ AU. With this new radius, the disk has an evolution timescale $\tau_\nu \approx 10^6$ yr. The properties of self-similarity according to Eq. (35) indeed guarantee that the surface density profile maintains the same analytical form for every time. Thus, the surface density of a disk can at every time be considered an initial solution of a new disk that is described by Eq. (36), with a different parameter R_1 and thus a different viscosity timescale τ_ν . The initial profile and the density profile after 50 000 yr (which we set conventionally as the instant $t = 0$) are shown in the top panel of Fig. 15. The initial $t = 0$ state fits (full line) the disk profile of Eq. (36) with $R_1 \approx 60$ AU. We considered the disk evolution to start from this configuration and plot the density profile for various times ($t = \tau_\nu/3 \approx 330\,000$ yr and $t \approx 1\,000\,000$ yr). We also compare this with the analytical predictions obtained by Eq. (35) (dashed lines). Results and model do not match because the numerical disk appears to evolve faster than the disk predicted by the analytical theory. For $t \approx 330\,000$ yr the discrepancy between the analytical density was about 45% higher than the numerical result at $R = R_1$, while for $t \approx 1\,000\,000$ yr the discrepancy increased up to 95%. The disk thus loses mass more quickly than would be expected based on analytical theory, and thus its density decreases too rapidly. This excessive loss can be ascribed to the inner border, which is absent in the analytical theory, where the disk matter flows onto the star in a single point. To check how incorrectly the SPH disk depleted its mass, we

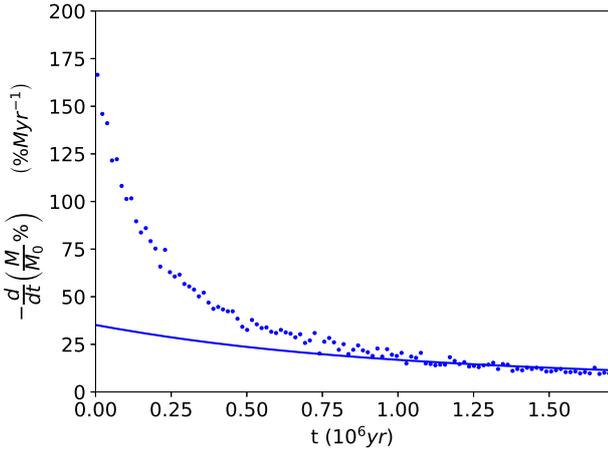


Fig. 15. Relative (percentage) mass-loss rate of the disk $-\frac{d}{dt}\left(\frac{M(t)}{M_0}\right)$, expressed in units of Myr^{-1} , and the time is in Myr. The results of our numerical simulation (dots) are compared with the theoretical behavior (full line).

considered the radial-cumulative disk mass, predicted at a given time by the analytical theory, which is strictly connected to the viscosity timescale,

$$M(\leq R, t) = M_D(0) \left(\frac{t}{\tau_v} + 1\right)^{-\frac{1}{2(2-\phi)}} \left[1 - \exp\left(\left(\frac{R}{R_1}\right)^{2-\phi} \left(\frac{t}{\tau_v} + 1\right)^{-1}\right)\right], \quad (38)$$

where $M_D(0)$ is the starting disk mass at $t = 0$. In Fig. 15 we illustrate the fractional disk mass-loss rate $-\frac{d}{dt}\left(\frac{M(t)}{M(0)}\right)$ as a function of time. The result obtained with our model is compared with the theoretical rate, where the mass as a function of time was considered as the integral of the analytical surface density in Eq. (35), from $R = 10$ AU and $R = \infty$, that is,

$$M(t) = \int_{10 \text{ AU}}^{\infty} \Sigma(R, t) 2\pi R dR = M_D(t) - M(\leq 10 \text{ AU}), \text{ where } M_D(t)$$

is the total disk mass at a given time. The SPH-disk mass-loss rate tends to reach the analytical curve only for $t \rightarrow \tau_v$, while for earlier stages, we have a huge mass-loss rate that decreases as time approaches τ_v .

To make a more substantial comparison, we moved out from the early stages and focused on the later phases at $t' = 850\,000$, where the mass-loss rate illustrated in Fig. 15 approached the analytic value to within 10%. For this time, we considered our model density distribution as the starting state of a new disk and studied its subsequent evolution. We summarize this evolution in Fig. 16. The figure shows that at t' , the disk density surface corresponds to a Lynden-Bell solution of the same form as the starting one, but with a different characteristic radius $R'_1 \approx 153 \text{ AU} \approx 2.5R_1$. This radius corresponds to a viscous timescale $\tau'_v \approx 2.4 \times 10^6$ yr. In the same figure we show the evolution of the new surface density after a time of $\tau'_v/3 \approx 790\,000$ yr, together with the theoretically expected value (dashed line). This was made by analogy with Fig. 14, where the result at $t = \tau_v/3 \approx 330\,000$ yr is shown (plotted in the middle). Comparing the two curves, we note that during the later stages the density of the SPH disk shows less deviation from the analytical prediction (within an error of 20% at $R = R'_1$), although the discrepancies are not negligible. The discrepancies are also smaller in internal regions at $R = 20 \text{ AU} \ll R'_1$, close to the inner border. To relate the discrepancy between the numerical density and its analytical prediction, we further verified this by adopting a smaller inner border radius, that is, by reducing the

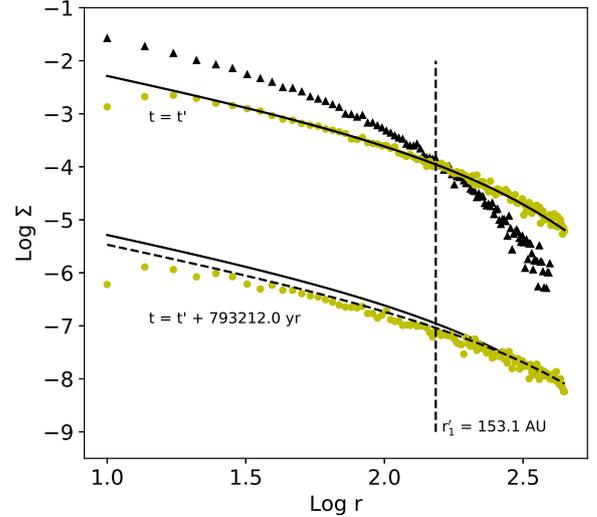


Fig. 16. Logarithm of the numerical radial profile of the disk density at $t' = 850\,000$ yr (dots). The density profile at the beginning of the simulation is also plotted (black triangles). The density at $t = t'$ matches the self-similar solution (Eq. (36)) with a parameter $R'_1 \approx 153$ AU and a viscosity timescale $\tau'_v = 2.4 \times 10^6$ yr. The evolution of this disk after a time $\tau'_v/3$ is also plotted (bottom, artificially shifted by -3) together with the analytical prediction (dashed line). The fit density profile at $t = t'$ is plotted twice (in the top and bottom curves) as a full line for clarity.

star sink from 10 AU to 5 AU, and we indeed observed a substantial reduction of the inner mass-loss rate of the disk in the initial integration phases. We also note that the gap of the surface density within the star sink radius is reduced in amplitude when a smaller inner boundary is chosen.

