Block structured adaptive mesh and time refinement for hybrid, hyperbolic + *N*-body systems

Francesco Miniati [a,b,*], Phillip Colella [c]

[a] *Physics Department, Wolfgang-Pauli-Strasse 16, ETH Zürich, CH-8093 Zürich, Switzerland*
[b] *Max-Planck-Institut für Astrophysik, Karl-Schwarzschild-Str. 1, 85740 Garching, Germany*
[c] *Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA*

Abstract

We present a new numerical algorithm for the solution of coupled collisional and collisionless systems, based on the block structured adaptive mesh and time refinement strategy (AMR). We describe the issues associated with the discretization of the system equations and the synchronization of the numerical solution on the hierarchy of grid levels. We implement a code based on a higher order, conservative and directionally unsplit Godunov's method for hydrodynamics; a symmetric, time centered modified symplectic scheme for collisionless component; and a multilevel, multigrid relaxation algorithm for the elliptic equation coupling the two components. Numerical results that illustrate the accuracy of the code and the relative merit of various implemented schemes are also presented.
© 2007 Published by Elsevier Inc.

1. Introduction

Astrophysical systems are typically complex and highly non-linear, providing the ground for the occurrence, either singly or concomitantly, of numerous physical processes. These include, among others, hydro- and magnetohydro-dynamics, gravity, radiation and many-body interactions. They operate on a wide range of spatial and temporal scales and it is often desirable to fully cover these ranges for a thorough understanding of the problem. Thus the problems are demanding both in terms of physics algorithms and dynamic range (resolution). While the development of high order numerical schemes certainly improves the quality of numerical solutions and the availability of ever more powerful computers has allowed the performance of larger calculations, special techniques are required in order to achieve very large dynamic ranges.

---

[*] Corresponding author. Address: Physics Department, Wolfgang-Pauli-Strasse 16, ETH Zürich, CH-8093 Zürich, Switzerland. Tel.: +41 44 633 6495; fax: +41 44 633 1238.
E-mail address: fm@phys.ethz.ch (F. Miniati).

The adaptive mesh refinement (AMR) technique offers a powerful solution for this purpose [1,2]. While there are difficulties associated with its implementation, the use of AMR in astrophysics and cosmology has grown significantly to include studies of nucleosynthesis in Supernovae explosions [3–6], of the multiphase interstellar medium and radiative shock hydrodynamics [7,8], the problem of star formation out of the collapse of protostellar clouds [9–11] and the formation of the first stars as well as the large scale structure in the universe [12,13]. The use of AMR technique in the above examples has often been instrumental in either revealing new properties of the investigated system (such as instabilities) or pointing out mistaken views based on limited resolution calculations.

We will consider two aspects of application of AMR to astrophysical problems. The first is the extension to incorporate self-gravity. This capability leads to new algorithmic difficulties due to the elliptic nature of the problem. In particular the solution has to be computed simultaneously on all levels of refinement and continuity of both the solution and its normal derivative has to be enforced at coarse–fine level interfaces [14]. However, this introduces non-trivial complications when time refinement is also employed: the coarser levels are advanced first and finer levels are advanced with the assumption that boundary conditions at fine–coarse interfaces are provided by the coarse level solutions interpolated in time. However, the full multilevel elliptic solution can only be computed when all levels are synchronized and thus is not available when the coarser levels are ahead of the finer levels. Thus the first implementations of a full multilevel elliptic solver for self-gravity [10,11,15] do not use refinement in time. And when employing time refinement the multilevel elliptic equation has been solved as a set of independent boundary value problems, one for each level, not a fully multilevel solution [16–19].

The second issue we will address is the application of AMR to hybrid systems, that is a self-gravitating gas coupled to a particle representation of a collisionless matter described by Vlasov–Poisson equations [20]. This is relevant to several problems in astrophysics, particularly for modeling the formation and evolution of structure in the universe. In this case the AMR technique is combined with Particle-Mesh methods to compute the right-hand side to the Poisson's equation due to the particles mass. In order to take advantage of the higher resolution of the finer grids, it is desirable to advance the particles with the force compute on, and the time-steps of, the finest grids that cover their spatial position. This introduces a complication for the elliptic solver similar to the one described above for the case of self-gravitating gas dynamics, in the sense that the particles contribution to the right-hand side of Poisson's equation needs to be accounted for even when the levels of refinement are not synchronized.

Various AMR codes for hybrid systems have been developed over the past several years [21,16–19,15]. The refinement strategy in [16,18,15] is based on splitting of individual cells and in [17] only the collisionless component is evolved. Virtually all schemes use Strang splitting [22] for the multidimensional version of the hydrodynamics and a modified leap-frog method for the integration of the equation of motion of the particles. This has also been done for problems in which there is no coupling to a collisional fluid component, as occurs in computations of collisionless plasmas [23].

In our approach we use the block structured scheme for adaptive mesh and time refinement proposed in [2] as a starting point and extend it include gravity and collisionless particle dynamics. We use an unsplit Godunov's method for hydrodynamics [24]; a symmetric, time centered modified symplectic scheme based on the kick–drift–kick sequence for the collisionless component; and a multilevel, multigrid relaxation algorithm for the elliptic equation coupling the two systems. We introduce two new procedures to solve synchronization issues described above that arise with the elliptic solver when the coarse and finer levels are not synchronized. We use a method analogous to those developed for AMR for incompressible flows [25,26] to compute a lagged estimate of the correction of the elliptic matching conditions at boundaries between refinement levels at times when the levels are not synchronized. We also present a detailed discussion of refinement in time in the presence of collisionless particles, including methods for associating particles with refinement levels, and a particle aggregation operation to cost-effectively estimate the density distribution on coarse levels due to particles evolved on finer levels without compromising the code accuracy. We provide a detailed description of the formal discretization of the system of equations and the issues involved with the synchronization of the numerical solution in the presence of refinement in time, at a level of detail which we feel is lacking in the current literature.

The paper is organized as follows. First, the evolution equations and single level algorithms are outlined in Section 2. In Section 3 we provide a formal definition of the employed AMR volume discretization, variables

and operators and then describe in detail the general AMR algorithm for hybrid system. Finally, in Section 4 we test the accuracy of the code by comparing its results against for set of standard solutions.

2. Evolution equations and temporal discretization

In this section we introduce the system equations and describe their temporal discretization on individual levels of refinement. Motivated by *cosmological* applications, the numerical schemes are formulated for a grid with a time dependent scale length, $a(t)$. Thus after a description of the expanding grid, we introduce the time discretization for the equations of hydrodynamics with gravity and the equations of motion for the collisionless component and then briefly outline the time step constraints and code units.

2.1. Comoving frame

Cosmic expansion is described by the first Friedmann's equation which reads [27]

$$\frac{\dot{a}}{a} = H_0(\Omega_{\mathrm{m}}a^{-3} + \Omega_{\mathrm{k}}a^{-2} + \Omega_{\Lambda})^{1/2} \tag{1}$$

where $a(t)$ is the scale of the universe as a function of time; H_0 measures the current ($a = 1$) rate of cosmic expansion (the Hubble constant); Ω_{m}, Ω_{k} and Ω_{Λ} are parameters representing the current energy density associated to matter, curvature and 'dark' component, respectively, in units of the closure value. The solution to Eq. (1) relating cosmic time and expansion parameter reads

$$H_0 t(a) = \int_0^a \frac{\mathrm{d}\tilde{a}}{\tilde{a}(\Omega_{\mathrm{m}}\tilde{a}^{-3} + \Omega_{\mathrm{k}}\tilde{a}^{-2} + \Omega_{\Lambda})^{1/2}} \tag{2}$$

and admits simple solutions for $\Omega_{\mathrm{k}} = 0$. Since the equations of motion in an expanding Universe are most naturally solved in a *comoving* frame, which expands at the rate $\dot{a}/a$ given by Eq. (1), we operate the change of coordinates

$$\boldsymbol{x} = a(t)^{-1}\boldsymbol{r} \tag{3}$$

where $\boldsymbol{r}$ and $\boldsymbol{x}$ are the coordinates in the laboratory and comoving reference frame, respectively, and transform all differential operators (time derivatives, gradient, laplacian) according to [27]

$$\frac{\partial}{\partial \boldsymbol{r}} \rightarrow \frac{1}{a}\frac{\partial}{\partial \boldsymbol{x}} \tag{4}$$

$$\left(\frac{\partial}{\partial t}\right)_{\boldsymbol{r}} \rightarrow \left(\frac{\partial}{\partial t}\right)_{\boldsymbol{x}} + \frac{\partial \boldsymbol{x}}{\partial t}\left(\frac{\partial}{\partial \boldsymbol{x}}\right)_t = \left(\frac{\partial}{\partial t}\right)_{\boldsymbol{x}} - \frac{\dot{a}\boldsymbol{x}}{a}\left(\frac{\partial}{\partial \boldsymbol{x}}\right)_t. \tag{5}$$

The velocity

$$\dot{\boldsymbol{r}} = \dot{a}\boldsymbol{x} + a\dot{\boldsymbol{x}} \equiv \frac{\dot{a}}{a}\boldsymbol{r} + a\dot{\boldsymbol{x}} \tag{6}$$

is decomposed into a Hubble flow, $(\dot{a}/a)\boldsymbol{r}$, and a peculiar proper component, $u = a\dot{\boldsymbol{x}}$. It is also convenient to introduce the density and pressure in terms of the comoving volume x^3, as opposed to the proper volume r^3,

$$\rho(t, \boldsymbol{x} = \boldsymbol{r}/a) = a^3 \rho_{\mathrm{p}}(t, \boldsymbol{r}) \tag{7}$$

$$P(t, \boldsymbol{x} = \boldsymbol{r}/a) = a^3 P_{\mathrm{p}}(t, \boldsymbol{r}) \tag{8}$$

where the subscript p indicates the proper quantities.

