Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 192 (2003) 157–174

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Smooth particle hydrodynamics: importance of correction terms in adaptive resolution algorithms

J.-M. Alimi [a], A. Serna [b,*], C. Pastor [b], G. Bernabeu [c]

[a] *Laboratoire de l'Univers et de ses Theories (LUTH-CNRS 4862), Observatoire de Paris-Meudon, F92195 Meudon, France*
[b] *Dept. Física y A.C., Universidad Miguel Hernández, E03202 Elche, Spain*
[c] *Dept. Física e Ingeniería de Sistemas, Universidad de Alicante, E03080 Alicante, Spain*

Received 21 March 2002; received in revised form 13 June 2003; accepted 13 June 2003

Abstract

We describe TREEASPH, a new code to evolve self-gravitating fluids, both with and without a collisionless component. In TREEASPH, gravitational forces are computed from a hierarchical tree algorithm (TREEcode), while hydrodynamic properties are computed by using a SPH method that includes the ∇h correction terms appearing when the spatial resolution $h(t, r)$ is not a constant. Another important feature, which considerably increases the code efficiency on sequential and vectorial computers, is that time-stepping is performed from a PEC scheme (Predict–Evaluate–Correct) modified to allow for individual timesteps. Some authors have previously noted that the ∇h correction terms are needed to avoid the introduction on simulations of a non-physical entropy. By using TREEASPH we show here that, in cosmological simulations, this non-physical entropy has a negative sign. As a consequence, when the ∇h terms are neglected, the density peaks associated to shock fronts are overestimated. This in turn results in an overestimated efficiency of star-formation processes.

Keywords: Methods: numerical; Hydrodynamics; Galaxies: formation

1. Introduction

The evolution of astrophysical systems is governed, on nearly all scales, by gravitation and hydrodynamics. Planet and star formation is thought to occur in accretion disks well represented as collapsing gas clouds in quasi-hydrostatic equilibrium. The formation and evolution of star clusters is governed by viscous forces and by interactions with the interstellar medium. It is also believed that galaxies and galaxy clusters result from dissipational collapse processes, where the radiative cooling and the viscous heating are essential clues to determine their observed structure.

[*] Corresponding author.
E-mail address: arturo.serna@umh.es (A. Serna).

0021-9991/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0021-9991(03)00351-6

Because of their complexity, the study of these problems generally requires computer simulations based on either Lagrangian or Eulerian hydrodynamic methods. Most of the Lagrangian methods used in astrophysics are based on the SPH (smooth particle hydrodynamics) technique proposed by Lucy [27] and Gingold and Monaghan [14]. In SPH, fluid elements constituting the system are sampled and represented by particles, and the dynamical equations are obtained from the Lagrangian form of the hydrodynamic conservation laws. The SPH method allows for an easy implementation of adaptive resolution scales and, therefore, it is well suited to efficiently simulate systems with regions of very different densities. Other physical processes, such as star formation, feedback, chemical evolution, etc., are also easily implemented in SPH. Eulerian methods are instead based on the so-called Godunov algorithm [16], where shock capturing is comparatively more accurate than in SPH. Their new formulations, using adaptive mesh refinements [5,26,35], are particularly well suited for astrophysical problems. For a comparison of the performances of different hydrodynamical codes of both kinds see, e.g. [13,22].

A rigorous introduction of variable resolution scales in SPH requires that additional terms must be included in the particle equations of motion. These terms, which account for the variability of $h(r,t)$, are usually termed as "the ∇h terms". Most SPH codes neglect such additional terms because, in simulations without large density gradients, they seem to have a negligible effect on the global dynamics of systems [12,15]. However, some authors [20,34,37,40] have noted that, if the ∇h terms are neglected, the particle equations of motion are no longer conservative and the SPH method introduces a non-physical entropy the effect of which is not clear. In principle, this problem is not exclusive of the SPH method but it would also appear in any other fluid algorithm with a variable spatial resolution.

Since SPH simulations have become an essential tool to study numerous problems in different fields of physics, it is very important to control any numerical artifact which could alter their results. In particular, it is necessary to analyze whether the ∇h terms can be ignored to economize computing time or, on the contrary, whether they are necessary to obtain reliable results. In this paper, we will describe a new code, TREEASPH, where gravitational forces are computed from a hierarchical tree algorithm (TREEcode), while hydrodynamic properties are computed with a SPH method. By using this code, we will also analyze the possible influence of the ∇h terms on final results.

The plan of the paper is as follows. After summarizing in Section 2 the SPH and treecode methods, we describe in Section 3 the basic aspects of our code. The influence of the ∇h terms on SPH simulations is analyzed in Section 4. The main conclusions obtained from this analysis are finally presented in Section 5.

2. Overview of the SPH and treecode methods

2.1. The smoothed particle hydrodynamics (SPH) technique

As we have mentioned in Section 1, the SPH method consists of representing the fluid elements by N_{g} gas particles which act as interpolation centers to determine the hydrodynamic properties (such as density, pressure gradients, etc.) at the position $\mathbf{r}_i$ of each particle i. This interpolation is performed through a kernel [1] function $W(|\mathbf{r}_i - \mathbf{r}_j|, h)$ giving different weights to the information provided by each particle j. The smoothing length h_i specifies the extension of the interpolation volume associated to particle i. It is commonly computed through the requirement that a sphere of radius $2h_i$, centered on particle i, contains a fix number N_{S} (typically $N_{\mathrm{S}} = 32$) of particles or neighbors.

[1] We will adopt throughout this paper a kernel with compact support such as the spline function proposed by Monaghan and Lattanzio [31].

The smoothed estimate of the density at $\mathbf{r}_i$ is given in SPH by:

$$\rho(\mathbf{r}_i) = \sum_{j=1}^{N_g} m_j W(r_{ij}, h_{ij}), \tag{1}$$

where $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$, and h_{ij} denotes a symmetrized smoothing length,

$$h_{ij} = (h_i + h_j)/2, \tag{2}$$

necessary to avoid a violation of the reciprocity principle [12].

The motion of particle i is then determined by (see, e.g. [3,21,28]):

$$\frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = -\sum_{j=1}^{N_g} m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W(r_{ij}, h_{ij}) - \nabla \Phi_i, \tag{3}$$

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j=1}^{N_g} m_j \left(\frac{P_i}{\rho_i^2} + \frac{\Pi_{ij}}{2} \right) \mathbf{v}_{ij} \nabla_i W(r_{ij}, h_{ij}) - \frac{\Lambda_i}{\rho_i}, \tag{4}$$

where $\mathbf{v}_i = \mathrm{d}\mathbf{r}_i/\mathrm{d}t$, Φ_i is the local gravitational potential, u_i is the specific internal energy, $P_i = (\gamma - 1)\rho_i u_i$ (with $\gamma = 5/3$) is the pressure, Λ_i is a cooling (or heating) function describing non-adiabatic processes other than shocks, [2] and Π_{ij} is the artificial viscosity for which we adopt the standard form proposed by Monaghan and Gingold [30]:

$$\Pi_{ij} = \frac{-\alpha \mu_{ij} \overline{c}_{ij} + \beta \mu_{ij}^2}{\overline{\rho}_{ij}} \quad \text{with } \mu_{ij} = \begin{cases} \frac{\mathbf{v}_{ij}\mathbf{r}_{ij}}{h_{ij}\left(r_{ij}^2/h_{ij}^2 + \eta^2\right)}, & \mathbf{v}_{ij}\mathbf{r}_{ij} < 0, \\ 0, & \mathbf{v}_{ij}\mathbf{r}_{ij} \geqslant 0. \end{cases} \tag{5}$$

Here, α and β are constant parameters of order unity, η^2 is a softening parameter to prevent numerical divergences (typically $\eta^2 = 0.01$), c_i is the local sound speed, $\overline{c}_{ij} = (c_i + c_j)/2$, and finally $\overline{\rho}_{ij} = (\rho_i + \rho_j)/2$.