4.2. Self-gravitating disk in a binary system

We tested our code on a more complex dynamical system by treating the evolution of a self-gravitating disk that interacts with a binary star. The system is characterized by a circumprimary disk around a $1 M_\odot$ star, truncated by the gravitational field of a $0.4 M_\odot$ external companion star. This topic has been treated in a relevant work by Marzari et al. (2009), who integrated the time evolution of this configuration and studied the effects of the stars on the parameters of the orbital disk: eccentricity and argument of periastron. The authors used the well-known Eulerian code FARGO, implemented with a full scheme for the self-gravity (see Masset 2000; Baruteau & Masset 2008), and performed a 2D simulation. We investigated the evolution of a system like this with GaSPH with a 3D model for the particular case of a binary with eccentricity 0.4.

4.2.1. System setup

The orbit of the binary system is characterized by an eccentricity $e_b = 0.4$ and a semimajor axis of 30 AU. As in Marzari et al. (2009), we kept the orbit of the two stars fixed during the integration because we focus on the gravitational effects of the two stars on the disk. This means that the dynamics of the binary star is not affected by the gas feedback. The initial configuration of the disk that was adopted in the original model is characterized by a radial surface density $\Sigma \propto R^{-1/2}$ that extended from 0.5 AU to 11 AU from the central primary star. Beyond 11 AU, the density quickly fades out; the total disk mass is $M_D = 0.04 M_\odot$. Instead of using a typical flared disk, we adopted a flat disk by setting the linear vertical scale height to $H = 0.05 R$.

The choice adopted by Marzari et al. (2009) for the disk shape and the viscosity law leads to a slightly different schematization than in the model we used in Sect. 4.1.2, for which we needed to set the parameters inside our SPH 3D code. The authors used a constant kinematic viscosity ν , which leads α_{SS} to vary. Furthermore, the constant value of the aspect ratio H/R causes the speed of sound to scale as $c_s \propto R^{-1/2}$. When $\nu = \text{const.}$ and given the alpha-disk law of Eq. (28), we have that $\alpha_{SS} \propto H^{-1} c_s^{-1} = H^{-2} \Omega^{-1} \propto R^{-1/2}$. The coefficient α_{SS} was indeed set by the above authors by calibrating it to correspond to the value $\alpha_{SS} = 2.5 \times 10^{-3}$ in the central regions about 5 AU within the disk. Thus, we can deduce it as follows:

$$\alpha_{SS} = 2.5 \times 10^{-3} \left(\frac{R}{R_{\text{ref}}} \right)^{-1/2}, \quad (39)$$

with $R_{\text{ref}} = 5$ AU. We applied the artificial viscosity term by setting the SPH α parameter according to the same expression of Eq. (37) as we used in the previous section, but now using the nonconstant coefficient α_{SS} with the radial profile of Eq. (39) illustrated above.

The disk is coplanar with the star orbit, and we built it by confining a set of SPH particles between $R = 0.5$ AU and $R = 11$ AU. We integrated the system for about 3000 yr. All the gas particles that flew across the inner border were excluded from the integration. Three runs for the same model were made with different particle numbers, $N = 20\,000$, $N = 50\,000$, and $N = 100\,000$.

As in the case of protoplanetary disk around one star we discussed in the previous section, the ratio h/H , and consequently α , are not constant in time. For $N = 20\,000$, the disk acquires a steady state configuration after an initial quick relaxation phase, where h/H has a rather slow evolution along the timescale of a binary rotation period. After about 500 yr, the disk acquires an average ratio $h/H \approx 6.5$ ($\alpha \approx 0.001$) in the inner regions ($R = 1$ AU), which reaches a minimum of about 1 ($\alpha \approx 0.011$) in the middle regions ($R = 5.5$ AU). Similarly, the disk with $N = 50\,000$ has an average ratio $h/H \approx 3.7$ in the inner regions, with a minimum of $h/H \approx 0.7$, corresponding to $\alpha \approx 0.004$ and $\alpha \approx 0.02$. The disk with $N = 100\,000$ has $\alpha \approx 0.008$ in the internal regions and a maximum $\alpha \approx 0.03$ in the intermediate radial regions. Regardless of the resolution, after an integration time of 3000 yr, the h -to- H ratio slightly changes in the inner regions while its minimum substantially increases, giving rise to a maximum value of α of about 0.008, 0.015, and 0.02 for the disk with the lower, medium, and higher particle numbers, respectively.

4.2.2. Disk deformation and gravitational feedback to the stars

The gravitational field of the stars affects the disk configuration by altering its average orbital parameters such as the eccentricity and the argument of periastron. In order to describe the disk evolution, we calculated the mean eccentricity and the mean argument of periastron by averaging over the disk surface, with the same prescription as was adopted by Pierens & Nelson (2007),

$$e_{\text{disk}} = \frac{1}{M_{\text{D}}} \sum_i e_i(R, \phi) m_i, \quad R_A \leq R \leq R_B$$

$$\omega_{\text{disk}} = \frac{1}{M_{\text{D}}} \sum_i \omega_i(R, \phi) m_i, \quad R_A \leq R \leq R_B. \quad (40)$$

Here M_{D} is the disk mass included within R_A and R_B (the latter being a quantity on the order of the effective radius R_{D}). The disk

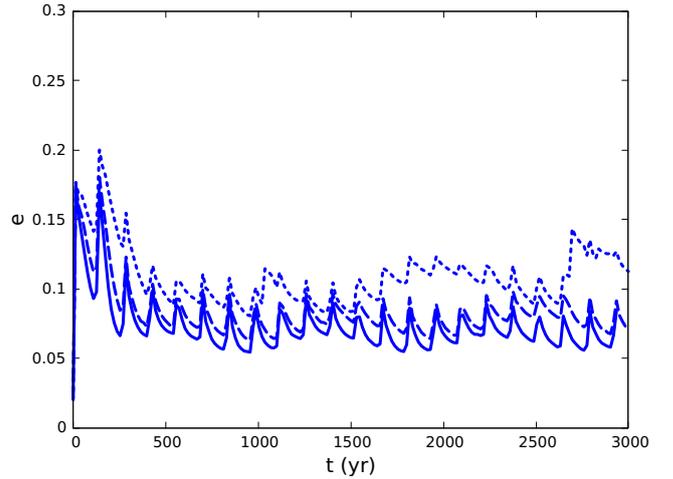


Fig. 17. Disk eccentricity, e_{disk} , evolution under the perturbation of a binary system of $e = 0.4$. Values referring to different simulations are plotted: $N = 20\,000$ (dotted line), $N = 50\,000$ (dashed line), and $N = 100\,000$ (full line).

radius is defined by the following expression:

$$R_{\text{D}} \equiv \langle R_{\text{D}} \rangle_{\text{L}} \propto \left(\frac{L}{M_{\text{D}}} \right)^2 \quad (41)$$

and is computed as the radial distance containing the total angular momentum L of the disk, with M_{D} its total mass. R_{D} is not very different from the half-mass radius.