2.2. Hydrodynamics

In comoving coordinates, the hydrodynamics is described by the following set of inhomogeneous partial differential equations

$$\frac{\partial \rho}{\partial t} + \frac{1}{a}\frac{\partial}{\partial x_k}(\rho u_k) = 0 \tag{9}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{1}{a}\frac{\partial}{\partial x_k}(\rho u_i u_k + P\delta ik) = -\frac{\dot{a}}{a}\rho u_i - \frac{1}{a}\rho\frac{\partial \phi}{\partial x_i} \tag{10}$$

$$\frac{\partial \rho e}{\partial t} + \frac{1}{a}\frac{\partial}{\partial x_k}[(\rho e + P)u_k] = -2\frac{\dot{a}}{a}\rho e - \frac{1}{a}\rho u_k\frac{\partial \phi}{\partial x_k} \tag{11}$$

$$\frac{\partial \rho s}{\partial t} + \frac{1}{a}\frac{\partial}{\partial x_k}(\rho s u_k) = -2\frac{\dot{a}}{a}\rho s \tag{12}$$

expressing (from top to bottom) mass, momentum, energy and entropy conservation. Here, ρ and P are the comoving density and pressure, respectively, u is the peculiar proper velocity, ϕ the proper gravitational potential, $e \equiv e_{\mathrm{th}} + e_{\mathrm{k}} = P/\rho(\gamma - 1) + u^2/2$ is the specific total energy, $s = P/\rho^\gamma$ is the specific entropy and γ the gas adiabatic index. The first inhomogeneous terms on the right hand side of Eqs. (10)–(12) describe the effects due to adiabatic cosmic expansion ($\propto \dot{a}/a$). In particular, the factor 2 for the last two equations arises by assuming that the internal energy of the gas is solely associated with translational degrees of freedom[1]. Finally, gravity is described by the gravitational potential ϕ, which is generated by the matter distribution of both the collisional and the collisionless components.

We use a cell-centered discretization for our primary dependent variables: $U(\boldsymbol{x}, t) \equiv (\rho, \rho\mathbf{u}, \rho e)^{\mathrm{T}} \to U_i^n$, where $\boldsymbol{i} \in \mathbb{Z}^{\mathbf{D}}$ indexes grid points in a space with $\mathbf{D}$ dimensions and n is a discrete time index. The starting point for our temporal discretization is a conservative finite-difference method for the hydrodynamic equations:

$$U^{n+1,h} = U^n - \Delta t \mathcal{A}(t)\left(D \cdot \vec{F}^{n+\frac{1}{2}}\right) \tag{13}$$

where $D \cdot \vec{F}^{n+\frac{1}{2}}$ approximates the spatial derivative terms on the left-hand side of (11), $\mathcal{A}(t)$ is a diagonal matrix with elements $\left[\frac{1}{a^{n+1/2}}, \frac{1}{a^{n+1}}, \frac{a^{n+1/2}}{(a^{n+1})^2}, \frac{a^{n+1/2}}{(a^{n+1})^2}\right]$, and Δt is the time step. We use an unsplit second-order Godunov's method [28,29,24] to compute the flux divergence. In addition to the cosmological expansion terms, the gas and particle components couple through the force field which is solution to the following Poisson's equation

$$\Delta \phi = \frac{4\pi G}{a}(\rho_m - \langle \rho_m \rangle). \tag{14}$$

Here $\rho_{\mathrm{m}} = \rho_{\mathrm{gas}} + \rho_{\mathrm{part}}$ is the total comoving mass density; the particle density, ρ_{part}, is computed through a Particle Mesh method (see details in the Appendix A). When periodic boundary conditions are used, $\langle \rho_{\mathrm{m}} \rangle$ is the volume average, otherwise it is zero. The details of the spatial discretization for Poisson's equation are given in the following section. For now, we assume that we can compute the gravitational acceleration at cell centers, $\boldsymbol{f_i} \approx -\nabla \phi$, with second order accuracy. To compute the effect of the source terms, we need to compute them before and after the hydrodynamic update, taking advantage of the fact that no sources appear in the density evolution equation:

$$S_i^n = \frac{1}{\Delta t}\begin{pmatrix} 0 \\ \rho_i^{n+1/2}\boldsymbol{f_i^n}\Delta t/a^{n+1} + \rho_i^n\boldsymbol{u_i^n}[(a^n/a^{n+1}) - 1] \\ \Delta(\rho e_{\mathrm{k}}) + P_i^n[(a^n/a^{n+1})^2 - 1]/(\gamma - 1) \\ \rho_i^n s_i^n[(a^n/a^{n+1})^2 - 1] \end{pmatrix} \tag{15}$$

where $\rho_i^{n+1/2} = (1/2)(\rho_i^n + \rho_i^{n+1})$, and $\Delta(\rho e_{\mathrm{k}}) = (1/2)[\rho_i^{n+1}(u_i^{n+1})^2 - \rho_i^n(u_i^n)^2]$. After the hydrodynamics update given in Eq. (13) we apply the source terms as follows

$$U^{n+1,h,s_1} = U^{n+1,h} + \Delta t S^n. \tag{16}$$

---

[1] More generally, in D dimensions, the energy losses due to expansion are $\dot{e}_{\mathrm{th}}/e_{\mathrm{th}} = D(\gamma - 1)\dot{a}/a$ for the internal energy and $\dot{e}_{\mathrm{k}}/e_{\mathrm{k}} = 2\dot{a}/a$ for the specific kinetic energy, respectively.

The above estimate of the source term, after being converted to primitive variable form, is also used in the hydrodynamic predictor step in order to obtain fully time-centered fluxes. S^n accounts for the expansion terms to the desired accuracy but is only first order accurate as far as the gravity term is concerned (hence the superscript s_1). After the new gravitational potential has been computed a source correction term is estimated as

$$\delta S_i^{n+\frac{1}{2}} = \frac{1}{2} \rho_i^{n+1} \frac{\delta \boldsymbol{f}_i^{n+1/2}}{a^{n+1}} \begin{pmatrix} 0 \\ 1 \\ \boldsymbol{u}_i^{n+1} + [\Delta t/4a^{n+1}]\delta \boldsymbol{f}_i^{n+1/2} \\ 0 \end{pmatrix} \tag{17}$$

with, $\delta \boldsymbol{f}_i^{n+1/1} = \boldsymbol{f}_i^{n+1} - \boldsymbol{f}_i^n$, and the final source update is

$$U^{n+1} = U^{n+1,h,s_1} + \Delta t \,\delta S^{n+\frac{1}{2}}. \tag{18}$$

2.2.1. Hypersonic flows

Accretion flows induced by gravity are typically hypersonic and can be characterized by very large Mach numbers $\mathcal{M} \geqslant 100$. This situation is common in cosmological simulations [30].

In this case the total energy is largely dominated by the kinetic component, $e_k \sim \mathcal{M}^2 e_{th}$. Since conservative hydro-schemes track the total energy, $e = e_{th} + e_k$, relatively small errors in the partition of the two components can produce spurious values of e_{th}. This is particularly worrisome when a 4-byte (single precision) digital representation of the numerical data is employed. For this reason, we introduce the additional Eq. (12) describing the evolution of the gas entropy. When the Mach number of the bulk flow is very high, $\mathcal{M} \geqslant 50$, and away from shocks, it provides a more accurate solution from the thermal energy than the total energy equation. Eq. (12) is naturally incorporated in the numerical scheme for hydrodynamics adopted here because of its conservative form. In addition, being a simple advection equation, the conservative fluxes for its integration are a byproduct of the Riemann solution and require virtually no extra effort to compute.

The equation for the internal energy could be alternatively integrated [31], but its non-conservative form makes it less attractive. Authors in Ref. [32] already employed the entropy equation in order to improve the accuracy of their Total Variation Diminishing scheme for hypersonic flows, although their implementation required solving twice for the hydrodynamic equations with an extra cost of 30%.

It is worth pointing out that for high Mach number flows, errors in the velocity field are propagated into the Riemann solver solution for the pressure at the cell interface, P^*, with a coefficient $\sim \mathcal{M}$, that is, largely amplified. More precisely we find

$$\frac{\delta P^*}{P^*} = \gamma \frac{\delta \Delta u}{u} \mathcal{M} \tag{19}$$

where Δu is the one dimensional velocity jump at the cell interface. However, as illustrated above, the pressure terms enter the hydrodynamic equations with a weight $\mathcal{M}^{-1}$ as compared to kinetic terms and, therefore, the numerical solution is not degraded in the case of high Mach number flows because of approximations involved in the Riemann solver solution.

2.3. Collisionless component

The collisionless component is described by a set of particles whose evolution in phase space is computed according to

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} = \frac{1}{a}\boldsymbol{u} \tag{20}$$

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = -\frac{\dot{a}}{a}\boldsymbol{u} + \frac{1}{a}\boldsymbol{f} \tag{21}$$

where $\boldsymbol{y}$ and $\boldsymbol{u}$ are the comoving coordinate and peculiar proper velocity, respectively. The acceleration acting on the particle, $\boldsymbol{f}$, is obtained by first computing the acceleration on the grid, using a cell- or face-centered scheme, and then by interpolating it to the particle position through a Particle Mesh method [20]. In order

to advance in time the particle positions and velocities we propose the following integration scheme based on a kick–drift–kick sequence [33]: first the particles velocity and positions are updated as

$$\boldsymbol{u}^{n+1/2} = \boldsymbol{u}^n \frac{a^n}{a^{n+1/2}} + \frac{1}{a^{n+1/2}} \boldsymbol{f}^n(\boldsymbol{y}^n) \frac{\Delta t}{2} \tag{22}$$

$$\boldsymbol{y}^{n+1} = \boldsymbol{y}^n + \frac{1}{a^{n+1/2}} \boldsymbol{u}^{n+1/2} \Delta t. \tag{23}$$

After computing the acceleration at the new timestep, the particle velocity is finally updated as

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{n+1/2} \frac{a^{n+1/2}}{a^{n+1}} + \frac{1}{a^{n+1}} \boldsymbol{f}^{n+1}(\boldsymbol{y}^{n+1}) \frac{\Delta t}{2}. \tag{24}$$