In the above expressions for $\dot{\mathbf{v}}$ and $\dot{u}$, we have considered that each particle has an individual value of h, that is, we have considered variable smoothing lengths. This fact must be taken into account when the derivative of functions depending on h through the kernel $W(r, h)$ is computed. As a result, Eqs. (3) and (4) will contain not only spatial gradients $\nabla_r W$, but also additional terms containing $\partial W/\partial h$ derivatives. The form of these additional terms, called the ∇h terms, has been given by Nelson and Papaloizou [33] in the case of smoothing lengths symmetrized as in Eq. (2) (see [34] for other symmetrizations). In a compact way, such terms can be written as ([37] hereafter paper I):

$$\begin{aligned} \tilde{\mathbf{a}}_{ij} &= -\frac{m_j}{4}\left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2}\right)\frac{\partial W_{ij}}{\partial h_{ij}}\frac{\mathbf{r}_{ii_m}}{r_{ii_m}} - \delta_{ij_m}\frac{m_j}{4m_i}\frac{\mathbf{r}_{ij}}{r_{ij}}\sum_{k=1}^{N_g} m_k\left(\frac{P_j}{\rho_j^2} + \frac{P_k}{\rho_k^2}\right)\frac{\partial W_{jk}}{\partial h_{jk}}, \\ \dot{\tilde{u}}_{ij} &= \frac{m_j}{4}\frac{P_i}{\rho_i^2}\frac{\partial W_{ij}}{\partial h_{ij}}\left[\frac{\mathbf{v}_{ii_m}\cdot\mathbf{r}_{ii_m}}{r_{ii_m}} + \frac{\mathbf{r}_{jj_m}\cdot\mathbf{v}_{jj_m}}{r_{jj_m}}\right], \end{aligned} \tag{6}$$

where $\tilde{a}_{ij}$ and $\dot{\tilde{u}}_{uj}$ must be added in the sums of Eqs. (3) and (4), respectively, while the subscript k_m denotes the most distant neighbor of particle k. As we will see below (Section 3.2), the identification of k_m does not require additional computations in TREEASPH because this task is performed, anyway, when the smoothing lengths are updated.

[2] We adopt the cooling function of [7,44] for an optically thin primordial mixture of H and He ($X = 0.76$, $Y = 0.24$) in collisional equilibrium and in absence of any significant background radiation field.

2.2. The hierarchical tree method: the TREECODE algorithm

Unlike hydrodynamic interactions, where the properties of a particle are actually determined by the contribution of a modest number of neighbors, the gravitational potential and acceleration of each particle are determined by the interactions with all the remaining particles in the system. Consequently, if gravitational accelerations are computed as the direct sum of all interactions between particles (Particle–Particle method), the computing time on sequential and vectorial computers scales as $\propto N^2$. Simulations with large numbers of particles then become prohibitively expensive in terms of computing time. More efficient procedures to solve the gravitational dynamics are then needed. For example, mesh-based methods [11,12,31,36], or hierarchical tree methods [8,21,39,45].

In TREEASPH, gravitational interactions are computed using the hierarchical tree algorithm (Treecode) introduced by Barnes and Hut [2] and implemented on Fortran by Hernquist [17–19]. Like the PP and SPH methods, the Treecode algorithm is fully Lagrangian and does not use grids which could compromise the spatial resolution and/or impose geometric constraints. Furthermore, the data structure used to manipulate the grouping of particles can be directly applied to certain aspects required in the SPH calculations. The Treecode algorithm relies on a hierarchical subdivision of space into regular cubic cells. The root cell (or node) is a cubic volume which contains all particles in the system. This cell is subdivided into eight cubic cells (in three dimensions) of equal volume, which constitute the immediate descendants of the root node. These cells are subdivided at the next step in smaller unities, and this process continues recursively until each subcell contains either one or zero particles.

The force on a particle is the sum of the (exact) force from nearby particles, plus the force from distant cells approximated as multipoles. These terms are determined by walking through the tree, starting from the highest part of the hierarchy (that is, from the largest volume). At each step, we check to see if the following condition [2] is satisfied

$$\frac{s}{\theta} + \delta < d, \tag{7}$$

where d is the distance from the particle to the cell center of mass, s is the cell size, δ is the distance from the cell center of mass to its geometric center, and θ is a fix tolerance parameter (here, $\theta = 0.4$). If condition Eq. (7) is satisfied, the effect of all particles contained in that cell is computed by approximating the cell as a multipole. Otherwise, the cell is subdivided by continuing the descent through the tree until either the tolerance criterion is satisfied or an elementary cell is reached. In this way, all operations, including the tree construction and evaluation of forces, can be performed in a time proportional to $N \log N$.

Since particles just represent elements of a continuous fluid, their gravitational interaction must be softened by using the techniques described in Section 2.1. To that end, our code uses the same procedure as in [14,21].

3. TREEASPH

3.1. PEC integration scheme with individual timesteps

The simulation of a system constituted by N particles usually requires a computational effort which considerably varies from some regions (or particles) to other. For example, zones of high density and submitted to strong shock waves must be simulated with timesteps much shorter than in the rest of the system. In most of the SPH codes described in literature, all the particles in the system are simultaneously advanced at each timestep. The particle needing the highest time resolution determines the timestep length of all the others. Consequently, a few particles can slow down the simulation of a system.

To make a code more efficient in handling problems with multiple time scales, the computational effort must be centered on those particles that require it, avoiding useless computations for the remaining particles. In other words, it is necessary to allow for different timesteps for each particle. Such a requirement has been implemented in TREEASPH by modifying the usual PEC (Predict–Evaluate–Correct) integration scheme in the following way:

1. We enter the step n (which corresponds to the time t^n) with known positions $\mathbf{r}_i^n$, velocities $\mathbf{v}_i^n$, and accelerations $\mathbf{a}_i^n$, for all the N particles. It is also known, for all the N_{g} gas particles, their smoothing lengths h_i^n, specific internal energies u_i^n, and their derivatives $\dot{u}_i^n$. Furthermore, any integration scheme with individual timesteps needs additional information to identify, at each step, the particles needing a recomputation of their accelerations and their thermal energy derivatives. This information is stored in two vectors t_i^{last} and t_i^{next}. The former contains, for each particle, the time at which the last recomputation of $\mathbf{a}$ and $\dot{u}$ was performed. The second vector, $t_i^{\mathrm{next}} = t_i^{\mathrm{last}} + \Delta t_i$, corresponds to the time at which a recomputation of those variables will be necessary in the future $\mathbf{r}_i^n, \mathbf{v}_i^n, \mathbf{a}_i^n, h_i^n, u_i^n, \dot{u}_i^n, t_i^{\mathrm{next}}, t_i^{\mathrm{last}}$.
2. A list is constructed with those particles j whose value of t^{next} is close to the current time t^n. Such particles are labelled as *active* because they will soon require a complete recomputation of their variables. We then take

$$\Delta t = \min_j (t_j^{\mathrm{next}} - t^n). \tag{8}$$

3. For all particles, active or not, we predict the value of $\mathbf{r}^{n+2}$, $\mathbf{v}^{n+1}$, and u^{n+1} at t^{n+1}

$$\tilde{\mathbf{r}}_i^{n+1} = \mathbf{r}_i^n + \mathbf{v}_i^n \Delta t + \mathbf{a}_a^n (\Delta t)^2/2, \tag{9}$$

$$\tilde{\mathbf{v}}_i^{n+1} = \mathbf{v}_i^n + \mathbf{a}_i^n \Delta t, \tag{10}$$

$$\tilde{u}_i^{n+1} = u_i^n + \dot{u}_i^n \Delta t. \tag{11}$$

4. Only for active particles, we compute $\mathbf{a}_{\mathrm{grav}}^{n+1}$ using $\tilde{\mathbf{r}}^{n+1}$
5. For all gas particles, we compute the hydrodynamic variables h^{n+1}, ρ^{n+1}, and $(P/\rho^2)^{n+1}$ using $\tilde{\mathbf{r}}^{n+1}$ and $\tilde{u}^{n+1}$.
6. Only for active gas particles, we compute $\dot{u}^{n+1}$ and $\mathbf{a}_{\mathrm{hydro}}^{n+1}$. Then, we correct $\mathbf{r}^{n+1}$, $\mathbf{v}^{n+1}$, and u^{n+1}:

$$\begin{aligned} \mathbf{r}_j^{n+1} &= \tilde{\mathbf{r}}_i^{n+1} + A(\mathbf{a}_j^{n+1} - \mathbf{a}_j^n)(\delta t_j)^2/2, \\ \mathbf{v}_j^{n+1} &= \tilde{\mathbf{v}}_i^{n+1} + B(\mathbf{a}_j^{n+1} - \mathbf{a}_j^n)(\delta t_j), \\ u_j^{n+1} &= \tilde{u}_i^{n+1} + C(\dot{u}_j^{n+1} - \dot{u}_j^n)(\delta t_j), \end{aligned} \tag{12}$$

where $B = 1/2$ is required to obtain accurate velocities to second order, while the choice of A and C is somewhat arbitrary. The choice $A = 1/3$ and $C = 1/2$ maintains accuracy to second order both in positions and internal energies [11]. In these expressions, δt_j represents the time interval elapsed from the last evaluation of $\mathbf{a}$ and $\dot{u}$

$$\delta t_j = t^n + \Delta t - t_j^{\mathrm{last}}. \tag{13}$$

Note that, unlike Δt, the δt_j value can be different for each active particle.