The local orbital parameters are evaluated using the eccentricity vector,

$$\mathbf{e} = \frac{\mathbf{r} \times \mathbf{l}}{GM_{\text{c}}} - \hat{\mathbf{r}}, \quad (42)$$

where $\mathbf{l} = \mathbf{r} \times \mathbf{v}$ represents the angular momentum per unit mass, and M_{c} is the mass of the central star. The vector \mathbf{e} characterizes the orbit described by the position \mathbf{r} of a point about the center of mass of the binary, assuming that it corresponds to an elliptical trajectory. The absolute value $e = |\mathbf{e}|$ corresponds to the orbital eccentricity. Moreover, \mathbf{e} is always parallel to the semi-major axis, thus, its normalized components $e_x/e = \cos(\omega)$ and $e_y/e = \sin(\omega)$ provide the local argument of periastron ω , and finally, the local semimajor axis. In Eq. (40), only the particles on orbits that are tied to the central primary star are considered part of the disk and thus are included in the summation.

The choice of a 3D model introduces some new degrees of freedom with respect to a 2D scheme because the self-gravity of the disk and the gravity of the stars act even along the vertical direction. In particular, the angular momentum, whose flux across the disk plays a crucial role on the disk evolution itself, is also spread out in the vertical direction.

Figures 17 and 18 show the evolution of the disk eccentricity and of the angle ω_{disk} , respectively. After a few orbital periods, the secondary star initially truncates the disk and a chaotic phase arises during which the mean eccentricity increases abruptly. In a second phase, the eccentricity stabilizes around an average value that is similar to the value obtained by Marzari et al. (2009), which is $e_{\text{disk}} \approx 0.075$. Moreover, in the same way as the other investigation, Fig. 17 shows that e_{disk} slightly oscillates. The oscillation is modulated with the binary period ($P_{\text{bin}} \approx 134$ yr), and can be ascribed to the strong variation of the gravitational field of the companion star at periastron.

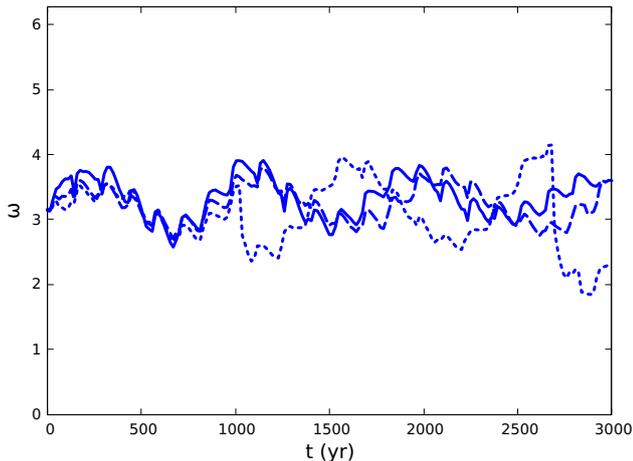


Fig. 18. Time evolution of argument of periastron, ω_{disk} , in an eccentric ($e = 0.4$) binary system. Like in Fig. 17, results for different simulations are illustrated: $N = 20\,000$ (dotted line), $N = 50\,000$ (dashed line), and $N = 100\,000$ (full line).

Similarly, Fig. 18 shows the mean inclination of the oscillations in the disk semimajor axis around the initial value (which conventionally was taken as π). Even for the argument of periastron of the disk, a convergence can be observed near the simulations with higher resolution.

5. Summary and conclusions

The primary intent of this paper is the presentation, testing, and preliminary application of our new SPH code, GaSPH, which is designed to be a multi-purpose code that is applicable to a variety of astrophysical, multi-phase, and self-gravitating environments. We briefly summarize the main points below.

- We presented and discussed the characteristics of our code in some detail. At the moment, the code does not treat radiative transfer, but properly takes internal gas gravity and the gas-star mutual gravity into account.
- The code fully passes the classic tests in slowly varying situations, which assess its stability, and also in violently varying cases that reproduce the Sedov–Taylor blast wave well.
- The code performs well numerically (speed) and has a good stability and quality, as we discussed in Sect. 3.2.
- The capability of the code to treat the evolution of a protoplanetary disk when it interacts with a single star and with a binary star was tested.

In a near future, we aim to reach a much better resolution, which is achievable with an MPI parallel version of our code. This will allow us to study disks on the smaller, planetary, scale. Further scientific applications of our code will include the evolution of protoplanetary disks in a star cluster environment in order to study the star-to-disk feedback.

Acknowledgements. We express our gratitude to Fabrizio Capaccioni of INAF-IAPS-Istituto di Astrofisica e Planetologia Spaziali (Rome, Italy) for the valuable support and useful discussions throughout the preparation of this work. We also wish to thank the anonymous referee for the suggestions and comments that greatly helped to improve the paper.

References

Aarseth, S. J. 1999, *PASP*, **111**, 1333
 Allen, M. P., & Tildesley, D. J. 1989, *Computer Simulation of Liquids* (New York, USA: Clarendon Press)
 Andersen, H. C. 1983, *J. Comput. Phys.*, **52**, 24