The proposed scheme is *reflexive* and hence symplectic [34]. This has the nice property that the integral of motion will be conserved on average preventing secular accumulation of error and keeping the system about its true trajectory in phase space (see, e.g. discussion in [33]). We have also implemented an alternative method, based on the more common drift–kick–drift sequence, which does the following: first particle positions at half the time step are predicted as

$$\boldsymbol{y}^{n+1/2} = \boldsymbol{y}^n + \frac{1}{a^n} \boldsymbol{u}^n \frac{\Delta t}{2}, \tag{25}$$

then particle positions and velocity are further temporarily updated to

$$\boldsymbol{u}^{n+1,*} = \boldsymbol{u}^n \frac{a^n}{a^{n+1}} + \frac{1}{a^{n+1}} \boldsymbol{f}^n(\boldsymbol{y}^{n+1/2}) \Delta t \tag{26}$$

$$\boldsymbol{y}^{n+1,*} = \boldsymbol{y}^{n+1/2} + \frac{1}{a^{n+1}} \boldsymbol{u}^{n+1,*} \frac{\Delta t}{2}. \tag{27}$$

Based on the gravitational potential at the new time-step, a final correction term is applied, that allows second order accuracy

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{n+1,*} + \frac{1}{a^{n+1}} \left[\boldsymbol{f}^{n+1}(\boldsymbol{y}^{n+1/2}) - \boldsymbol{f}^n(\boldsymbol{y}^{n+1/2}) \right] \frac{\Delta t}{2} \tag{28}$$

$$\boldsymbol{y}^{n+1} = \boldsymbol{y}^{n+1/2} + \frac{1}{a^{n+1}} \boldsymbol{u}^{n+1} \frac{\Delta t}{2}. \tag{29}$$

The above scheme, however, is not fully reflexive. The need of temporary states that approximate the solution at the end of the time step in order to calculate the final correction step, breaks the time symmetry of the scheme.

Note that in both schemes, there is no need for storage of extra information about either the particle positions or their velocities at any old or intermediate step. In addition, the gravitational potential is computed only once per time step making their overall computational cost of rather inexpensive.

It should be noticed, however, that in the case of AMR reflexivity is lost even in the former scheme without additional precautions. This occurs when a particle is transferred to a different level of refinement, or even during a refinement operation. In both cases, the gravitational potential changes as a result of these procedures. Since this change takes place after the correction steps, the backward application of the scheme would not reproduce the initial configuration. Although in principle this could be fixed by storing information about the last timestep and acceleration for each particle, we have not implemented any of this.

2.4. Time step

The time-step is subjected to the following constraints: In accord with the Courant–Friedrichs–Lewy (CFL) condition for stability of finite difference methods [35], we require

$$\Delta t = C_{\text{hydro}} \frac{a(t)\Delta x}{\text{Max}(|u_i| + c_s)} \tag{30}$$

where $C_{\text{hydro}} < 1$ is the CFL number, and u_i and c_s are the fluid velocity in the ith direction and sound speed of the flow, respectively. In presence of a source term, S, we modify the estimate of the fluid velocity according to

$$|u_i| + c_s \rightarrow \frac{|S_i|\Delta x}{[(|u_i| + c_s)^2 + 2|S_i|\Delta x]^{1/2} - (|u_i| + c_s)} \tag{31}$$

where S_i is the component of the source term affecting the velocity u_i. For the purpose of accuracy, rather than stability, for the collisionless particles we likewise require

$$\Delta t = C_{\text{part}} \frac{a(t)\Delta x}{\text{Max}(|u_i|)} \tag{32}$$

with $C_{\text{part}} < 1$ and with U_i corrected as in Eq. (31) but with $c_s = 0$. Finally, we require that the background expansion remains limited during each integration cycle. This allows us to time center the value of $a(t)$ in our integration schemes above and neglect its changes with time. We thus enforce

$$\Delta t < C_{\text{exp}} \frac{a}{\dot{a}} \tag{33}$$

with $C_{\text{exp}} \simeq (1\text{–}2) \times 10^{-2}$.

2.5. Code units

The natural choice for the dimensional units of the above physical equations is given by the following lengths, mass and time scales

$$L_* = L_{\text{box}} \tag{34}$$

$$\rho_* = \rho_c \Omega_m \tag{35}$$

$$t_* = H_0^{-1} \tag{36}$$

where L_{box} is the size of the computational box and $\rho_c \equiv 3H_0^2/8\pi G = 1.879 \times 10^{-29} h^2$ is the critical density of the universe, with $h \equiv H_0/100 \text{ km s}^{-1} \text{ Mpc}^{-1}$. The units for the other quantities are defined in terms of these as

$$u_* = H_0^{-1} L_{\text{box}} \tag{37}$$

$$P_* = \rho_* u_*^2 \tag{38}$$

$$\phi_* = u_*^2 \tag{39}$$

$$T_* = m_{\text{proton}} P_* / k_B \rho_* \tag{40}$$

where m_{proton} is the proton mass and k_B Boltzmann's constant.

3. Adaptive mesh refinement approach

In this section we outline the structure of the AMR algorithm. After introducing the formal notation, we describe in some detail the scheme for hydrodynamics with self-gravity and its modifications when a collisionless component is also included, in the simple case of two levels of refinement. We then describe the extension of the scheme to the general multilevel case.

3.1. Multilevel volume discretization, variables and operators

The underlying discretization of the **D**-dimensional space is given as points $(i_0, \ldots, i_{\mathbf{D}-1}) = \boldsymbol{i} \in \mathbb{Z}^{\mathbf{D}}$. The problem domain is discretized using a grid $\Gamma \subset \mathbb{Z}^{\mathbf{D}}$ that is a bounded subset of the lattice. Γ is used to represent a finite-volume discretization of the continuous spatial domain into a collection of control volumes: $\boldsymbol{i} \in \Gamma$ represents a region of space,

$$V_{\boldsymbol{i}} = [\boldsymbol{i}h, (\boldsymbol{i} + \boldsymbol{u})h] \tag{41}$$

where h is the mesh spacing, and $\boldsymbol{u} \in \mathbb{Z}^{\mathbf{D}}$ is the vector whose components are all equal to one. We can also define face-centered discretizations of space based on those control volumes: $\Gamma^{e^d} = \{\boldsymbol{i} \pm \frac{1}{2}\boldsymbol{e}^d : \boldsymbol{i} \in \Gamma\}$, where $\boldsymbol{e}^d$ is the unit vector in the d direction. Γ^{e^d} is the discrete set that indexes the faces of the cells in Γ whose normals are $\boldsymbol{e}^d$:

$$A_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = [(\boldsymbol{i} + \boldsymbol{e}^d)h, (\boldsymbol{i} + \boldsymbol{u})h], \quad \boldsymbol{i} + \frac{1}{2}\boldsymbol{e}^d \in \Gamma^{e^d}. \tag{42}$$

We define cell-centered discrete variables on Γ:

$$\phi : \Gamma \to \mathbb{R}^m$$

and denote by $\phi_{\boldsymbol{i}} \in \mathbb{R}^m$ the value of ϕ at cell $\boldsymbol{i} \in \Gamma$. We can also define face-centered vector fields on Γ:

$$\vec{F} = (F_0, \ldots, F_{\mathbf{D}-1}), \quad F_d : \Gamma^{e^d} \to \mathbb{R}^m$$

and define a discretized divergence operator on such a vector field:

$$(D \cdot \vec{F})_{\boldsymbol{i}} = \frac{1}{h} \sum_{d=0}^{\mathbf{D}-1} (F_{d,\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - F_{d,\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}), \quad \boldsymbol{i} \in \Gamma. \tag{43}$$

We will find it useful to define a number of operators on points and subsets of $\mathbb{Z}^{\mathbf{D}}$. We define a coarsening operator by: $\mathcal{C}_r : \mathbb{Z}^{\mathbf{D}} \to \mathbb{Z}^{\mathbf{D}}$,

$$\mathcal{C}_r(\boldsymbol{i}) = \left(\left\lfloor \frac{i_0}{r} \right\rfloor, \ldots, \left\lfloor \frac{i_{d-1}}{r} \right\rfloor \right)$$

where r is a positive integer. These operators acting on subsets of $\mathbb{Z}^{\mathbf{D}}$ can be extended in a natural way to the face-centered sets: $\mathcal{C}_r(\Gamma^{e^d}) \equiv (\mathcal{C}_r(\Gamma))^{e^d}$. For any set $\Upsilon \subseteq \Gamma$, we define $\mathcal{G}(\Upsilon, r)$, $r > 0$, to be the set of all points within a $|\cdot|$-distance r of Υ that are still contained in Γ:

$$\mathcal{G}(\Upsilon, r) = \Gamma \cap \bigcup_{|\boldsymbol{i}| \leqslant r} \Upsilon + \boldsymbol{i}$$

where $|\boldsymbol{i}| = \max_{d=0\ldots\mathbf{D}-1}(|i_d|)$. We can extend the definition to the case $r < 0$:

$$\mathcal{G}(\Upsilon, r) = \Gamma - \mathcal{G}(\Gamma - \Upsilon, -r).$$

Thus $\mathcal{G}(\Upsilon, r)$ consists of all of the points in Υ that are within a distance $-r$ from points in the complement of Υ in Γ. In the case that there are periodic boundary conditions in one or more of the coordinate directions, we think of the various sets appearing here and in what follows as consisting of the set combined with all of its periodic images for the purpose of defining set operations and computing ghost cell values. For example, $\mathcal{G}(\Upsilon, r)$ is obtained by growing the union of Υ with its periodic images, and performing the intersections and differences with the union of Γ with its periodic images.