7. For all gas particles, we compute the variation by radiative cooling of their specific internal energies by following the integrated scheme of Section 3.4.
8. We update the global time: $t^{n+1} = t^n + \Delta t$.

9. We estimate the individual timestep Δt_j, and update the $t_j^{\text{last}} = t^{n+1}$ and $t_j^{\text{last}} = t^{n+1} + \Delta t_j$ values for each active particle.

In the set of SPH computations outlined in the above PEC scheme, there are some aspects which have not been already discussed and which differ (at least in some points) from our previous PPASPH code: (1) Updating of smoothing lengths, (2) Setting of individual timesteps, (3) Implementation of cooling processes, and (4) Star-formation criteria.

3.2. Search of neighbors and computation of smoothing lengths

Since our code uses a spline kernel with compact support, the SPH properties of particle i are determined only by the contribution of particles j within a distance $2h_{ij} = h_i + h_j$. On the other hand, the individual smoothing lengths h_i are updated at each time step from the requirement that a sphere of radius $2h_i$, centered on particle i, contains a fix number N_{S} of neighbors. Thus, a fundamental requirement of SPH is an efficient algorithm for the neighbor search, both in the case in which the search sphere around particle i has a radius $2h_{ij}$ (which depends on the smoothing lengths of neighbors) and $2h_i$ (independent of h_j).

The first of these tasks is performed in TREEASPH in the following way. When the tree structure is constructed, we save the maximum h_i value of gas particles contained in each cell, $h_{\text{cell}}^{\max}$. Later, in the SPH computations, the particles contributing to the local properties of i (that is, those satisfying $r_{ij} < 2h_{ij}$) are identified as follows. First, we enclose the particle i in a cubic volume of side $2h_i$ (hereafter, the search cube). Then, we descend through the tree as for the force calculation. At each level, we enclose the volume represented by the current node in the tree in a cube of side $s_{\text{cell}} + 2h_{\text{cell}}^{\max}$, where s_{cell} is the cell size. We then check to see if this volume overlaps the search cube. If not, we do not continue the descent down that particular path. Otherwise, the cell is subdivided and we descend down the next level. If the current cell represents a particle, we check to see if it lies within a distance $2h_{ij}$ from particle i, recording it if appropriate. This procedure continues recursively until all paths in the tree are exhausted.

A similar algorithm is used to count the number N_i of neighbors ($r_{ij} < 2h_i$) for particle i but, now, the volume represented by each cell is enclosed in a cube of side $s_{\text{cell}} + 2h_i$ (instead of $s_{\text{cell}} + 2h_{\text{cell}}^{\max}$). Using this N_i value, the procedure to update h_i is then similar to that used by Hernquist and Katz [21]. For each SPH particle, we start with a prediction

$$h_i^{n+1} = h_i^n \frac{1}{2}[1 + (N_{\text{S}}/N_i)^{1/3}] \tag{14}$$

for the smoothing length at time t^{n+1}. Using this prediction, we compute the number of neighbors N_i of particle i. If N_i differs from N_{S} by more than a prescribed tolerance, then h_i^{n+1} is corrected so that the N_i value falls within the allowed range surrounding N_{S}.

3.3. Individual timestep lengths

In order to maintain the numerical integration stability, the timestep of each particle must be modified at each step according to different criteria. A first timestep control concerns the time scale for significant displacements or changes in velocity due to accelerations:

$$\Delta t_i^{\text{a}} = 0.25(h_i^2/a_i^2)^{1/4}, \tag{15}$$

where, for dark-matter particles, the smoothing length h_j is replaced by the softening parameter ϵ_i.

A second limit on Δt_i is usually given by a timestep control which combines the Courant and the viscous conditions:

$$\Delta t_i^{\mathrm{cv}} = \left[\frac{0.2 h_i}{c_i + 1.5(\alpha c_i + \beta \max_j |\mu_{ij}|)} \right]. \tag{16}$$

The timestep of particle i is then given by

$$\Delta t_i = \kappa \min(\Delta t_i^{\mathrm{a}}, \Delta t_i^{\mathrm{cv}}), \tag{17}$$

where the normalization factor κ has been taken equal to 0.5, as required [42] by the integrated cooling algorithm of our code.

3.4. Implementation of cooling

In the presence of cooling processes, an additional constraint should be imposed on the integration timestep: $\Delta t_i < u_i / \dot{u}_i^{\mathrm{cool}}$, where

$$\frac{\mathrm{d}u_i^{\mathrm{cool}}}{\mathrm{d}t} = -\frac{\Lambda(T_i)}{\rho_i}. \tag{18}$$

Such a constraint can be circumvented by implementing the cooling processes in the integrated form proposed and tested by Thomas and Couchman [43] (see also [1]). As a matter of fact, since the Courant condition ensures that densities do not change considerably over one step, Eq. (18) can be integrated to give [43]:

$$\int_{u_i}^{u_i - \Delta u_i^{\mathrm{cool}}} \frac{\mathrm{d}u_i^{\mathrm{cool}}}{\Lambda_i} = -\frac{\Delta t}{\rho_i}. \tag{19}$$

Special treatment is required to prevent an overestimation of cooling processes in a few underresolved zones coincident with the recombination front. To this end, as in [1], the particles submitted to such a nonphysical catastrophic cooling phenomenon were enforced to satisfy a pressure equilibrium condition. To be specific, particles which overcool are selected by using two criteria:

1. The local cooling time ($\Delta t_i^{\mathrm{cool}} = u_i / \dot{u}_i^{\mathrm{cool}}$) is smaller than the viscous-sound crossing time implied by the Courant condition, Δt_i^{cv}:

$$\Delta t_i^{\mathrm{cool}} < \Delta t_{\mathrm{cv}}. \tag{20}$$

2. The local pressure after updating the cooling term of all particles is smaller than half the average pressure $\overline{P}_i$ of the neighbors of particle i:

$$P_i \leqslant \overline{P}_i / 2. \tag{21}$$

When both criteria are satisfied for a particle, we enforce the condition $P_i = \overline{P}_i$. It is important to note that the second of these criteria requires that cooling processes have been updated for all the neighbor particles. Therefore, when individual timesteps are used, this special treatment must be applied only when all the gas particles are simultaneously active. Our code enforces this situation at least once every $\delta t = \max(\Delta t_i^{\mathrm{sim}})$, where $\Delta t_i^{\mathrm{sim}}$ denotes the individual timesteps at the last simultaneous activation of all particles.

3.5. Star formation

3.5.1. Star-formation criteria

Our criteria for star formation are similar to those of [23]. A gas particle is considered to be eligible to form stars if it satisfies the following conditions:

1. The particle is in a collapsing region or, equivalently, the SPH estimate for the velocity divergence is negative:

   $$\nabla \cdot \mathbf{v}_i = -\frac{1}{\rho_i} \sum_{j=1}^{N_g} m_j \mathbf{v}_{ij} \cdot \nabla_i W(r_{ij}, h_{ij}) < 0, \tag{22}$$

   where $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$.
2. It is Jeans-unstable, which implies that pressure forces are unable to halt the collapse. Jeans instability is determined locally by requiring that the sound crossing time be larger than the gravitational dynamical time:

   $$\frac{h_i}{c_i} > \frac{1}{\sqrt{4\pi G \rho_i}}. \tag{23}$$

   Since the softened gravitational force limits the maximum density that a region can obtain, a region could fail the above criterion merely because it has reached such a maximum density and, then, its collapse has stopped. To prevent star formation from being artificially quenched, we do not apply the Jeans criterion in regions where the effects of softened gravity are severe. We consider a particle to be in a softened region if [23]

   $$0.89553 \left(\frac{\pi \epsilon_i}{\sqrt{2}} \right)^2 \frac{\mu m_{\mathrm{H}} G}{k T_i} > 10, \tag{24}$$

   where ϵ_i is the gravitational softening length of the gas particle, T_i is the gas temperature, μ is the mean molecular weight, m_{H} is the mass of hydrogen, and k is Boltzmann's constant.
3. Following [41], we also require that regions of star formation have a minimum physical density. We have chosen the same threshold density as in [24]:

   $$\rho_i > 1.67 \times 10^{-25} \ \mathrm{g/cm^3}, \tag{25}$$

   which is a plausible value on observational grounds [25] and prevents us from overestimating the effects of supernova feedback. In addition, this threshold density is high enough to almost ensure that the gas particle will cool quickly and remain cool as it collapses [32].