Armitage, P. J. 2011, *ARA&A*, **49**, 195
 Artymowicz, P., & Lubow, S. H. 1994, *ApJ*, **421**, 651
 Attwood, R. E., Goodwin, S. P., & Whitworth, A. P. 2007, *A&A*, **464**, 447
 Balsara, D. S. 1995, *J. Comput. Phys.*, **121**, 357
 Barnes, J. E. 1986, *The Use of Supercomputers in Stellar Dynamics: Proceedings of a Workshop Held at the Institute for Advanced Study Princeton, USA, June 2–4, 1986* (Berlin, Heidelberg: Springer, Berlin Heidelberg), 175
 Barnes, J., & Hut, P. 1986, *Nature*, **324**, 446
 Baruteau, C., & Masset, F. 2008, *ApJ*, **678**, 483
 Binney, J., & Tremaine, S. 1987, *Galactic dynamics* (Princeton, NJ: Princeton University Press)
 Boss, A. P. 1998, *ApJ*, **503**, 923
 Boss, A. P. 2003, *ApJ*, **599**, 577
 Capuzzo-Dolcetta, R., & Miocchi, P. 1998, *J. Comput. Phys.*, **143**, 29
 Capuzzo-Dolcetta, R., Spera, M., & Punzo, D. 2013, *J. Comput. Phys.*, **236**, 580
 Chandrasekhar, S. 1958, *An Introduction to the Study of Stellar Structure* (New York, US: Dover Publications, Inc.)
 Chapman, B., Jost, G., & Van Der Pas, R. 2007, *Using OpenMP - Portable Shared Memory Parallel Programming* (Cambridge, USA: The MIT Press)
 Ciardi, D. R., Crossfield, I. J. M., Feinstein, A. D., et al. 2018, *AJ*, **155**, 10
 D’Alessio, P., Calvet, N., Hartmann, L., Lizano, S., & Cantó, J. 1999, *ApJ*, **527**, 893
 Dullemond, C. P., Hollenbach, D., Kamp, I., & D’Alessio, P. 2007, *Protostars and Planets V*, 555
 Feagin, T. 2012, *Sci. Comput.*, **20**, 437
 Garcia, P. J. V. 2011, *Physical Processes in Circumstellar Disks Around Young Stars* (Chicago: The University of Chicago Press)
 Gingold, R. A., & Monaghan, J. J. 1977, *MNRAS*, **181**, 375
 Handy, J. 1998, *The Cache Memory Book* (USA: Academic Press, inc.)
 Hartmann, L., Calvet, N., Gullbring, E., & D’Alessio, P. 1998, *ApJ*, **495**, 385
 Hernquist, L. 1987, *ApJS*, **64**, 715
 Hernquist, L., & Katz, N. 1989, *ApJS*, **70**, 419
 Hockney, R. W., & Eastwood, J. W. 1981, *Computer Simulation Using Particles* (New York: McGraw-Hill)
 Hockney, R. W., & Eastwood, J. W. 1988, *Computer Simulation Using Particles* (Bristol: Hilger)
 Hosono, N., Saitoh, T. R., & Makino, J. 2016, *ApJS*, **224**, 32
 Hubber, D. A., Allison, R. J., Smith, R., & Goodwin, S. P. 2013, *MNRAS*, **430**, 1599
 Hut, P., Makino, J., & McMillan, S. 1995, *ApJ*, **443**, L93
 Kratter, K., & Lodato, G. 2016, *ARA&A*, **54**, 271
 Kratter, K. M., Matzner, C. D., & Krumholz, M. R. 2008, *ApJ*, **681**, 375
 Liu, C.-J., Yao, Z., & Ding, W.-B. 2017, *Res. Astron. Astrophys.*, **17**, 078
 Lucy, L. B. 1977, *AJ*, **82**, 1013
 Lynden-Bell, D., & Pringle, J. E. 1974, *MNRAS*, **168**, 603
 Muzari, F., Scholl, H., Thébaud, P., & Baruteau, C. 2009, *A&A*, **508**, 1493
 Masset, F. 2000, *A&AS*, **141**, 165
 Mayer, L., Quinn, T., Wadsley, J., & Stadel, J. 2002, *Science*, **298**, 1756
 Meglicki, Z., Wickramasinghe, D., & Bicknell, G. V. 1993, *MNRAS*, **264**, 691
 Meru, F., & Bate, M. R. 2012, *MNRAS*, **427**, 2022
 Miocchi, P., & Capuzzo-Dolcetta, R. 2002, *A&A*, **382**, 758
 Monaghan, J. J. 1988, *Comput. Phys. Commun.*, **48**, 89
 Monaghan, J. J. 1989, *J. Comput. Phys.*, **82**, 1
 Monaghan, J. J. 1992, *ARA&A*, **30**, 543
 Monaghan, J. J. 2005, *Rep. Progr. Phys.*, **68**, 1703
 Monaghan, J. J., & Lattanzio, J. C. 1985, *A&A*, **149**, 135
 Morris, J. P., & Monaghan, J. J. 1997, *J. Comput. Phys.*, **136**, 41
 Nelson, A. F., Benz, W., Adams, F. C., & Arnett, D. 1998, *ApJ*, **502**, 342
 Picogna, G., & Marzari, F. 2013, *A&A*, **556**, A148
 Pierens, A., & Nelson, R. P. 2007, *A&A*, **472**, 993
 Plummer, H. C. 1911, *MNRAS*, **71**, 460
 Price, D. J., & Monaghan, J. J. 2007, *MNRAS*, **374**, 1347
 Pringle, J. E. 1981, *ARA&A*, **19**, 137
 Quinn, T., Katz, N., Stadel, J., & Lake, G. 1997, ArXiv e-prints [arXiv:astro-ph/9710043]
 Rosswog, S., & Price, D. 2007, *MNRAS*, **379**, 915
 Rosswog, S., Davies, M. B., Thielemann, F.-K., & Piran, T. 2000, *A&A*, **360**, 171
 Saitoh, T. R., & Makino, J. 2009, *ApJ*, **697**, L99
 Sedov, L. 1959, *Similarity and Dimensional Methods in Mechanics* (New York: CRC Press Inc.)
 Shakura, N. I., & Sunyaev, R. A. 1973, *A&A*, **24**, 337
 Springel, V. 2005, *MNRAS*, **364**, 1105
 Springel, V., & Hernquist, L. 2002, *MNRAS*, **333**, 649
 Stamatellos, D., Maury, A., Whitworth, A., & André, P. 2011, *MNRAS*, **413**, 1787
 Tasker, E. J., Brunino, R., Mitchell, N. L., et al. 2008, *MNRAS*, **390**, 1267
 Toomre, A. 1964, *ApJ*, **139**, 1217

Appendix A: Details of the numerical method

A.1. Velocity-Verlet method

At each time iteration, \mathbf{a} and \dot{u} are evaluated twice, in correspondence to the current step n and the next $n + 1$. Starting from a generic n th time iteration, we use $\mathbf{a}^{[n]}$ and $\dot{u}^{[n]}$ to predict the velocity and the energy:

$$\begin{cases} \mathbf{v}^{*[n+1]} = \mathbf{v}^{[n]} + \mathbf{a}^{[n]}\Delta t \\ \mathbf{u}^{*[n+1]} = \mathbf{u}^{[n]} + \dot{u}^{[n]}\Delta t \end{cases}, \quad (\text{A.1})$$

while the position is directly updated to the next step $n + 1$, without any prediction:

$$\mathbf{r}^{[n+1]} = \mathbf{r}^{[n]} + \mathbf{v}^{[n]}\Delta t + \frac{1}{2}\mathbf{a}^{[n]}\Delta t^2. \quad (\text{A.2})$$

With these new quantities, a new calculation is performed for \mathbf{a} and \dot{u} ; we thus have

$$\begin{cases} \mathbf{a}^{*[n+1]} = \mathbf{a}(\mathbf{r}^{[n+1]}, \mathbf{v}^{*[n+1]}, u^{*[n+1]}) \\ \dot{u}^{*[n+1]} = \dot{u}(\mathbf{r}^{[n+1]}, \mathbf{v}^{*[n+1]}, u^{*[n+1]}) \end{cases}, \quad (\text{A.3})$$

which we can use to correct velocity and energy,

$$\begin{cases} \mathbf{v}^{[n+1]} = \mathbf{v}^{[n]} + (\mathbf{a}^{[n]} + \mathbf{a}^{*[n+1]})\frac{\Delta t}{2} \\ \mathbf{u}^{[n+1]} = \mathbf{u}^{[n]} + (\dot{u}^{[n]} + \dot{u}^{*[n+1]})\frac{\Delta t}{2} \end{cases}. \quad (\text{A.4})$$