We use a finite-volume discretization of space to represent a nested hierarchy of grids that discretize the same continuous spatial domain. We assume that our problem domain can be discretized by a nested hierarchy of grids $\Gamma^0 \cdots \Gamma^{l_{\max}}$, with $\Gamma^{l+1} = \mathcal{C}_{n_{\text{ref}}^l}^{-1}(\Gamma^l)$, and that the mesh spacings h^l associated with Γ^l satisfy $\frac{h^l}{h^{l+1}} = n_{\text{ref}}^l$. The integer n_{ref}^l is the *refinement ratio* between level l and $l + 1$. These conditions imply that the underlying continuous spatial domains defined by the control volumes are all identical. In this paper we will further assume that n_{ref}^l is even. In the case where there are only two levels, we will refer to them as *coarse* and *fine*, with the notation $\{l = 0, l = 1\} \to \{c, f\}$, and $n_{\text{ref}}^0 \to n_{\text{ref}}$.

AMR calculations are performed on a hierarchy of meshes, $\{\Omega^l\}_{l=0}^{l_{\max}}$ with $\Omega^l \subset \Gamma^l$ and $\Omega^l \supset \mathcal{C}_{n_{\text{ref}}^l}(\Omega^{l+1})$. Typically Ω^l is decomposed into a disjoint union of rectangles to perform calculations efficiently.

We make two assumptions about the nesting of grids at successive levels. We require that the control volume corresponding to a cell in Ω^l is either completely contained in the control volumes defined by Ω^{l-1} or its intersection has zero volume. We also assume that there is at least one layer of level-l cells separating level-$(l+1)$ cells from level-$(l-1)$ cells: $\mathcal{G}(\mathcal{C}_{n_{\text{ref}}^l}(\Omega^{l+1}), 1) \subseteq \Omega^l$. We refer to grid hierarchies that meet these two conditions as *properly nested*.

From a formal numerical analysis standpoint, a solution on an adaptive mesh hierarchy $\{\Omega^l\}_{l=0}^{l_{\max}}$ approximates the exact solution to the Partial Differential Equations only on those cells that are not covered by a grid at a finer level. We define the valid region of Ω^l as

$$\Omega^l_{\text{valid}} = \Omega^l - \mathcal{C}_{n^l_{\text{ref}}}(\Omega^{l+1}).$$

A composite array ψ^{comp} is a collection of discrete values defined on the valid regions at each of the levels of refinement:

$$\psi^{\text{comp}} = \{\psi^{l,\text{comp}}\}^{l_{\max}}_{l=0}, \quad \psi^{l,\text{comp}} : \Omega^l_{\text{valid}} \to \mathbb{R}^m.$$

We can also define valid regions and composite arrays for face-centered variables: $\Omega^{l,e^d}_{\text{valid}} = \Omega^{l,e^d} - \mathcal{C}_{n^l_{\text{ref}}}(\Omega^{l+1,e^d})$. Thus, $\Omega^{l,e^d}_{\text{valid}}$ consists of d-faces that are not covered by the d-faces at the next finer level. A composite vector field $\vec{F}^{\text{comp}} = \{\vec{F}^{l,\text{valid}}\}^{l_{\max}}_{l=0}$ is defined as follows:

$$\vec{F}^{l,\text{comp}} = (F^{l,\text{comp}}_0 \dots F^{l,\text{comp}}_{\mathbf{D}-1}), \quad F^{l,\text{comp}}_d : \Omega^{l,e^d}_{\text{valid}} \to \mathbb{R}$$

Thus a composite vector field has values at level l on all of the faces not covered by faces at the next finer level.

We want to define a composite divergence $D^{\text{comp}}(\vec{F}^{l+1,\text{comp}}, \vec{F}^{l,\text{comp}})_i$ for $i \in \Omega^l_{\text{valid}}$. To do this, we construct an extension of $\vec{F}^{l,\text{comp}}$ to the edges adjacent to Ω^l_{valid} that are covered by fine level faces. On the valid coarse-level d-faces, $F^l_d = F^{l,\text{comp}}_d$. On the faces adjacent to cells in Ω^l_{valid}, but not in $\Omega^{l,e^d}_{\text{valid}}$, we set $F^l_d = \langle F^{l+1,\text{comp}}_d \rangle$, the average of F^{l+1}_d onto the next coarser level:

$$\langle F^{l+1}_d \rangle_{i_l + \frac{1}{2}e^d} = \frac{1}{(n_{\text{ref}})^{\mathbf{D}-1}} \sum_{i+\frac{1}{2}e^d \in \mathcal{F}^d} F^{l+1}_{d,i+\frac{1}{2}e^d}, \quad i_l + \frac{1}{2}e^d \in \zeta^{l+1}_{d,+} \cup \zeta^{l+1}_{d,-}.$$

Here $\mathcal{F}^d$ is the set of all fine level d-faces that are covered by $A_{i+\frac{1}{2}e^d}$. $\zeta^{l+1}_{d,\pm}$ consists of all the d-faces in Ω^l on the boundary of Ω^{l+1}, with valid cells on the low ($\pm = -$) or high ($\pm = +$) side:

$$\zeta^{l+1}_{d,\pm} = \{i \pm \frac{1}{2}e^d : i \pm e^d \in \Omega^l_{\text{valid}}, \quad i \in \mathcal{C}_{n^l_{\text{ref}}}(\Omega^{l+1})\}.$$

Given that extension, our composite divergence is defined as:

$$D^{\text{comp}}(\vec{F}^{l+1,\text{comp}}, \vec{F}^{l,\text{comp}})_i = D \cdot \vec{F}^l_i, \quad i \in \Omega^l_{\text{valid}}. \tag{44}$$

It is useful to express D^{comp} as the application of the level divergence operator D applied to extensions of $\vec{F}^{l,\text{comp}}$ to the entire level, followed by a step that corrects the cells in Ω^l_{valid} that are adjacent to Ω^{l+1}. We define a *flux register* $\delta\vec{F}^{l+1}$ associated with the fine level

$$\delta\vec{F}^{l+1} = (\delta F^{l+1}_0, \dots, \delta F^{l+1}_{\mathbf{D}-1})$$

$$\delta F^{l+1}_d : \zeta^{l+1}_{d,+} \cup \zeta^{l+1}_{d,-} \to \mathbb{R}^m.$$

Let $\vec{F}^l$ be *any* coarse level vector field that extends $\vec{F}^{l,\text{comp}}$, i.e.

$$F^l_d = F^{l,\text{comp}}_d \quad \text{on } \Omega^{l,e^d}_{\text{valid}}.$$

Then, for $i \in \Omega^l_{\text{valid}}$,

$$D^{\text{comp}}(\vec{F}^{l+1,\text{comp}}, \vec{F}^{l,\text{comp}})_i = (D\vec{F}^l)_i + D_{\text{R}}(\delta\vec{F}^{l+1})_i \tag{45}$$

where $\delta\vec{F}^{l+1}$ is a flux register set to be

$$\delta F^{l+1}_d = \langle F^{l+1}_d \rangle - F^l_d \quad \text{on } \zeta^l_{d,+} \cup \zeta^l_{d,-}$$

and D_{R} is the reflux divergence operator, given by the following for valid coarse level cells adjacent to Ω^{l+1}:

$$D_{\text{R}}(\delta\vec{F}^{l+1})_i = \frac{1}{h^l} \sum^{\mathbf{D}-1}_{d=0} \sum_{\substack{\pm=+,-: \\ i\pm\frac{1}{2}e^d \in \zeta^{l+1}_{d,\mp}}} \pm\delta F^{l+1}_{d,i\pm\frac{1}{2}e^d}.$$

For the remaining cells in Ω^l_{valid}, $D_{\text{R}}(\delta\vec{F}^{l+1})$ is defined to be identically zero.

We can use this notation to define the discretizations of Poisson's equation we will be using to compute self-gravity. On a single level, Ω^l, we define Δ^l, the discrete Laplacian, to be the standard $2\mathbf{D} + 1$ point operator, with the values used on ghost cells computed using quadratic interpolation:

$$\Delta^l \phi^l = D \cdot \vec{F}^{l,\phi} \tag{46}$$

$$F_d^{l,\phi} = \frac{\phi_{i+e^d} - \phi_i}{h^l} \tag{47}$$

$$\phi_i^l = \mathcal{I}(\phi^l, \phi^{l-1})_i \quad \text{for } i \in \partial\Omega^l \tag{48}$$

where $\partial\Omega^l \equiv \mathcal{G}(\Omega^l, 1) - \Omega^l$. Here the interpolation function $\mathcal{I}$ is an $O(h^3)$ estimate of the value on the ghost cell obtained from interpolating from values of ϕ^{l-1} on $\Omega_{\text{valid}}^{l-1}$ and from the values of ϕ^l on Ω^l; for details, see [36]. We can then define the composite Laplacian Δ^{comp} applied to all of the valid data on all levels, in terms of that operator and refluxing operations:

$$(\Delta^{\text{comp},l}\phi)_i = (\Delta^l \phi^{l,\text{ext}})_i + D_{\text{R}}(\delta\vec{F}^{l,\phi}) \quad \text{for } i \in \Omega_{\text{valid}}^l \tag{49}$$

$$\delta\vec{F}^{l,\phi} = \langle \vec{F}^{l+1,\phi} \rangle - \vec{F}^{l,\phi} \tag{50}$$

where $\phi^{l,\text{ext}}$ is some extension of $\phi^{l,\text{comp}}$ to all of Ω^l. The resulting operator depends only on the valid values of ϕ in the grid hierarchy (modulo roundoff considerations; cf. Ref. [37] for further details).