Once a gas particle is eligible to form stars, we must decide how and how fast it is turned into stars. We adopt a rate of the form

$$\frac{\mathrm{d}\rho_{\mathrm{g}}}{\mathrm{d}t} = -\frac{\mathrm{d}\rho_*}{\mathrm{d}t} = -\frac{c_* \rho_{\mathrm{g}}}{t_{\mathrm{g}}}, \tag{26}$$

where c_* is a dimensionless star-formation rate parameter, and t_{g} is a characteristic time-scale chosen to be equal to the maximum of the local gas-dynamical time $t_{\mathrm{dyn}} = (4\pi G \rho_{\mathrm{g}})^{-1/2}$, and the local cooling time, $t_{\mathrm{cool}} = u_i/\dot{u}_i$. If the cooling time is shorter than the dynamical time, the region can collapse unimpeded by gas pressure, but if the cooling time is longer than the local dynamical time, the region must wait to cool before it can collapse and make stars. We consider that a region has already cooled if its temperature is smaller than 30,000 K. In that case, we set t_{g} equal to t_{dyn}.

Eq. (26) implies that the probability p that a gas particle forms stars in a time Δt is

$$p = 1 - \mathrm{e}^{-c_* \Delta t / t_{\mathrm{g}}}. \tag{27}$$

Since numerical constraints force us to transform gas into stars in discrete steps, we compute p at each system time step for all eligible gas particles and draw random numbers to decide which particles actually form stars. These particles form then a new star particle of mass $m_i\epsilon_*$, where m_i is the mass of its parent gas particle and $\epsilon_* = 1/3$ is a star-formation efficiency parameter [23]. When this parameter is smaller than unity, the star-formation process for the gas particle i could be quenched, at later times, by the energy injected through supernovae or stellar winds. The new star particle inherits from its parent its position, velocity, and gravitational softening length. We allow each gas particle to form a maximum of four stars and, once a gas particle has formed a star, we reduce its mass in $m_i\epsilon_*$.

3.5.2. Feedback

Star particles essentially behave as collisionless matter. However, unlike dark-matter particles, stars affect the surrounding gas by processes other than purely gravitational interactions. For example, some fraction of the mass and internal energy of stars can return to the gas through stellar winds and supernova explosions. Since our current understanding of these feedback processes is rather poor, their numerical treatment remains the most uncertain aspect of hydrodynamic codes.

For simplicity, we neglect here any effect produced by normal stars and stellar winds and, then, we only consider feedback processes from supernovae. To model these processes, one must take into account that star particles usually have individual masses much larger than those of real stars. Consequently, each star particle represents a group of stars, rather than an individual object.

In order to determine the number and mass fraction of supernovae within a star particle of mass m_i, we approximate the initial mass function (IMF) by a power law: $N(M) = K(M/M_\odot)^{-(1+x)}$, with $x = 1.5$ and lower and upper mass cutoffs of $0.1M_\odot$ and $40M_\odot$, respectively [6,32]. We also assume that every star more massive than $8M_\odot$ ends as a supernova which injects both mass and energy to the surrounding gas.

The mass injection from supernovae is modelled by assuming [23] that each supernova leaves a stellar remnant of $1.4M_\odot$, while its remaining mass is returned as recycled gas to its environment. The adopted IMF implies that a newly formed star particle of mass m_i represents a number of stars given by $N_* = m_i/\overline{m}_*$ ($\overline{m}_* = 0.29M_\odot$). Since only $\sim 0.13\%$ of such stars ($\sim 6.5\%$ of stellar mass) have masses larger than $8M_\odot$, the particle mass in form of supernovae is $0.065m_i$, what leaves stellar remnants of mass $0.006m_i$. Using an instantaneous recycling approximation, the mass of a new star particle is decreased by an amount of $\Delta m_i^{\mathrm{SN}} = 0.059m_i$, which is added to its parent gas particle [23].

The injection of energy is modelled by assuming that each supernova yields 10^{51} erg. A fraction $(1-f)$ of this energy is added to the surrounding gas in the form of heat (thermal feedback), while a fraction f is deposited as kinetic energy (kinetic feedback). The value of f is a rather uncertain input parameter usually taken as 0 (e.g. [24]) or 0.1 (e.g. [32]).

For the adopted IMF, a new star particle contains 0.0045 supernovae per solar mass. Thus, an energy of 4.5×10^{48} erg for every $M_\odot$ turned into stars is smoothed over the neighboring gas particles. Unlike the injection of mass, this energy is gradually added according to an energy deposition function (EDF) assumed to be exponential [24] with decay time of 4.3×10^7 years, the approximate lifetime of an $8M_\odot$ star. Writing this EDF as a percentage of the total energy distributed, and integrating over Δt, we can compute for each 'active' star particle the energy released by supernovae during the current timestep (ΔE_i^{SN}). The amount of energy received by each particle j near the star particle i is computed by kernel smoothing as [41]:

$$\Delta E_{ij}^{\mathrm{SN}} = \Delta E_i^{\mathrm{SN}} \frac{W(r_{ij}, h_{ij})}{\sum_{\mathrm{neigh}} W(r_{ij}, h_{ij})}, \tag{28}$$

where an amount $(1-f)\Delta E_{ij}^{\mathrm{SN}}$ is used to increase the thermal energy of particle j, while $f\Delta E_{ij}^{\mathrm{SN}}$ goes into kinetic energy as a velocity perturbation directed radially away from the star. This velocity perturbation of neighbor j due to the supernova explosion of particle i is computed as

$$\Delta \mathbf{v}_{ij}^{\mathrm{SN}} = |\Delta \mathbf{v}_{ij}^{\mathrm{SN}}|(\mathbf{r}_{ji}/r_{ji}), \tag{29}$$

where $|\Delta \mathbf{v}_{ij}^{\mathrm{SN}}|$ is obtained from the second-order algebraic equation:

$$f\Delta E_{ij}^{\mathrm{SN}} = m_i\left[(\mathbf{v}_{ji}\cdot\mathbf{r}_{ji})\frac{|\Delta \mathbf{v}_{ij}^{\mathrm{SN}}|}{r_{ij}} + \frac{1}{2}|\Delta \mathbf{v}_{ij}^{\mathrm{SN}}|^2\right], \tag{30}$$

where $\mathbf{v}_{ji}$ being the velocity of neighbor j relative to the star.

We must note that star formation and feedback from supernovae involve several processes which do not conserve the total energy: thermal and kinetic energy added through supernovae, new gravitational interactions between recent star particles and their parent gas particles, and the energy lost through gas being converted into stars. We record at each step the energy changes through all these processes to allow us to calculate energy conservation.

4. Influence of the ∇h terms

4.1. The simulations

As we have mentioned in Section 1, the neglect of the ∇h terms introduces on SPH simulations a non-physical entropy the effect of which is not clear. In this section, we will analyze the possible influence of this entropy. To that end, we have performed different simulations of the collapse of a rotating sphere constituted by gas submitted to radiative cooling and by a dominant amount of dark matter.

Initial conditions were exactly the same in all simulations. That is, we consider $N_{\mathrm{gas}} = N_{\mathrm{DM}} = 1736$ particles distributed on a $\rho(r) \propto r^{-1}$ spherical perturbed grid of initial radius $R_{\mathrm{tot}} = 100$ kpc, and total mass $M_{\mathrm{tot}} = 10^{12} M_{\odot}$. The gas represents 10 per cent of this mass and is initially at uniform temperature $T = 10^3$ K. Particle velocities were assigned by considering the sphere in solid-body rotation around the z-axis with spin parameter $\lambda = J|E|^{1/2}/GM_{\mathrm{tot}}^{5/2} \sim 0.1$ (J and E stand for the total angular momentum and total energy, respectively). Gravitational softening parameters were taken to be 2 and 5 kpc for the gas and dark matter, respectively, and the artificial viscosity parameters were $\alpha = 1$, $\beta = 2$. Units were chosen so that $G = 1$, $[M] = 10^{10} M_{\odot}$, $[L] = 1$ kpc.