It can be straightforwardly shown that when acceleration only depends on the positions, that is, in a Newtonian problem without hydrodynamics, the Verlet method described above is equivalent to a standard second-order kick-drift-kick (KDK) leap-frog method. When the acceleration depends neither on the velocity field nor on the internal energy, the quantity $\mathbf{a}^{*[n+1]}$ corresponds to the actual acceleration $\mathbf{a}^{[n+1]}$ that is related to the next step. The numerical method can thus be rewritten in the following way:

$$\begin{cases} \mathbf{r}^{[n+1]} = \mathbf{r}^{[n]} + \left(\mathbf{v}^{[n]} + \frac{1}{2}\mathbf{a}^{[n]}\Delta t\right)\Delta t = \mathbf{r}^{[n]} + \mathbf{v}^{[n+1/2]}\Delta t \\ \mathbf{v}^{[n+1]} = \mathbf{v}^{[n]} + \frac{1}{2}\mathbf{a}^{[n]}\Delta t + \frac{1}{2}\mathbf{a}^{[n]}\Delta t = \mathbf{v}^{[n+1/2]} + \mathbf{a}^{[n]}\frac{\Delta t}{2} \end{cases}, \quad (\text{A.5})$$

which represents indeed the standard expression of a KDK leap-frog integrator that requires just one force calculation per time step (see [Hockney & Eastwood 1988](#); [Hut et al. 1995](#); [Quinn et al. 1997](#), for leap-frog methods and further improvements).

Each particles has its own individual time steps, sorted by the code as submultiples of the maximum time step Δt_{\max} . For a generic particle of index i , the time step is calculated according to the criteria expressed by Eqs. (17) and (18) and approximated to the nearest value $\Delta t_i = 2^{-P} \cdot \Delta t_{\max}$, where P is a positive integer number. The simple scheme in Fig. A.1 shows that a single time iteration between two consecutive steps n and $n + 1$ is performed between the time $t^{[n]}$ and $t^{[n+1]} = t^{[n]} + \Delta t_{\min}$, while a generic particle of index i is updated between two characteristic times: $t_i^{[\text{PREV}]}$ and $t_i^{[\text{NEXT}]} = t_i^{[\text{PREV}]} + \Delta t_i$. The routines dedicated to the neighbor search, to the hydrodynamic forces, and to the gas self-gravity are activated for the particle i only when its time step is synchronized, that is, if the conditions $t^{[n]} = t_i^{[\text{PREV}]}$ or $t^{[n+1]} = t_i^{[\text{NEXT}]}$ are satisfied. Figure A.2 shows a scheme of a single time iteration. First, the code sorts and maps all the particles by building the tree, calculating the quadrupole momentum

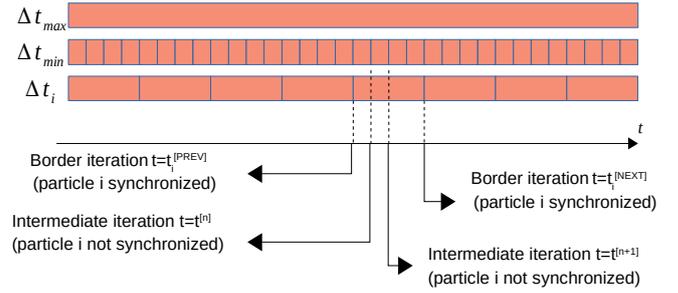


Fig. A.1. Sketch of the hierarchical time-step subdivision. A generic individual time step, Δt_i is a power of two multiples of the minimum time step Δt_{\min} , and a power of two submultiple of the maximum one, Δt_{\max} . The integration is performed by means of elementary iterations from $t^{[n]}$ to $t^{[n+1]} = t^{[n]} + \Delta t_{\min}$, while the particle is updated from $t_i^{[\text{PREV}]}$ to $t_i^{[\text{NEXT}]} = t_i^{[\text{PREV}]} + \Delta t_i$. The hydrodynamics and force routines are activated only for particles that are synchronized at $t^{[n]} = t_i^{[\text{PREV}]}$ or at $t^{[n+1]} = t_i^{[\text{NEXT}]}$.

and all the other key quantities related to the cubes. This operation is thus independent on the particle time steps. Then, the code runs the main cycles of neighbor search and acceleration computation. For each i th particle, it verifies whether the particle is synchronized at $t = t^{[n]}$, that is, whether the condition $t^{[n]} = t_i^{[\text{PREV}]}$ is satisfied. In that case, the algorithm computes the density $\rho_i^{[\text{PREV}]}$, the pressure $P_i^{[\text{PREV}]}$, the velocity gradient $\nabla \cdot \mathbf{v}_i^{[\text{PREV}]}$ and the velocity rotor $\nabla \times \mathbf{v}_i^{[\text{PREV}]}$, the switching coefficient of Eq. (12), the quantities $\omega_i^{[\text{PREV}]}$, $\zeta_i^{[\text{PREV}]}$, the hydro-gravitational acceleration $\mathbf{a}_i^{[\text{PREV}]}$, and the time variation of the internal energy $\dot{u}_i^{[\text{PREV}]}$. These quantities are suitably stored in memory, to be used in further phases of the integration and in further iterations. For the remaining nonsynchronized particles, the algorithm indeed uses the quantities that were calculated in previous stages.

If some stars are included in the simulation, $\mathbf{a}_i^{[\text{PREV}]}$ is incremented by a contribution from the star-gas interaction for every i th particle.

After this, we update the time of \mathbf{v} , \mathbf{u} , and \mathbf{r} . For particles synchronized at $t = t^{[n+1]}$, so that $t_i^{[\text{NEXT}]} = t^{[n+1]}$, the velocity and the energy are updated with the same predictor scheme as expressed by Eq. (A.1), thus, we have

$$\begin{cases} \mathbf{v}_i^{*[n+1]} = \mathbf{v}_i^{*[\text{NEXT}]} = \mathbf{v}_i^{[\text{PREV}]} + \mathbf{a}_i^{[\text{PREV}]}\Delta t_i \\ \mathbf{u}_i^{*[n+1]} = \mathbf{u}_i^{*[\text{NEXT}]} = \mathbf{u}_i^{[\text{PREV}]} + \dot{u}_i^{[\text{PREV}]}\Delta t_i \end{cases}. \quad (\text{A.6})$$

The position is updated as well, in the same manner as indicated by Eq. (A.2):

$$\mathbf{r}_i^{[n+1]} = \mathbf{r}_i^{[\text{NEXT}]} = \mathbf{r}_i^{[\text{PREV}]} + \mathbf{v}_i^{[\text{PREV}]}\Delta t_i + \frac{1}{2}\mathbf{a}_i^{[\text{PREV}]}\Delta t_i^2. \quad (\text{A.7})$$