3.2. AMR for compressible flows with self-gravity

The starting point for this work is the algorithm described in [2] for solving hyperbolic conservation laws on nested refined grids. For the case of two levels, we assume that the solution on both levels is known at time t^c. The basic steps to evolve the solution on both levels to time $t^c + \Delta t^c$ can be summarized as follows:

1. Update the solution on the coarse grid:

$$U^c(t^c + \Delta t^c) = U^c(t^c) - \Delta t^c(D \cdot \vec{F}^c) + \Delta t^c S(U^c) \quad \text{on } \Omega^c. \tag{51}$$

Here $\mathcal{S}(U^c)$ is computed as in Eq. (15) (with the body force, f, set to zero), and the discrete fluxes $\vec{F}$ are local functions of U^c. We also initialize flux registers associated with Ω^f using the same fluxes

$$\delta\vec{F}^f = -\vec{F}^c.$$

2. Advance the solution from t^f to $t^f + \Delta t^f$ on the fine grid n_{ref} times, $n_{\text{ref}}\Delta t^f = \Delta t^c$:

$$U^f(t^f + \Delta t^f) = U^f(t^f) - \Delta t^f(D \cdot \vec{F}^f) + \Delta t S(U^f) \quad \text{on } \Omega^f$$

$$\delta\vec{F}^f + = \frac{1}{n_{\text{ref}}}\langle \vec{F}^f \rangle \tag{52}$$

$$t^f + = \Delta t^f.$$

Any values required to compute the stencil that are contained in $\Gamma_f - \Omega^f$ are computed by interpolating the coarse grid values $U^c(t^c)$, $U^c(t^c + \Delta t^c)$, using linear interpolation in time, and piecewise linear interpolation in space.

3. Synchronize the values at the old and new times:

$$U^c(t + \Delta t^c) = \langle U^f(t + \Delta t^c) \rangle \quad \text{on } \mathcal{C}_{n_{\text{ref}}}(\Omega^f)$$

$$U^c(t + \Delta t^c) + = D_{\text{R}}(\delta\vec{F}) \tag{53}$$

where $\langle \cdot \rangle_i$ denotes the arithmetic average onto the coarse cell i of all of the values defined on fine grid cells contained in i.

To extend this algorithm to the case of self-gravity, we must solve the Poisson's equation for the gravitational potential due to the mass distribution of the fluid on the coarse and fine levels. As usual the coarse level is advanced first, and the solution at $t^c + \Delta t^c$ is used to provide time interpolated boundary conditions for the fine level at intermediate time steps. In the case of hyperbolic equations the finite characteristic speeds ensure that a fully consistent multilevel solution can be recovered at synchronization time with the refluxing operation. Because of its elliptic nature, however, in order to preserve its multilevel character Poisson's equation

should be solved simultaneously on all levels. Notice that the coupling among levels is enforced by the continuity of the potential (Dirichlet) and of its normal derivative (Neumann) at the coarse/fine grid interface. Therefore, to maintain the multilevel character of Poisson's equation when the levels are not synchronized yet, we obtain a single level solution to Poisson's equation on the coarse level and apply a *lagged* estimate of the effect of the coarse/fine matching conditions at refinement boundaries, following the ideas developed in [25,26,38] for incompressible fluids. This leads to the following modifications to the algorithm given above.

0. At simulation start, when all levels are synchronized, we compute a composite grid solution to Poisson's equation

$$(L^{\mathrm{comp,c}}\phi^{\mathrm{comp}})(t^{\mathrm{c}})_{\boldsymbol{i}} = \rho^{\mathrm{c}}(t^{\mathrm{c}})_{\boldsymbol{i}}, \quad \boldsymbol{i} \in \Omega^{\mathrm{c}}_{\mathrm{valid}} \tag{54}$$

$$(L^{\mathrm{comp,f}}\phi^{\mathrm{comp}})(t^{\mathrm{c}})_{\boldsymbol{i}} = \rho^{\mathrm{f}}(t^{\mathrm{c}})_{\boldsymbol{i}}, \quad \boldsymbol{i} \in \Omega^{\mathrm{f}} \tag{55}$$

as well as the coarse grid solution

$$(L^{\mathrm{c}}\phi^{\mathrm{c}})(t^{\mathrm{c}}) = \rho^{\mathrm{c}}(t^{\mathrm{c}}) \quad \text{on } \Omega_{\mathrm{c}}. \tag{56}$$

Here, and in what follows, we will denote by $L(t) \equiv \frac{a(t)}{4\pi G}\Delta$ with superscripts l, c, f, comp indicating the particular discretization of the Laplacian operator.

1. Together with $U(t^{\mathrm{c}})$, the acceleration $\boldsymbol{f}^{\mathrm{c}}(t)$ derived from the composite solution of the potential on the coarse grid is used to compute the coarse grid fluxes and source terms. After updating the conserved quantities using (51), we compute the coarse grid potential at the new time

$$(L^{\mathrm{c}}\phi^{\mathrm{c}})(t^{\mathrm{c}} + \Delta t^{\mathrm{c}}) = \rho^{\mathrm{c}}(t^{\mathrm{c}} + \Delta t^{\mathrm{c}}) \tag{57}$$

$$\tilde{\phi}^{\mathrm{c,comp}}(t^{\mathrm{c}} + \Delta t^{\mathrm{c}}) = \phi^{\mathrm{c}}(t^{\mathrm{c}} + \Delta t^{\mathrm{c}}) + (\phi^{\mathrm{c,comp}}(t^{\mathrm{c}}) - \phi^{\mathrm{c}}(t^{\mathrm{c}})) \tag{58}$$

where in the latter step we have approximately corrected the coarse grid single-level-solution of the potential for the effects due to the solution on the finer grid [25,26]. We use the solution in Eq. (58) with $\phi^{\mathrm{c,comp}}(t^{\mathrm{c}})$ to obtain boundary conditions interpolated in time for the potential at the fine level at intermediate timesteps.

2. We apply the update (51) on the finer level. At $t^{\mathrm{f}} = t^{\mathrm{c}}$ in order to compute the hydrodynamic fluxes and source terms we use the force, $\boldsymbol{f}^{\mathrm{f}}$, derived from the composite potential solution to Eq. (55). At intermediate steps, $t^{\mathrm{c}} < t^{\mathrm{f}} < t^{\mathrm{c}} + \Delta t^{\mathrm{c}}$, we solve the following Poisson's equation on the fine grid with interpolated boundary conditions

$$\begin{aligned} (L^{\mathrm{f}}\phi^{\mathrm{f}})(t^{\mathrm{f}}) &= \rho^{\mathrm{f}}(t^{\mathrm{f}}) \quad \text{on } \Omega^{\mathrm{f}} \\ \phi^{\mathrm{f}}(t^{\mathrm{f}}) &= \mathcal{I}[\phi^{\mathrm{f}}(t^{\mathrm{f}}), \tilde{\phi}^{\mathrm{c}}(t^{\mathrm{f}})] \quad \text{on } \partial\Omega^{\mathrm{f}} \end{aligned} \tag{59}$$

where $\tilde{\phi}^{\mathrm{c}}(t^{\mathrm{f}})$ is obtained by linear interpolation in time as

$$\tilde{\phi}^{\mathrm{c}}(t^{\mathrm{f}}) = (1 - \alpha)\phi^{\mathrm{c,comp}}(t^{\mathrm{c}}) + \alpha\tilde{\phi}^{\mathrm{c,comp}}(t^{\mathrm{c}} + \Delta t^{\mathrm{c}}), \quad \alpha = \frac{t^{\mathrm{f}} - t^{\mathrm{c}}}{\Delta t^{\mathrm{c}}}.$$

We compute the forces at the new timestep and apply the correction (17) and (18) to the fluid momentum and kinetic energy.

3. At time of synchronization, $t^{\mathrm{f}} = t^{\mathrm{c}} + \Delta t^{\mathrm{c}}$, we solve the set of Eqs. (54)–(56), for the composite and single level solution of the potential. We use the composite solution to derive the new force, $\boldsymbol{f}^{n+1}$, and apply momentum and kinetic energy corrections (17) and (18) to obtain time centered forces on both coarse and fine levels. The flow of the calculation restarts from step 1 with the gravitational force known at all levels.

3.3. AMR with particles

Due to the time refinement character of the AMR technique the solution on different levels is advanced with different timesteps. This implies that the density field represented by the particles evolved on the finer level may not be available on the coarser level unless they are synchronized. However, this is information is necessary to

solve Poisson's equation. Therefore, we find it convenient to introduce *effective* particles to recover such information in a computationally inexpensive way and without compromising the code accuracy.

Thus we introduce an *aggregation* operation $\mathcal{P} \rightarrow \langle \mathcal{P} \rangle^l$ that projects a collection of particles covered by Ω^l onto a set of effective particles, with no more that one particle per cell. If $p \in \langle \mathcal{P} \rangle^l$, then

$$m_p = \sum_{p':\boldsymbol{x}_{p'} \in V_i} m_{p'} \tag{60}$$

$$\boldsymbol{x}_p = \frac{1}{m_p} \sum_{p':\boldsymbol{x}_{p'} \in V_i} m_{p'} \boldsymbol{x}_{p'} \tag{61}$$

$$\boldsymbol{u}_p = \frac{1}{m_p} \sum_{p':\boldsymbol{x}_{p'} \in V_i} m_{p'} \boldsymbol{u}_{p'} \tag{62}$$

The aggregation operation conserves the monopole and dipole terms but causes information to be lost on the quadrupole moment of the aggregated particles [39], which provides corrections to the potential of order h^2. Thus the aggregation step preserves second order accuracy. Note, also, that the potential and force fields obtained through the aggregated particles are only used to provide boundary conditions for the finer level.