In a first simulation, the ∇h terms were ignored as well as star-formation processes. This simulation is useful as a test to compare our results to those found from other codes ([32] and paper I). Fig. 1 shows the time evolution of this test simulation. We see that the system has a first phase, where the gas collapses without pressure support until it becomes centrifugally supported in a thin disk-like structure. The disk is almost completely formed soon after the collapse time ($t \sim 120$) and evolves little thereafter. A spiral-like structure in the gaseous disk starts also to be apparent at nearly the same time and remains during the rest of the simulation.

In a second simulation, where the ∇h terms were taken into account, the final results were not very different from those obtained in the previous simulation. As a matter of fact, the gas density profile has a final central peak which appears to be very slightly less dense than in simulations neglecting the ∇h terms. However, since the [33] formulation introduces some scatter on the density field, such differences are here smaller than the error bars estimated from the local scatter of the individual SPH densities. Therefore, in order to study the influence of the ∇h terms, a different approach must be considered. To that end, Springel and Hernquist (2002) have applied a new formulation for the ∇h terms, which reduces the local scatter, to analyze the global properties of several galaxy-like objects obtained from cosmological simulations. Another possibility consists of considering irreversible processes where any change introduced by the ∇h terms is permanent and adds to other similar changes occurring during the system evolution. A clear example of such processes is star formation, usually associated to collapsing regions and to strong shock fronts. We will consider in this paper the latter approach.

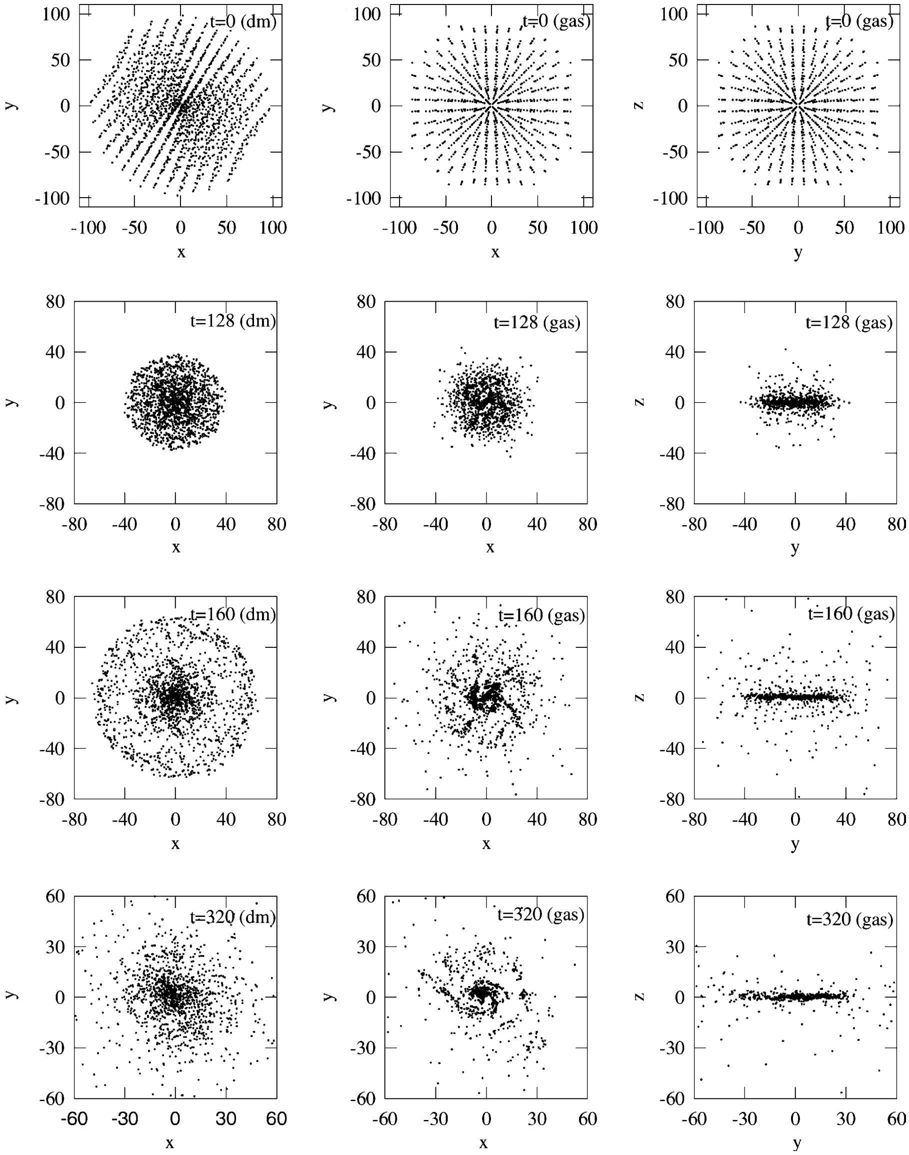

Fig. 1. Time evolution of a two-component rotating sphere without star formation. Units are $G = 1$, $[M] = 10^{10}M_{\odot}$, $[L] = 1$ kpc.

4.2. Star-formation results

We have then performed two other simulations starting from the same initial conditions as before, but now including star-formation and feedback processes (with parameters $c_* = 0.1$, $\epsilon_* = 1/3$, and $f = 0$). In the simulation labelled as NOH, the ∇h terms were switched-off while, in simulation labelled as H, such correction terms were included. Fig. 2 shows the time evolution of the star component in both simulations. We see that the global features of the system evolution are similar to those observed in the previous control

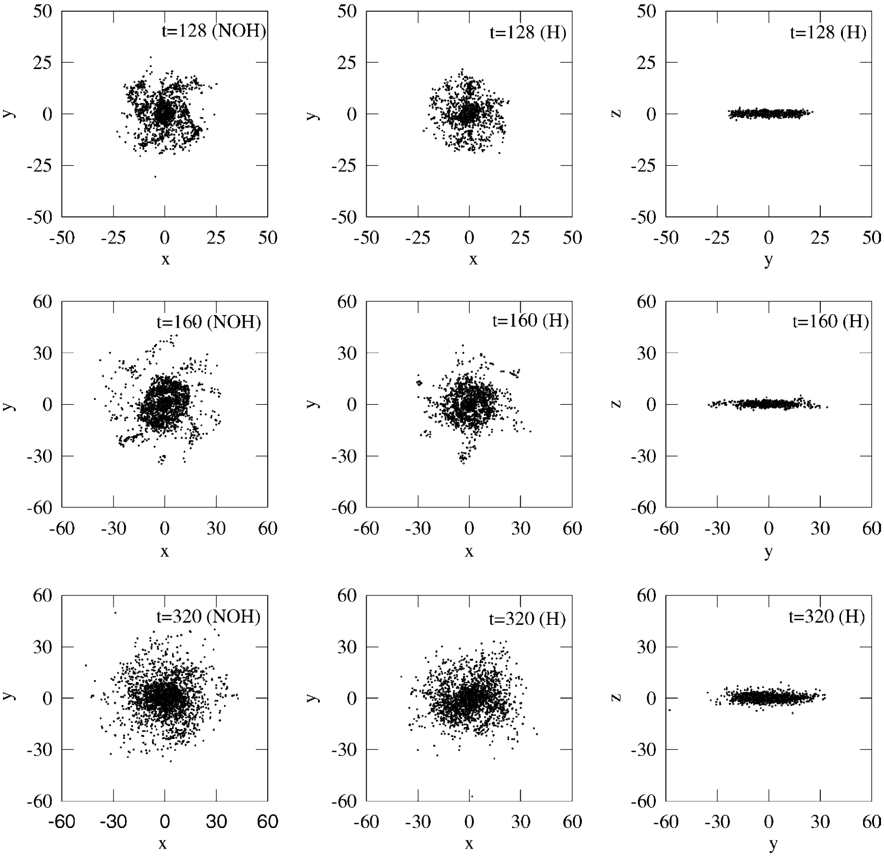

Fig. 2. Time evolution of stars in the simulations H and NOH defined in the text.

simulation: a centrifugally supported stellar disk is formed soon after the collapse time and remains during the rest of both simulations. However, the spiral-like structure is now less evident than in the control simulation (without star formation).