On the other hand, for nonsynchronized points, we do not update, but estimate the quantities $\mathbf{v}^{*[n+1]}$, $\mathbf{u}^{*[n+1]}$ and $\mathbf{r}^{[n+1]}$ by means of the following predictor scheme:

$$\begin{cases} \mathbf{v}_i^{*[n+1]} = \mathbf{v}_i^{[\text{PREV}]} + \mathbf{a}_i^{[\text{PREDICT}]}\delta t_i \\ \mathbf{u}_i^{*[n+1]} = \mathbf{u}_i^{[\text{PREV}]} + \dot{u}_i^{[\text{PREV}]}\delta t_i \\ \mathbf{r}_i^{[n+1]} = \mathbf{r}_i^{[\text{PREV}]} + \mathbf{v}_i^{[\text{PREDICT}]}\delta t_i + \frac{1}{2}\mathbf{a}_i^{[\text{PREDICT}]}\delta t_i^2, \end{cases} \quad (\text{A.8})$$

which contains the following quantities:

$$\delta t_i = t^{[n+1]} - t_i^{[\text{PREV}]}$$

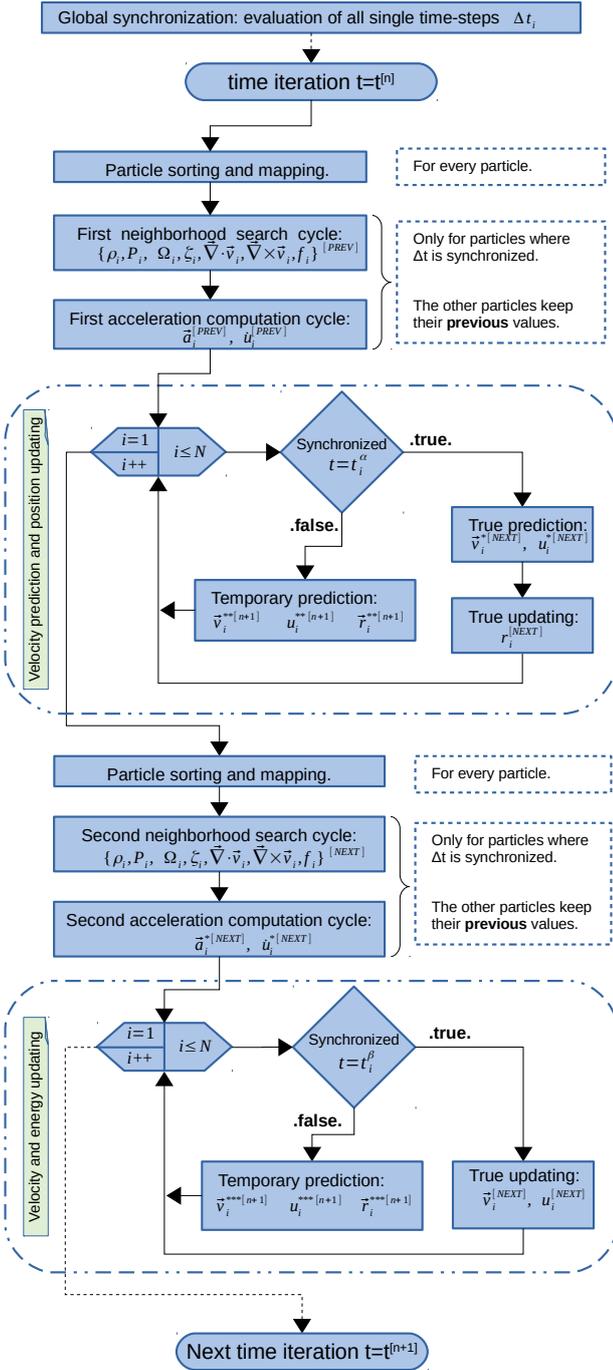


Fig. A.2. Flow chart showing the main scheme of a single time iteration.

$$\mathbf{a}_i^{[\text{PREDICT}]} = \frac{1}{2} (\mathbf{a}_i^{[\text{PREV}]} + \mathbf{a}_i^{[n]})$$

$$\mathbf{v}_i^{[\text{PREDICT}]} = \frac{1}{2} (\mathbf{v}_i^{[\text{PREV}]} + \mathbf{v}_i^{*[n+1]})$$

$\mathbf{a}_i^{[n]}$ is exactly equal to the old value $\mathbf{a}_i^{[\text{PREV}]}$ in simulations with pure gas because the algorithm does not compute the acceleration at the current iteration. When one or more stars interact with the gas, however, the two accelerations are different because they contain different contributions from the gas-star interaction, which is independently calculated at every step in the particle synchronization.

Based on the new velocities, energies, and space coordinates, a second phase of recalculating the hydrodynamical variables and the accelerations occurs, of course, which is limited to the synchronized points at $t = t^{[n+1]}$. Thus, each synchronized i th particle owns the following updated hydrodynamical quantities: $\rho_i^{[\text{NEXT}]}$, $P_i^{[\text{NEXT}]}$, $\nabla \cdot \mathbf{v}_i^{[\text{NEXT}]}$, $\nabla \times \mathbf{v}_i^{[\text{NEXT}]}$, $f_i^{[\text{NEXT}]}$, $\omega_i^{[\text{NEXT}]}$, and $\zeta_i^{[\text{NEXT}]}$. The accelerations and energy can thus be computed in the same manner as shown by Eq. (A.3):

$$\begin{cases} \mathbf{a}_i^{*[n+1]} = \mathbf{a}_i^{*[\text{NEXT}]} = \mathbf{a}(\mathbf{r}_i^{[\text{NEXT}]}, \mathbf{v}_i^{*[\text{NEXT}]}, u_i^{*[\text{NEXT}]}) \\ \dot{u}_i^{*[n+1]} = \dot{u}_i^{*[\text{NEXT}]} = \dot{u}(\mathbf{r}_i^{[\text{NEXT}]}, \mathbf{v}_i^{*[\text{NEXT}]}, u_i^{*[\text{NEXT}]}) \end{cases} \quad (\text{A.9})$$

Then, by applying the same formalism as expressed by Eq. (A.4), energy and velocities can be finally updated:

$$\begin{cases} \mathbf{v}_i^{[n+1]} = \mathbf{v}_i^{[\text{NEXT}]} = \mathbf{v}^{[\text{PREV}]} + (\mathbf{a}_i^{[\text{PREV}]} + \mathbf{a}_i^{*[\text{NEXT}]}) \frac{\Delta t_i}{2} \\ u_i^{[n+1]} = u_i^{[\text{NEXT}]} = u_i^{[\text{PREV}]} + (\dot{u}_i^{[\text{PREV}]} + \dot{u}_i^{*[\text{NEXT}]}) \frac{\Delta t_i}{2} \end{cases} \quad (\text{A.10})$$

A.2. 14th-order Runge–Kutta method

For a generic set of N_{ob} objects we wish to integrate the following differential equations that are associated with a generic i th object:

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i \quad ; \quad \frac{d\mathbf{v}_i}{dt} = \mathbf{f}_i. \quad (\text{A.11})$$