Restricting again the discussion to the case of two levels of refinement, the changes in the algorithm described above are given as follows:

0. At simulation start, we partition the particles into ones that will be evolved using the coarse and fine time steps. If $\mathcal{P}$ is the set of all particles,

$$\mathcal{P}^{\mathrm{f}} = \{p \in \mathcal{P} : \boldsymbol{x}_p \in \mathcal{G}(\Omega^{\mathrm{f}}, -n_{\mathrm{ref}} n_{\mathrm{buf}})\} \tag{63}$$

$$\mathcal{P}^{\mathrm{c}} = \mathcal{P} - \mathcal{P}^{\mathrm{f}}. \tag{64}$$

The parameter n_{buf} is chosen so that the support for the Particle-Mesh interpolation function, used to calculate the force acting on the particle and the particle mass distribution on the grid, is completely contained in Ω^{f} for all of the fine grid time steps, $n_{\mathrm{ref}} \Delta t^{\mathrm{f}} = \Delta t^{\mathrm{c}}$. Using $n_{\mathrm{buf}} = 1$ and $C_{\mathrm{part}} = 0.5$ is sufficient for all choices of Particle-Mesh scheme used here (see Appendix A). We then define the set $\langle \mathcal{P}^{\mathrm{f}} \rangle^{\mathrm{c}}$ of fine particles aggregated on the coarse grid using Eqs. (60)–(62). Finally, in computing the gravitational potential the densities in Eqs. (54)–(56) are modified to account for the mass distribution of the particles:

$$\rho_{\boldsymbol{i}}^{\mathrm{c}}(t^{\mathrm{c}}) = \rho_{\boldsymbol{i}}^{\mathrm{c,fluid}}(t^{\mathrm{c}}) + \frac{1}{(h^{\mathrm{c}})^{\mathbf{D}}} \sum_{p \in \mathcal{P}^{\mathrm{c}} \cup \langle \mathcal{P}^{\mathrm{f}} \rangle^{\mathrm{c}}} m_p W\left(\frac{(\boldsymbol{i} + \frac{1}{2}\boldsymbol{u})h^{\mathrm{c}} - \boldsymbol{x}_p(t^{\mathrm{c}})}{h^{\mathrm{c}}}\right) \tag{65}$$

$$\rho_{\boldsymbol{i}}^{\mathrm{f}}(t^{\mathrm{c}}) = \rho_{\boldsymbol{i}}^{\mathrm{f,fluid}}(t^{\mathrm{c}}) + \frac{1}{(h^{\mathrm{f}})^{\mathbf{D}}} \sum_{p \in \mathcal{P}^{\mathrm{f}}} m_p W\left(\frac{(\boldsymbol{i} + \frac{1}{2}\boldsymbol{u})h^{\mathrm{f}} - \boldsymbol{x}_p(t^{\mathrm{c}})}{h^{\mathrm{f}}}\right) + \frac{1}{(h^{\mathrm{c}})^{\mathbf{D}}} \sum_{p \in \mathcal{P}^{\mathrm{c}}} m_p W\left(\frac{(\mathcal{C}_{n_{\mathrm{ref}}}(\boldsymbol{i}) + \frac{1}{2}\boldsymbol{u})h^{\mathrm{c}} - x_p(t^{\mathrm{c}})}{h^{\mathrm{c}}}\right).$$
$$\tag{66}$$

Here W is one of the Particle-Mesh assignment schemes used to spread the particle mass on the grid, described in Appendix A. Note that the addition to ρ^{c} of the density field due to $\langle \mathcal{P}^{\mathrm{f}} \rangle^{\mathrm{c}}$ has no effect on the composite solution, because the support of W for each particle $\in \langle \mathcal{P}^{\mathrm{f}} \rangle^{\mathrm{c}}$ is contained entirely in $\mathcal{C}_{n_{\mathrm{ref}}}(\Omega^{\mathrm{f}})$.

1. The gravitational force resulting from the composite solution of the potential is used to compute both the fluid fluxes and source terms, as well as to perform the update (22) and (23) for the particles in $\mathcal{P}^{\mathrm{c}} \cup \langle \mathcal{P}^{\mathrm{f}} \rangle^{\mathrm{c}}$. For the latter, we interpolate the accelerations from the grid cells to particle positions with one of the methods described in Appendix A.

2. At the end of each fine time step, while $t^{\mathrm{f}} + \Delta t^{\mathrm{f}} < t^{\mathrm{c}} + \Delta t^{\mathrm{c}}$, we compute the new fine potential, ϕ^{f}, modifying the mass density as

$$\rho^{\mathrm{f}}(t^{\mathrm{f}})_{\boldsymbol{i}} = \rho^{\mathrm{f,fluid}}(t^{\mathrm{f}})_{\boldsymbol{i}} + \frac{1}{(h^{\mathrm{f}})^{\mathbf{D}}} \sum_{p \in \mathcal{P}^{\mathrm{f}}} m_p W\left(\frac{(\boldsymbol{i} + \frac{1}{2}\boldsymbol{u})h^{\mathrm{f}} - \boldsymbol{x}_p(t^{\mathrm{f}})}{h^{\mathrm{f}}}\right) + \frac{1}{(h^{\mathrm{c}})^{\mathbf{D}}} \sum_{p \in \mathcal{P}^{\mathrm{c}}} m_p W\left(\frac{(\mathcal{C}_{n_{\mathrm{ref}}}(\boldsymbol{i}) + \frac{1}{2}\boldsymbol{u})h^{\mathrm{c}} - x_p(t^{\mathrm{f}})}{h^{\mathrm{c}}}\right), \boldsymbol{i} \in \Omega^{\mathrm{f}}$$
$$\tag{67}$$

where the positions of the particles in $\mathcal{P}^c$ at the intermediate times are given by linear interpolation between $\boldsymbol{x}_p(t^c)$ and $\boldsymbol{x}_p(t^c + \Delta t^c)$. We then use the acceleration due to this field to update the fine particle velocities using (24), and the fine fluid state using (17) and (18).

3. The synchronization step is analogous to the one in Section 3.2: we calculate a single grid and composite grid solution of the potential using the total mass density distribution of fluid and particles given by Eqs. (65) and (66). The gravitational force derived from the composite potential is used to apply the corrections to the particle velocity, given in Eq. (24), and to the fluid momentum and kinetic energy, given in Eqs. (17) and (18), at all levels. Finally, the sets $\mathcal{P}^c, \mathcal{P}^f$ and $\langle \mathcal{P}^f \rangle^c$ are upgraded following the definitions in (63) and (64) to account for the new particle positions. The flow of the calculation restarts from step 1.

3.4. The general multilevel algorithm

For the case of a general $(l_{\max} + 1)$-level calculation, we assume that at simulation start all the particles have been partitioned into groups corresponding to the levels on which they shall be advanced, and that the particles being advanced by finer grids have been aggregated into effective particles for the next coarser levels. This is summarized in the procedure:

for $l = 0, \ldots, l_{\max} - 1$ **do**
$\quad \mathcal{P}^{l+1} = \{ \boldsymbol{x} \in \mathcal{P}^l : \boldsymbol{x}_p(t^l) \in \mathcal{G}(\Omega^{l+1}, -n_{\mathrm{ref}} n_{\mathrm{buf}}) \}$
$\quad \mathcal{P}^l \leftarrow \mathcal{P}^l - \mathcal{P}^{l+1}$
end for
$\mathcal{P}^{l_{\max}}_c = \emptyset$
for $l = l_{\max} - 1, \ldots, 0$ **do**
$\quad \mathcal{P}^l_c = \langle \mathcal{P}^{l+1} \cup \mathcal{P}^{l+1}_c \rangle^l$
end for

We can then describe the algorithm for **advance**(l) that advances the solution at level l, $0 \leqslant l \leqslant l_{\max}$ by one time step. We assume that the solution is attained at time $t^l + \Delta t^l$ in a process that includes a recursive application of **advance**. At the beginning of **advance**, we assume that: we know the fluid state U^l and particle state P^l associated with that level at time t^l; we know the fluid, particle, and potential at the next coarse level at times t^{l-1}, $t^{l-1} + \Delta t^{l-1}$; a composite solution as well as single level solution of the gravitational potential at time t^l has been computed on level $l-1$ and on all finer levels.

We then advance the solution by a time step Δt^l, with $n_{\mathrm{ref}}^{l-1} \Delta t^l = \Delta t^{l-1}$, as follows:

1. Using $U(t^l)$ along with the accelerations, $\boldsymbol{f}^l_{\boldsymbol{i}}$, $\boldsymbol{i} \in \Omega^l$, computed from ϕ^l, we calculate the hyperbolic fluxes $\vec{F}^l$ and the source terms $S(t^l)$. We update the conserved quantities

$$
\begin{aligned}
&U(t^l + \Delta t^l) = U(t^l) - \Delta t^l D \cdot \vec{F}^l + \Delta t^l S^l(t^l) \\
&\delta F^{l+1} = -\vec{F}^l \quad \text{if } l < l_{\max} \\
&\delta F^l += \langle \vec{F}^l \rangle \quad \text{if } l > 0
\end{aligned}
\tag{68}
$$

and advance the positions and velocities of the particles in $\mathcal{P}^l \cup \mathcal{P}^l_c$ using (22) and (23).

2. With $t^l \leftarrow t^l + \Delta t^l$, we solve Poisson's equation on level l

$$
(L^l \phi^l)(t^l)_{\boldsymbol{i}} = \rho^{\mathrm{fluid},l}(t^l)_{\boldsymbol{i}} + \frac{1}{(h^l)^{\mathbf{D}}} \sum_{p \in \mathcal{P}^l} m_p W\left(\frac{(\boldsymbol{i} + \frac{1}{2}\boldsymbol{u})h^l - \boldsymbol{x}_p(t^l)}{h^l} \right) + \frac{1}{(h^l)^{\mathbf{D}}} \sum_{p \in \mathcal{P}^l_c} m_p W\left(\frac{(\boldsymbol{i} + \frac{1}{2}\boldsymbol{u})h^l - \boldsymbol{x}_p(t^l)}{h^l} \right)
$$
$$
+ \frac{1}{(h^{l-1})^{\mathbf{D}}} \sum_{p \in \mathcal{P}^{l-1}} m_p W\left(\frac{\left(\mathcal{C}^{l-1}_{n_{\mathrm{ref}}}(\boldsymbol{i}) + \frac{1}{2}\boldsymbol{u} \right)h^{l-1} - x_p(t^l)}{h^{l-1}} \right), \ \boldsymbol{i} \in \Omega^l
\tag{69}
$$
$$
\phi^l(t^l)_{\boldsymbol{i}} = \mathcal{I}[\phi^l(t^l), \tilde{\phi}^{l-1}(t^l)]_{\boldsymbol{i}}, \quad \boldsymbol{i} \in \partial\Omega^l
$$

where $\tilde{\phi}^{l-1}(t)$ is obtained by linear interpolation in time between $\tilde{\phi}^{l-1,\mathrm{comp}}(t^{l-1})$ and $\tilde{\phi}^{l-1,\mathrm{comp}}(t^{l-1} + \Delta t^{l-1})$ with the latter defined in Eq. (58). As in Eq. (58) we then estimate

$$\tilde{\phi}^{l,\mathrm{comp}}(t^l) = \phi^l(t^l) + (\phi^{l,\mathrm{comp}}(t^l - \Delta t^l) - \phi^l(t^l - \Delta t^l)) \tag{70}$$

which is used to provide boundary conditions for the potential at the next finer level, $\phi^{l+1}(t)$.