In order to compare quantitatively the star-formation results in both H and NOH simulations, we display in Fig. 3 the total mass of stars, M_*, as a function of time. The solid line represents the M_* evolution when the ∇h terms are ignored (denoted by M_*^{NOH}), while the dotted line corresponds to the case in which such terms were switched-on (denoted by M_*^{H}). We see that, although both simulations give identical results during the initial phase of collapse, predictions on M_* become considerably different after $t \sim 100$. In particular, the M_*^{NOH} value grows faster than M_*^{H} during a time interval around the collapse time until a maximum deviation is reached. Such a deviation remains almost unchanged during the rest of the system evolution and, at the final state, star-formation processes in simulation NOH appear to be overestimated by about 20 per cent (in this specific example) with respect to those found in simulation H. Such a behavior is also observed in Fig. 4, which displays the time evolution of the star-formation rate (SFR). We see from this latter figure that the observed differences between the H and NOH simulations are mainly produced during a wide time interval around $t \sim 100$. Such an overestimation of star-formation processes is higher in the densest regions (see Fig. 5), what can be easily understood because such regions are precisely the main sites of star formation.

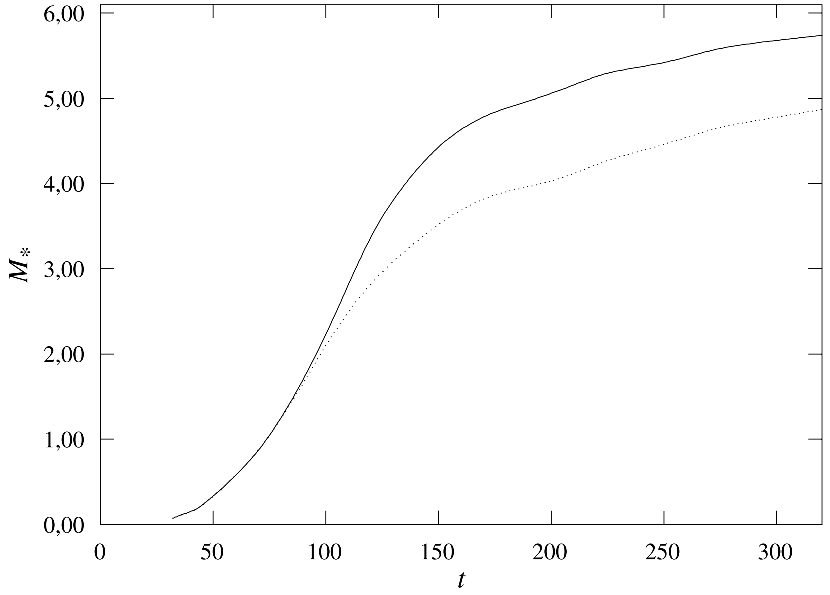

Fig. 3. Total mass of stars as a function of time (code units).

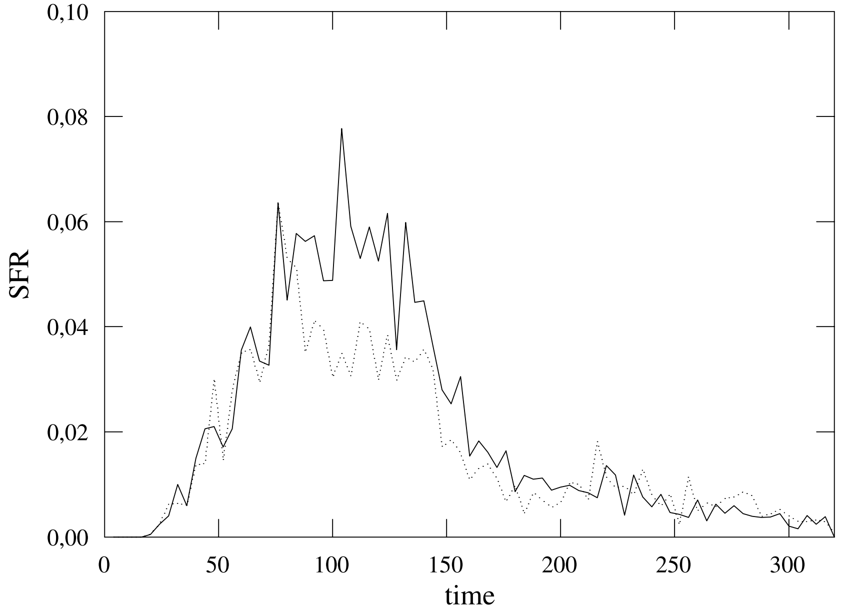

Fig. 4. Star-formation rate (SFR) as a function of time (code units). The solid and dotted lines correspond to the NOH and H simulations, respectively.

Two additional series of simulations have been performed to analyze the possible dependence of our results on the mass and spatial resolution. In series A, we have progressively increased the mass resolution (both for the H and NOH simulations) by considering larger numbers of particles: $N_{\text{gas+DM}} = 3472$, 6944,

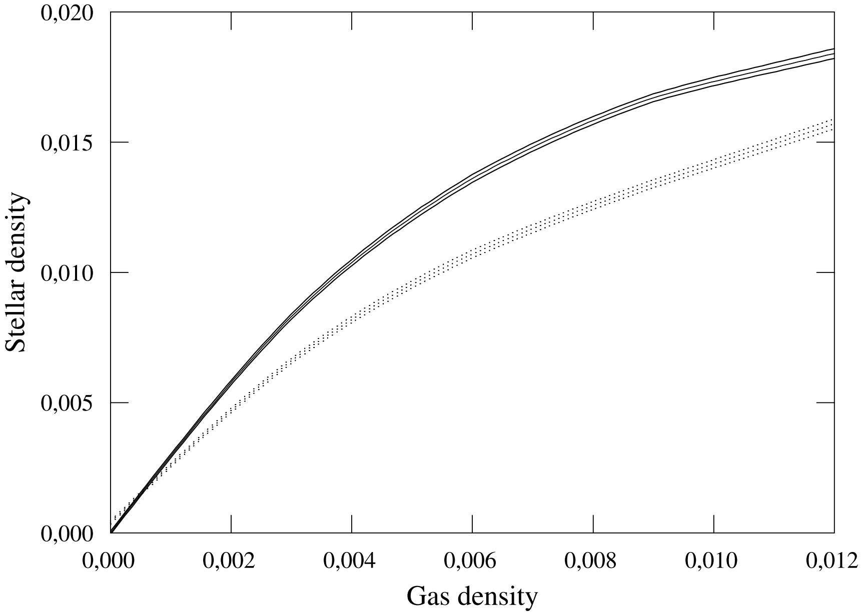

Fig. 5. Density of stars as a function of the gas density at $t = 320$. The 99% confidence region is indicated.

13,888 and 27,776. Typically, the number of the additional star particles at the end of each simulation is about $3N_{\mathrm{gas}}$. In this first series of simulations, the individual smoothing lengths have been updated by imposing a constant number of SPH neighbors (see Section 3.2). In series B, we have instead imposed a minimum value for the smoothing lengths: $h_{\min} = \epsilon_{\mathrm{gas}} = 2$ kpc. In this latter case, all particles with $h = h_{\min}$ have vanishing ∇h terms and, in addition, do not contribute to the ∇h terms of other particles. Fig. 6 shows the final $M_*^{\mathrm{NOH}}/M_*^{\mathrm{H}}$ ratios obtained in both series of simulations. We see that the $M_*^{\mathrm{NOH}}/M_*^{\mathrm{H}}$ ratio exhibits a very slight decrease with $N_{\mathrm{gas+DM}}$ and that, even when the mass resolution is improved by a factor of eight, a difference of about 20% remains in the star-formation results of simulations H and NOH. Furthermore, the results obtained from simulations B imply $M_*^{\mathrm{NOH}}/M_*^{\mathrm{H}}$ ratios which are only slightly smaller than those found in series A. This latter result can be understood from the fact that regions denser than the threshold density for star formation (see Eq. (25)) have a size much larger than ϵ_{gas}. As a matter of fact, we will see later (in Fig. 8) that most regions at $t = 128$ are denser than such a threshold density (0.25×10^{-3} in code units).