The method begins with a first estimation of the explicit derivatives at the iteration n :

$$\begin{cases} \mathbf{K}\mathbf{v}_1^{(i)} = \mathbf{f}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \dots, \mathbf{r}_i^{[n]}, \dots, \mathbf{r}_N^{[n]}) \\ \mathbf{K}\mathbf{r}_1^{(i)} = \mathbf{v}_i^{[n]} \end{cases}, \quad (\text{A.12})$$

which can be used to estimate the further quantities corresponding to a second substep $n + c_2$:

$$\begin{aligned} \mathbf{K}\mathbf{v}_2^{(i)} &= \mathbf{f}(\mathbf{r}_1^{[n+c_2]}, \mathbf{r}_2^{[n+c_2]}, \dots, \mathbf{r}_i^{[n+c_2]}, \dots, \mathbf{r}_N^{[n+c_2]}) \\ \mathbf{K}\mathbf{r}_2^{(i)} &= \mathbf{v}_i^{[n]} + a_{21} \mathbf{K}\mathbf{v}_1^{(i)} \Delta t, \end{aligned} \quad (\text{A.13})$$

where $\mathbf{r}_i^{[n+c_2]} = \mathbf{r}_i^{[n]} + a_{21} \mathbf{K}\mathbf{r}_1^{(i)} \Delta t$ represents the i th particle position updated to an intermediate time $t + c_2 \Delta t$. In the same manner, further consecutive estimations of β th terms can be performed:

$$\begin{aligned} \mathbf{K}\mathbf{v}_\beta^{(i)} &= \mathbf{f}(\mathbf{r}_1^{[n+c_\beta]}, \mathbf{r}_2^{[n+c_\beta]}, \dots, \mathbf{r}_i^{[n+c_\beta]}, \dots, \mathbf{r}_N^{[n+c_\beta]}) \\ \mathbf{K}\mathbf{r}_\beta^{(i)} &= \mathbf{r}_i^{[n]} + \sum_{\gamma=1}^{\beta-1} a_{\beta\gamma} \mathbf{K}\mathbf{v}_\gamma^{(i)} \Delta t, \end{aligned} \quad (\text{A.14})$$

with $\mathbf{r}_i^{[n+c_\beta]} = \mathbf{r}_i^{[n]} + \sum_{\gamma=1}^{\beta-1} a_{\beta\gamma} \mathbf{K}\mathbf{r}_\gamma^{(i)} \Delta t$ the vector position of the i th particle at a generic intermediate time $t + c_\beta \Delta t$. Because $\gamma < \beta$, every quantity explicitly depends on previous estimations. The coefficients $a_{\beta\gamma}$ are the elements of a 35×34 matrix, while b_β and c_β represent two arrays of 35 elements. The matrix a requires the following restriction:

$$\sum_{\gamma=1}^{34} a_{\beta\gamma} = c_\beta. \quad (\text{A.15})$$

Since we work with a fully explicit method, a is triangular, and $a_{\beta\gamma} = 0$, for $\gamma > \beta$. In total, each star will have 35 velocity RK coefficients $\mathbf{K}\mathbf{r}_\beta$ and 35 acceleration RK coefficients $\mathbf{K}\mathbf{v}_\beta$, the resulting velocity and position at the next time step will be given by

$$\begin{aligned} \mathbf{v}_i^{[n+1]} &= \mathbf{v}_i^{[n]} + \sum_{\beta=1}^{35} b_\beta \mathbf{K}\mathbf{v}_\beta^{(i)} \Delta t \\ \mathbf{r}_i^{[n+1]} &= \mathbf{r}_i^{[n]} + \sum_{\beta=1}^{35} b_\beta \mathbf{K}\mathbf{r}_\beta^{(i)} \Delta t. \end{aligned} \quad (\text{A.16})$$

To integrate the time evolution of a system composed of both gas and stars, we coupled the Verlet and the RK integration methods as follows. At the starting iteration n , the gas particles feel the gravity field from the stars, the mutual interactions of the SPH, and eventually, its self-gravity. Then, $\mathbf{v}^{*[n+1]}$ and $\mathbf{u}^{*[n+1]}$ are predicted, and their positions are thus updated according to Eq. (A.2). At the same time, the star positions and velocities were first updated with the Runge–Kutta method, then, the explicit force contributions due to the SPH particles $\mathbf{a}_{\text{part}}^{[n]}$ were added to Eq. (A.16). Stars and SPH particles were coupled by direct point-to-point interaction, without any approximation for the gravitational field. Finally, we corrected the gas positions and velocity according to Eq. (A.4) by recalculating the accelerations and the energy rates at the new stage $n + 1$.

Appendix B: Technical features

B.1. Neighbor search and acceleration updating

Before we employed it to compute gravitational interactions, the tree grid was also used to support the operations related to the nearest-neighbor search. To calculate the density and the smoothing length, each i th particle starts with a first-guess value of h_i and makes a tree walk by adopting an opening criterion that is slightly different from Eqs. (20) and (21). Given a generic particle of index i , in order to find the other points enclosed in its SPH kernel support, we checked for each cube the overlap with a sphere of radius $2h_i$ centered onto the point $\mathbf{r}_i = (x_i, y_i, z_i)$. A box is opened only if the following three conditions are valid:

$$\begin{aligned} |x_i - x_A| &< 2h_i + \frac{D_L}{2} \\ |y_i - y_A| &< 2h_i + \frac{D_L}{2} \\ |z_i - z_A| &< 2h_i + \frac{D_L}{2}, \end{aligned} \quad (\text{B.1})$$

where (x_A, y_A, z_A) are the coordinates of the geometrical center of the cube and D_L is its side length. The tree walk continues by opening the further cubes according to the last rule, until the single particles are reached and the neighborhood is thus determined. At the end of the walk, a temporary value of density ρ_i is calculated. If the number of encountered neighbors differs from the expected number (and the difference exceeds the tolerance number), h_i is updated according to Eq. (5). Then, a new tree walk is performed, and a new value for the density is computed. The tree walk and the h_i, ρ_i updating are executed cyclically until the number of neighbors converges on the desired value. During the tree walk, the various quantities $P_i, \Omega_i, \zeta_i, f, \nabla \cdot \mathbf{v}_i$, and $\nabla \times \mathbf{v}_i$ are computed at the same time together with the density.

After this preliminary phase, the acceleration \mathbf{a} and the thermal energy rate \dot{u} can be computed by performing a further tree

walk for each particle. During the walk, the hydrodynamics contribution to the acceleration and the gravity field are computed at the same time. For each box, before Eq. (20) (or Eq. (21)) is applied, the code primarily checks Eq. (B.1) to distinguish cubes that might contain some neighbor particles for the SPH interpolation. If this is satisfied, the box is opened, otherwise the particles inside will not contribute at all to hydrodynamics. After this primary check, the algorithm performs a secondary control by applying the opening criterion of Eq. (20) (or Eq. (21)) to decide whether the multipole approximation can be applied to the Newtonian field. As a result, practically, a generic i th particle will interact with its local SPH neighborhood following Eqs. (7) and (8), experiencing the Newtonian force through a direct particle-to-particle interaction. The remaining particles that lie outside the SPH domain contribute to the accelerations with or without multipole approximation, according to the basic criteria of Eq. (20) or (21).