3. We call advance recursively n_{ref}^l times.

$\quad t^{l+1} = t^l$
while $t^{l+1} < t^l + \Delta t^l$ **do**
$\quad$ **advance**$(l+1)$
$\quad t^{l+1} = t^{l+1} + \Delta t^{l+1}$
end while

4. If $t^{l-1} < t^l < t^{l-1} + \Delta t^{l-1}$, we obtain $\{\phi^{l',\mathrm{comp}}(t^l)\}_{l' \geqslant l}$ by solving the composite equations:

$$(L^{\mathrm{comp},l'}\phi^{\mathrm{comp}})(t^l)_{\boldsymbol{i}} = \rho^{\mathrm{fluid},l'}(t^l)_{\boldsymbol{i}} + \frac{1}{(h^{l'})^{\mathbf{D}}} \sum_{p \in \mathcal{P}^{l'}} m_p W\left(\frac{(\boldsymbol{i} + \frac{1}{2}\boldsymbol{u})h^{l'} - x_p(t^l)}{h^{l'}}\right)$$

$$+ \frac{1}{(h^{l'-1})^{\mathbf{D}}} \sum_{p \in \mathcal{P}^{l'-1}} m_p W\left(\frac{\left(\mathcal{C}_{n_{\mathrm{ref}}^{l'-1}}(\boldsymbol{i}) + \frac{1}{2}\boldsymbol{u}\right)h^{l'-1} - x_p(t^l)}{h^{l'-1}}\right), \boldsymbol{i} \in \Omega_{\mathrm{valid}}^{l'}, \quad l' = l, \ldots, l_{\max}$$

$$\tag{71}$$

$$\phi^{l,\mathrm{comp}}(t^l)_{\boldsymbol{i}} = \mathcal{I}[\phi^{l,\mathrm{comp}}(t^l), \tilde{\phi}^{l-1}(t^l)]_{\boldsymbol{i}}, \quad \boldsymbol{i} \in \partial\Omega^l.$$

The field so obtained is used to compute accelerations at the new time, and update the fluid and particle velocities using (17), (18) and (24), respectively.

5. The solution at level l is synchronized with the solutions at the finer levels:

$$U(t^l + \Delta t^l) = \langle U(t^{l+1}) \rangle \quad \text{on } \mathcal{C}_{n_{\mathrm{ref}}^l}(\Omega^{l+1})$$

$$U(t^l + \Delta t^l) + = D_{\mathrm{R}}(\delta \vec{F}^{l+1})$$

We upgrade the sets $\mathcal{P}^{l'}$ for $l' = l - 1, \ldots, l_{\max}$ and $\langle \mathcal{P}^{l'+1} \rangle^{l'}$ for $l' = l - 1, \ldots, l_{\max} - 1$, according to the new particle positions. The flow of the calculation restarts from step 1.

4. Convergence tests

We have implemented the above schemes in a Cosmological Hydromagnetic AMR Radiation Many-body (CHARM) code. The code is based on the CHOMBO AMR library and it is implemented in a hybrid C++/Fortran77 language. Additional physics modules, such as radiation [40], cosmic-rays [41,42] and magnetohydrodynamics will be presented elsewhere. In the following, we focus on numerical tests to assess the performance of the algorithms in terms of accuracy and applicability to problems of direct interest. Performance tests will be presented elsewhere.

Unless explicitly stated otherwise, in the following we use these CFL coefficients for the time step: $C_{\mathrm{hydro}} = C_{\mathrm{part}} = 0.5$ and $C_{\mathrm{exp}} = 0.01$. In addition, we restrict the results to the case of a TSC interpolation scheme which, in accord with previous authors, we find to give the most accurate results. Errors and convergence rates are calculated as follows. At a given resolution, r, for any given cell or particle, $\mathbf{i}$, we estimate the error on a *computed* quantity, $q_r^{\mathrm{c}}(\mathbf{i})$, with respect to the *analytic* solution, $q^a(\mathbf{i})$, as

$$\delta q_r(\mathbf{i}) = q_r^{\mathrm{c}}(\mathbf{i}) - q_r^a(\mathbf{i}). \tag{72}$$

We then compute the n-norm of the error, i.e.

$$L_n(\delta q_r) = \|\delta q_r\|_n = \left[\sum |\delta q_r(\mathbf{i})|^n v_{\mathbf{i}}\right]^{1/n} \tag{73}$$

where $v_{\mathbf{i}}$ is either the $\mathbf{i}$th cell volume or the inverse of the number of particles; finally we estimate the convergence rate as

$$R_n = \frac{\ln[L_n(\delta q_r)/L_n(\delta q_s)]}{\ln(\Delta x_r/\Delta x_s)}. \tag{74}$$

For the cases studied below we report the L_1, L_2 and L_∞ norm of the errors.

4.1. Zel'dovich's pancake

We begin with a classical test problem for cosmological codes, the evolution of a one-dimensional plane-wave perturbation in an expanding background. In Zel'dovich's formulation [43], the comoving position and peculiar velocities of collisionless matter evolve as

$$x(t) = q + \frac{b(t)}{a(t)}p(q) \tag{75}$$

$$v(t) = a(t)\dot{x}(t) \tag{76}$$

where q, p are the Lagrangian initial position and displacements, $a(t)$ is the expansion factor and $b(t)/a(t)$ describes the growth factor of the perturbation. For a closed universe ($\Omega_m = 1$), $a(t) = (3H_0t/2)^{2/3}$ (Eq. (2)) and $b(t) = 2a^2/5$ [43].

Setting the initial displacements to a sinusoidal form, $p(q) = 5A\sin(kq)/2$, where k is the perturbation wavenumber, we obtain

$$x(t) = q + aA\sin(kq) \tag{77}$$

$$v(t) = a\dot{a}A\sin(kq) \tag{78}$$

$$\rho(t) = \rho_0[1 + aAk\cos(kq)]^{-1}. \tag{79}$$

The solution described by Eq. (77) becomes singular, that is $\partial x/\partial q = 0$ when $a_{\text{collapse}} = (Ak)^{-1}$. At this stage, particles trajectories cross at $x = q = \pi/k$ and a caustic forms. In the following we use $a_{\text{start}} = 1/51$, $a_{\text{collapse}} = 1/2$ and $k = 2\pi/h^{-1}L_{\text{box}}$.

4.1.1. Collisionless component

Table 1 demonstrates the second order accuracy of the code. The different columns report, as a function of numerical resolution, the L_1, L_2 and L_∞ norms of the error on the particles position, velocity and force, and the corresponding convergence rates of the errors. The L_2 norm of these errors is also reported graphically in the left hand side panels of Fig. 1. For this case we use equal number of cells and particles, $N_{\text{part}} = N_{\text{cell}}$, a cell-centered force scheme, and a constant $\frac{\Delta t}{\Delta x} = 1.6 \times 10^{-4}$. In Fig. 1 we also report the L_2 norm of the errors for the case in which $N_{\text{part}} = \frac{1}{2}N_{\text{cell}}$ and the particles are initially placed either at cell nodes (central panels, NCP for node centered particle) or cell centers (right panels, CCP for cell centered particle). Both these configurations lead to a slower convergence rate than our reference case illustrated in the left hand side panels.

In cosmological simulations, the timestep during the initial stages is determined by the expansion rate of the background. Thus in Table 2 we report the same quantities as in Table 1 but for the case of a (variable) timestep determined by $\frac{\Delta t}{\Delta x} = C_{\text{exp}}(\frac{a}{\dot{a}})$, with $C_{\text{exp}} = 10^{-2}$. The L_2 norm of these errors are also reported as filled symbols in the left hand side panels of Fig. 2. The errors were computed after ten timesteps so that the system is still in the linear regime. And, in fact, the convergence rates are the same as in Table 1. The same correspondence in terms of convergence rates also exists for the case in which $N_{\text{part}} = \frac{1}{2}N_{\text{cell}}$, as illustrated for the NCP case by the star symbols in right hand side panels of Fig. 2.

Next, we consider the errors during the non-linear regime of the calculation. In particular, we consider the solution just prior to the caustic formation, when the background expanded by a factor 25 since the simulation start. On the left hand side of Table 3 we report the L_1 and L_∞ error norms and convergence rates as in Table 2 while the L-2 errors are shown as open symbols in the left hand side panels of Fig. 2. We see that in the non-linear regime the convergence rates of the errors in the particle positions, velocities and forces have worsened in a minor, appreciable and considerable way, respectively.