4.3. Physical interpretation

In order to understand the physical origin of these different results on star formation, we have analyzed the time evolution of the total entropy both in simulation H and NOH. This evolution is displayed in Fig. 7, which shows that the total entropy is smaller in simulation NOH than in simulation H. Taking into account the fact that simulation H is constructed so that entropy variations have a physical origin, Fig. 7 implies that the non-physical entropy introduced in simulation NOH has a negative sign. This result has an immediate consequence:

Since cooling processes are very efficient in these simulations, the temperature has always an almost constant value, 10^4 K, which corresponds to the effective cut-off of the cooling function. On the other hand, from the definition of the entropy

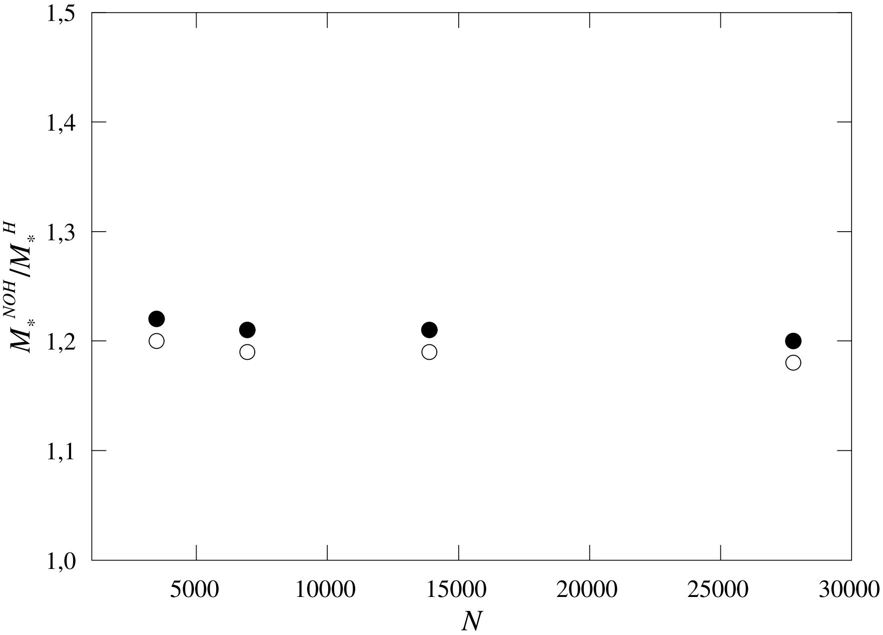

Fig. 6. Ratio of the final stellar mass in simulations NOH to that obtained in simulations H. Solid circles correspond to simulations without any constraint on $h_{\min}$, whereas open circles correspond to simulations with $h_{\min} = \epsilon_{\text{gas}}$.

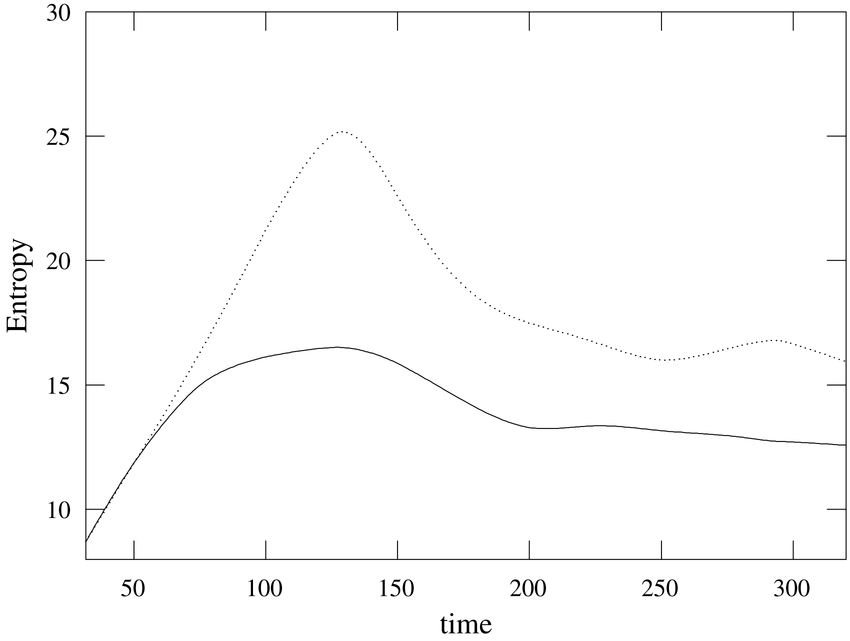

Fig. 7. Time evolution of the total entropy.

$$s = \ln\left(T/\rho^{2/3}\right), \tag{31}$$

it is clear that a smaller entropy implies a larger density. Therefore, those regions with a larger introduction of non-physical entropy (non-relaxed and dense regions as, e.g., shock fronts) will have an over-estimated density. This is clear in Fig. 8, which shows (upper panels) the density profiles at $t = 128$. We see from this figure that both simulations have, at this time, a central density peak and also a secondary peak which corresponds to a strong shock wave traversing the system after the collapse. This latter density peak is considerably larger in simulation NOH than in simulation H, as a consequence of the lack of entropy when the ∇h terms have been neglected.

The above over-estimation of density peaks produces a permanent effect on star-formation processes. As a matter of fact, when we look for the star-formation sites at this time (lower panels in Fig. 8), we find that they are mainly associated to the two density peaks. In the inner core, the star-formation rate in simulation NOH is nearly the same as in simulation H. On the contrary, in regions coincident with the shock front, the higher strength of shocks in simulation NOH results in an enhanced star-formation rate as compared to that found in the H case.

We have also analyzed if such an increase in star formation results could be avoided by choosing smaller values for the artificial viscosity parameters α and β in simulation NOH. In other words, we

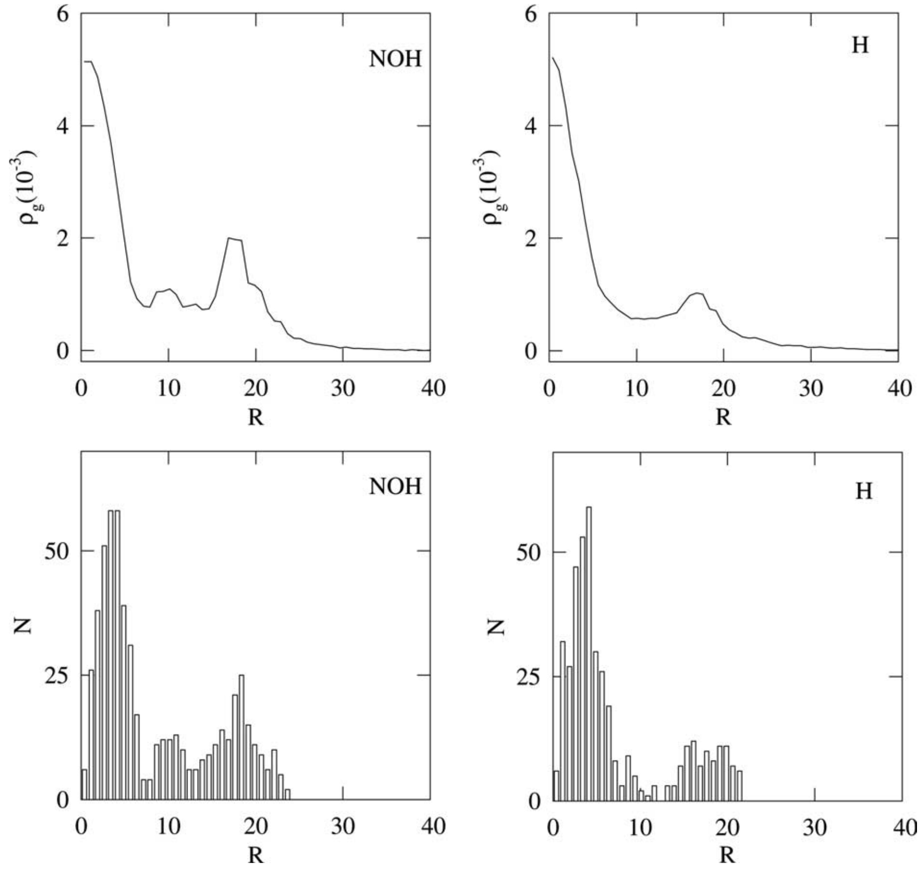

Fig. 8. Gas density (upper panels) and number of eligible particles (lower panels) as a function of radius at $t = 128$.

have analyzed if the viscosity parameters in simulations ignoring the ∇h terms can be 'normalized' so that results coincide with those found when such terms are taken into account. To that end, we have performed a series of simulations neglecting the ∇h terms and starting from the same initial conditions as before, but with progressively smaller α and β parameters. We have however found that a decrease in the α and β parameters results only in a very modest decrease on the final M_* value. As a matter of fact, in the extreme case in which the α and β values are taken very small (e.g., $\alpha = 0.1$ and $\beta = 0.2$) star-formation results are intermediate between the previous H and NOH case, but the remaining system features (gas density profiles, etc.) are considerably degraded because of the very poor simulation of shocks. In the same way, the parameters c_* and ρ_* giving the efficiency of star-formation processes could be calibrated to have the same final mass of stars in both simulations. However, this latter procedure would change the SFR history and, consequently, we would obtain in simulation NOH final stellar populations with ages which are considerably different from those found in simulation H. We then find no way to reconcile the results of both kinds of simulations.