During the computation of acceleration terms, \mathbf{a}_i , the neighbor domain search process may suffer some technical problems. The interpolation of the density (Eq. (6)) together with interpolations (9), (10), (12), and (13) are rather straightforward to perform because they require a sum over a domain that only depends on the local h_i . On the other hand, Eq. (7) contains sums that extend over a more complex region because the W and g_{soft} functions depend not only on the local smoothing length, but also on the h_j , which characterizes the domain extension of a surrounding j th particle. The condition of Eq. (B.1) guarantees that we can find all the nearest j th particles with a smoothing length $h_j \leq h_i$. Nevertheless, particles with $h_j > h_i$ also exist for which $2h_j$ is smaller than the mutual distance r_{ij} . They give a non-null contribution to the acceleration of the i th point, but they are excluded from the neighbor search, and the code does not take them into account, which leads to an incorrect implementation of the SPH equations. It is rather easy to solve this problem when the algorithm works with a uniform time step. Equation (7) contains pairs of terms that are symmetric with respect to the swap of indexes $i - j$; moreover, one term of each pair depends only on h_i , while the other depends only on h_j . This means that when the i th particle walks along the tree and finds the j th particle within the interpolation domain $2h_i$, the code may add to \mathbf{a}_i the quantity $-\frac{1}{2}g_{\text{soft}}(r, h_i)\mathbf{r}/r - (\chi_{Ai} + \chi_{Bi} + \chi_{Ci})\nabla_i W(h_i)$, and may add to \mathbf{a}_j the same quantity with the opposite sign. Here we used the following expressions for a generic particle of i th index:

$$\begin{aligned} \chi_{Ai} &= m_i \frac{G}{2} \frac{\zeta_i}{\Omega_i} \\ \chi_{Bi} &= m_i \frac{P_i}{\rho_i^2 \Omega_i} \\ \chi_{Ci} &= \frac{1}{2} m_i \Pi_{ij}. \end{aligned} \quad (\text{B.2})$$

When the j th particle in turn makes the tree walk, the code can find the i th particle within the domain $2h_j$. In this case, the quantity $-\frac{1}{2}g_{\text{soft}}(r, h_j)\mathbf{r}/r - (\chi_{Aj} + \chi_{Bj} + \chi_{Cj})\nabla_j W(h_j)$ is added to \mathbf{a}_j and the same quantity with opposite sign is added to \mathbf{a}_i .

This technique cannot be applied when an individual time step is assigned to each particle because during the force computation some particles are inactive and the symmetry of Eq. (7) is thus broken. When the i th particle is active and the j th particle is inactive and lies outside the radius $2h_i$, the algorithm adds to \mathbf{a}_i the full set of terms in Eq. (7) to take into account the contributions of the j th point, and it increases the neighbor-search radius. This means that only during the second-step tree walk does the code apply the criterion in Eq. (B.1) with a slight modification of

the radius because a value higher than $2h_i$ is used, which is estimated as follows. The algorithm calculates a local interpolation of the gradient of the softening length, estimated as

$$\nabla h_i = \frac{\partial h_i}{\partial \rho_i} \nabla \rho_i = -\frac{1}{3} \frac{h_i}{\rho_i} \sum_j m_j \nabla W(r_{ij}, h_i), \quad (\text{B.3})$$

which is computed in the same loop as was used for the density. Then, in place of h_i , the code uses the quantity $h_i \cdot \max(\delta_h h_i^{[\text{MAX}]} / h_i, 1 + |\nabla h_i|)$, with δ_h a suitable constant that we set to 1.3. The quantity $h_i^{[\text{MAX}]}$ represents an estimate of the maximum local value of the softening length. It tries to probe all the neighbors j th particles, including the points outside the radius $2h_i$, that can make an SPH interaction with the i th particle, that is, such that $h_j > h_i \vee 2h_j > r_{ij}$. In an earlier time step, when the j th particle is active, we therefore determine whether the length h_j is greater than the local maximum length around the i th position. This scheme increases the CPU time by increasing the direct point-to-point interactions. Nevertheless, it involves just local interactions, and as N increases, the CPU efforts become increasingly less relevant compared with the nonlocal gravity computations that are performed with the tree scheme. This method empirically represents a correct technique for finding the neighbors of a point in case of individual time steps, even though it cannot be mathematically proven that it is able to find 100% of the effective neighbor particles in all possible density configurations. We conducted a series of tests in which we even employed extreme density contrasts like in the case of a Sedov–Taylor blast-wave profile, and found that 100% of the effective neighbor particles were found when δ_h was above 1.2.

B.2. Optimization of the tree-code memory

The development of increasingly faster RAM memory architectures in the past years is promising. The CPU clock speed is no longer the only parameter that substantially affects the performance of a program. In order to write efficient algorithms, the number of CPU operations has to be minimized and the data need to be suitably stored in memory to be read as fast as possible. Moreover, modern architectures support a huge amount of

cache memory, with orders of magnitude from 1 MB to 10^2 MB. The cache represents a refined and fast-readable level of memory close to the CPU (for an exhaustive essay on cache memory architectures, we refer to Handy 1998). Each time a system needs to manipulate some data, a little chunk of memory in which the relative variables are contained is gathered from the RAM and copied into the cache, from which the processor can operate very quickly. For these reasons, the efficiency of an algorithm is strictly connected to its ability to make several consecutive operations using variables that are stored very close in memory. In this way, the data are loaded once into the cache and the CPU compute directly by minimizing the memory traffic with the RAM. For these purposes, it is straightforward to write efficient tree codes, and more specifically for our purposes, SPH algorithms, if the information related to the particles and cubes is suitably ordered inside the RAM. Barnes (1986) remarked that the particles should be ordered in memory according to their Cartesian position coordinates. The sorting criterion should accurately follow the same arrangement in which the boxes are mapped inside the RAM. Following this prescription, the closer the particles, the closer the areas of memory in which they are stored, and the closer the information of the related cubes.

B.3. OMP parallelization

By exploiting the OpenMP® libraries designed for Fortran-90, the code can even run with shared memory multi-core CPUs (see Chapman et al. 2007, for a modern treatment of the OpenMP paradigm). We implemented a parallelization for the density evaluation routine and for the acceleration field routine, so that different threads perform calculations on different particles. Given a generic i th point, the algorithm may need to update the quantities \mathbf{a}_i and \dot{u}_i by summing two or more different contributions at the same time. The so-called data-race problem arises: two or more threads are accessed and update the same memory location at the same time, which leads to errors in storing the correct values. To overcome these problems, each thread is provided with a private array to store partial values of acceleration and internal energy rate. After the tree descent, these temporary arrays are summed.