5. Conclusions

We have presented a new code (TREEASPH) for evolving self-gravitating fluids in astrophysics, both with and without a collisionless component. In TREEASPH, gravitational forces are computed with a hierarchical tree algorithm (TREEcode), while hydrodynamic properties are computed using an Adaptive Smoothed Particle Hydrodynamics (ASPH) method. By *adaptive* we mean that our code includes the ∇h correction terms appearing when the spatial resolution $h(t, r)$ is not a constant (in [38] *adaptive* means instead a variable spheroidal kernel). Another important feature, which considerably increases the code efficiency on sequential and vectorial computers, is that time stepping is performed from a PEC scheme (Predict–Evaluate–Correct) modified to allow for individual timesteps.

This code has been used to analyze in detail the possible influence of the ∇h correction terms. To that end, we have simulated the collapse of a rotating sphere both with and without such correction terms. We find that simulations neglecting the ∇h terms introduce a negative entropy. This non-physical entropy is mainly located in dense and non-relaxed regions as, for example, the density peaks associated to strong shock fronts. As a consequence, when the ∇h terms are neglected, these density peaks are over-estimated. Since the strong shock waves are one of the main sites of star formation, the higher strength of shocks in simulations neglecting the ∇h terms results in an enhanced star-formation efficiency. Such a numerical artifact cannot be avoided by just considering smaller values for the viscosity parameters, but only when such correction terms are considered.

Obviously, although we expect that our conclusions will be qualitatively the same in any other situation, the quantitative results depend of each specific problem. In general, situations involving shocks much stronger than those found in this paper will present a much more severe overestimation of star-formation processes. On the other hand, because of the high degree of arbitrariness in the current treatment of these processes, we also expect that differences between simulations using different star formation criteria could be more important than those found when the ∇h terms are neglected. However, since an improvement of such criteria is a matter of current research, we believe very important to control any numerical artifact which could alter the results obtained from simulations.

We would like to conclude by pointing out that SPH is becoming more and more important in many fields of physics and engineering. For example, in industrial problems such as high pressure die casting [9], brittle solids [4], flow through porous media [10], free surface flows [29], etc. Probably, these other fields also contain irreversible processes where the ∇h terms could be important. We then suggest that the ∇h effects on the final results of these other fields should be analyzed in more detail.

Acknowledgements

We are grateful to Lars Hernquist for providing us with his gravitational Treecode program. This work was partially supported by the Generalitat Valenciana (project numbers GC97-CB-13-81 and GV00-139-1), Spain.

References

[1] W.Y. Anninos, M.L. Norman, ApJ 429 (1994) 434.
[2] J. Barnes, P. Hut, Nature 324 (1986) 446.
[3] W. Benz, Numerical Modelling of Stellar Pulsation, Nato Workshop, Les Arcs, France, 1989.
[4] W. Benz, E. Asphaug, Comput. Phys. Commun. 87 (1995) 253.
[5] M.J. Berger, P. Colella, J. Comput. Phys. 82 (1989) 64;
Comput. Phys. Commun. 87 (1989) 253.
[6] J. Binney, S. Tremaine, Galactic Dynamics, Princeton University Press, Princeton, 1987, p. 554.
[7] J.R. Bond, J. Centrella, A.S. Szalay, J.R. Wilson, MNRAS 210 (1984) 515.
[8] S. Borgani, F. Governato, J. Wadsley, N. Menci, P. Tozzi, T. Quinn, J. Stadel, G. Lake, MNRAS 336 (2002) 409.
[9] P.W. Cleary, J. Ha, V. Ahuja, J. Cast. Metals Res. 12 (2000) 335.
[10] P.W. Cleary, J.J. Monaghan, J. Comput. Phys. 148 (1999) 227.
[11] H.M.P. Couchman, P.A. Thomas, F.R. Pearce, ApJ 452 (1995) 797.
[12] A.E. Evrard, MNRAS 235 (1988) 911.
[13] C.S. Frenk, S.D.M. White, P. Bode, et al., ApJ 525 (1999) 554.
[14] R.A. Gingold, J.J. Monaghan, MNRAS 181 (1977) 375.
[15] R.A. Gingold, J.J. Monaghan, J. Comput. Phys. 46 (1982) 429.
[16] S.K. Godunov, Mat. Sbornik 47 (1959) 271.
[17] L. Hernquist, ApJSS 64 (1987) 715.
[18] L. Hernquist, Comput. Phys. Commun. 48 (1988) 107.
[19] L. Hernquist, J. Comput. Phys. 87 (1990) 137.
[20] L. Hernquist, ApJ 404 (1993) 717.
[21] L. Hernquist, N. Katz, ApJSS 70 (1989) 419.
[22] H. Kang, J.P. Ostriker, R. Cen, et al., ApJ 430 (1994) 83.
[23] N. Katz, ApJ 391 (1992) 502.
[24] N. Katz, D.H. Weinberg, L. Hernquist, ApJSS 105 (1996) 19.
[25] R. Kennicutt, ApJ 344 (1989) 685.
[26] R.I. Klein, R.T. Fisher, C.F. McKee, J.K. Truelove, in: S.M. Miyama et al. (Eds.), Proceedings of the International Conference of Numerical Astrophysics, Kluwer Academic Publishers, Boston, 1998, p. 131.
[27] L.B. Lucy, AK 83 (1977) 1013.
[28] J.J. Monaghan, An. Rev. Astron. Astrophys. 30 (1992) 543.
[29] J.J. Monaghan, J. Comput. Phys. 110 (1994) 399.
[30] J.J. Monaghan, R.A. Gingold, J. Comput. Phys. 52 (1983) 374.
[31] J.J. Monaghan, J.C. Lattanzio, A&A 149 (1985) 135.
[32] J.F. Navarro, S.D.M. White, MNRAS 265 (1993) 300.
[33] R.P. Nelson, J.C.B. Papaloizou, MNRAS 264 (1993) 905.
[34] R.P. Nelson, J.C.B. Papaloizou, MNRAS 270 (1994) 1.
[35] M.L. Norman, G.L. Bryan, in: S.M. Miyama et al. (Eds.), Proceedings of the International Conference of Numerical Astrophysics, Kluwer Academic Publishers, Boston, 1998, p. 19.
[36] F.R. Pearce, H.M.P. Couchman, New Astronomy 2 (1997) 411.
[37] A. Serna, J.-M. Alimi, J.-P. Chièze, ApJ 461 (1996) 884.
[38] P.R. Shapiro, H. Martel, J.V. Villumsen, J.M. Owen, ApJSS 103 (1996) 269.
[39] V. Springel, N. Yoshida, S. White, New Astronomy 6 (2001) 79.
[40] V. Springel, L. Hernquist, MNRAS 333 (2002) 649.
[41] F.J. Summers, PhD Thesis, U.C. Berkeley, 1994.
[42] R.J. Thacker et al., MNRAS 319 (2000) 619.
[43] P.A. Thomas, H.M.P. Couchman, MNRAS 257 (1992) 11.
[44] W.H. Tucker, Radiation Processes in Astrophysics, Wiley, New York, 1975.
[45] H. Vedel, U. Hellsten, J. Sommer-Larsen, MNRAS 271 (1999) 743.