Table 1
Convergence tests: collisionless case[a]

N_{part}	L_1	R_1	L_2	R_2	L_∞	R_∞
Position						
8	1.3e − 07	1.9	1.4e − 07	1.9	1.9e − 07	1.8
16	3.5e − 08	2.0	3.9e − 08	2.0	5.4e − 08	2.0
32	8.9e − 09	2.0	9.8e − 09	2.0	1.3e − 08	2.1
64	2.2e − 09	2.0	2.4e − 09	1.9	3.0e − 09	1.6
128	5.6e − 10	–	6.2e − 10	–	9.8e − 10	–
Velocity						
8	1.3e − 04	1.9	1.4e − 04	1.9	1.9e − 04	1.8
16	3.5e − 05	2.0	3.8e − 05	2.0	5.3e − 05	2.0
32	8.8e − 06	2.0	9.7e − 06	2.0	1.3e − 05	2.1
64	2.2e − 06	2.0	2.4e − 06	1.9	3.0e − 06	1.6
128	5.5e − 07	–	6.2e − 07	–	9.7e − 07	–
Force						
8	6.5e − 02	1.9	7.1e − 02	1.9	9.4e − 02	1.8
16	1.7e − 02	2.0	1.9e − 02	2.0	2.7e − 02	2.0
32	4.4e − 03	2.0	4.8e − 03	2.0	6.5e − 03	2.1
64	1.1e − 03	2.0	1.2e − 03	1.9	1.5e − 03	1.6
128	2.8e − 04	–	3.1e − 04	–	4.8e − 04	–

[a] We use equal number of cells and particles, $N_{part} = N_{cell}$, a cell-centered force scheme, and a constant $\frac{\Delta t}{\Delta x} = 1.6 \times 10^{-4}$.
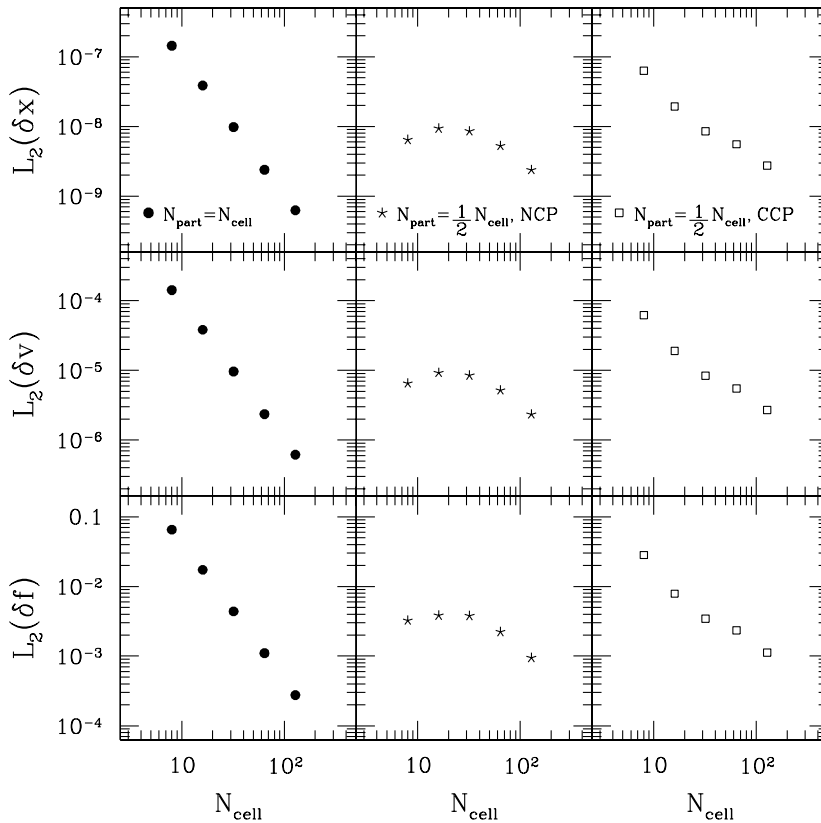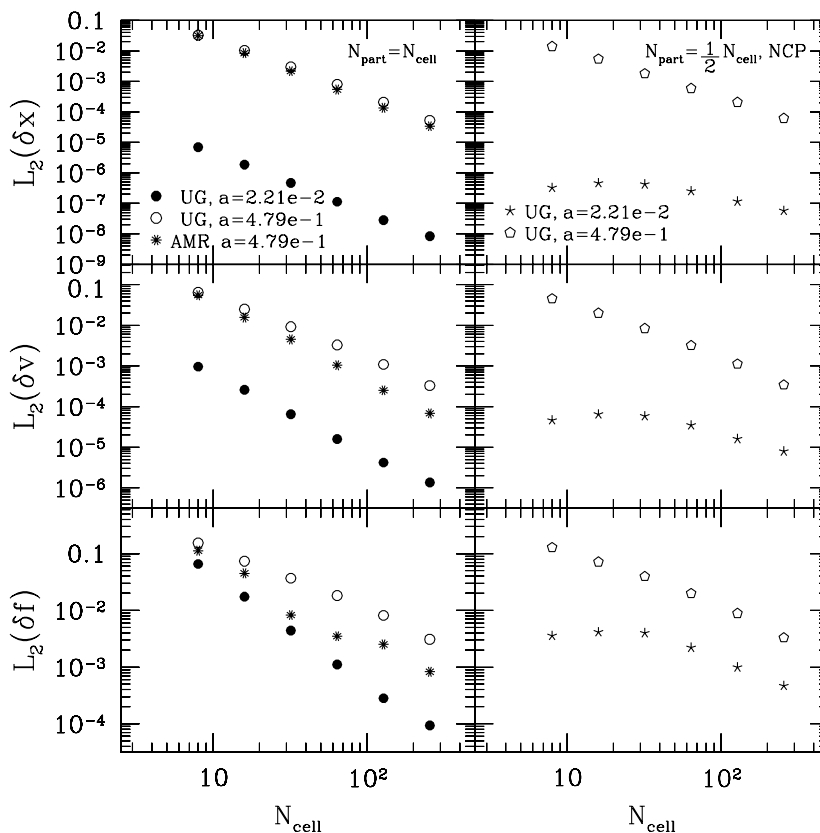

Fig. 1. L_2 norm of the error in position (top), velocity (center) and force (bottom) as a function of the number of grid cells. Left panels correspond to the case in which the number of particles and grid cells is the same, $N_{cell} = N_{part}$. Central and right panels correspond to $N_{part} = \frac{1}{2} N_{cell}$, with particles initially placed either at cell nodes (middle panels) or at cell centers (right panels). In all cases a two point cell centered force stencil is used. See legend for the meaning of the symbols (UG = Uniform Grid, a is the expansion parameter).

Table 2
Convergence tests: collisionless, variable timestep, linear phase[a]

N_{part}	L_1	R_1	L_2	R_2	L_∞	R_∞
Position						
8	6.4e − 06	1.9	6.9e − 06	1.9	9.2e − 06	1.8
16	1.7e − 06	2.0	1.9e − 06	2.0	2.6e − 06	2.0
32	4.3e − 07	2.0	4.7e − 07	2.0	6.4e − 07	2.1
64	1.1e − 07	2.0	1.1e − 07	1.9	1.4e − 07	1.6
128	2.5e − 08	–	2.8e − 08	–	4.5e − 08	–
Velocity						
8	8.9e − 04	1.9	9.6e − 04	1.9	1.3e − 03	1.8
16	2.3e − 04	2.0	2.6e − 04	2.0	3.6e − 04	2.0
32	5.9e − 05	2.0	6.5e − 05	2.0	8.9e − 05	2.1
64	1.5e − 06	2.0	1.6e − 05	1.9	2.0e − 05	1.6
128	3.7e − 07	–	4.2e − 06	–	6.6e − 06	–
Force						
8	6.5e − 02	1.9	7.1e − 02	1.9	9.4e − 02	1.8
16	1.7e − 02	2.0	1.9e − 02	2.0	2.7e − 02	2.0
32	4.4e − 03	2.0	4.8e − 03	2.0	6.5e − 03	2.1
64	1.1e − 03	2.0	1.2e − 03	1.9	1.5e − 03	1.6
128	2.8e − 04	–	3.1e − 04	–	5.1e − 04	–

[a] We use equal number of cells and particles, $N_{\text{part}} = N_{\text{cell}}$, a cell-centered force scheme, a variable timestep, $\frac{\Delta t}{\Delta x} = C_{\exp}\left(\frac{a}{\dot{a}}\right)$, $C_{\exp} = 10^{-2}$, and $a = 0.0221$.

Fig. 2. L_2 norm of the error in position (top), velocity (center) and force (bottom) as a function of the number of grid cells. Left panels correspond to the case in which the number of particles and grid cells is the same, $N_{\text{cell}} = N_{\text{part}}$. Right panels correspond to $N_{\text{part}} = \frac{1}{2}N_{\text{cell}}$, with particles initially placed at cell nodes. See legend for the meaning of the symbols (UG = Uniform Grid, a is the expansion parameter).

Table 3
Convergence tests: collisionless, variable timestep, non-linear phase[a]

N_{part}	Uniform grid				AMR				
	L_1	R_1	L_∞	R_∞	L_1	R_1	L_∞	R_∞	$l_{\max}$
Position									
8	3.0e − 02	1.7	4.6e − 02	1.6	2.9e − 02	1.9	4.5e − 02	1.9	1
16	9.5e − 03	1.8	1.6e − 02	1.8	7.9e − 03	2.0	1.2e − 02	2.0	1
32	2.7e − 03	1.9	4.7e − 03	1.8	2.0e − 03	2.0	3.1e − 03	1.9	2
64	7.2e − 04	2.0	1.3e − 03	2.0	5.1e − 04	2.0	8.3e − 03	1.9	2
128	1.8e − 04	2.0	3.3e − 04	1.8	1.2e − 04	2.0	2.2e − 04	1.3	3
256	4.6e − 05	–	9.2e − 05	–	3.0e − 05	–	9.2e − 05		3
Velocity									
8	5.9e − 02	1.4	9.4e − 02	1.2	5.1e − 02	1.8	7.9e − 02	1.9	1
16	2.2e − 02	1.6	4.1e − 02	1.2	1.4e − 02	1.8	2.1e − 02	1.5	1
32	7.4e − 03	1.6	1.8e − 02	1.3	4.1e − 03	2.1	7.4e − 03	2.2	2
64	2.4e − 03	1.7	7.3e − 03	1.4	9.6e − 03	2.1	1.6e − 03	1.9	2
128	7.4e − 04	1.8	2.8e − 03	1.6	2.2e − 04	2.0	4.5e − 04	1.3	3
256	2.1e − 04	–	9.1e − 04	–	5.7e − 05	–	1.8e − 04	–	3
Force									
8	1.5e − 01	1.1	2.6e − 01	0.7	1.1e − 01	1.3	1.8e − 01	0.9	1
16	7.4e − 02	1.0	1.6e − 01	0.5	4.5e − 02	2.4	9.3e − 02	2.6	1
32	3.7e − 02	1.0	1.2e − 01	0.6	8.2e − 02	1.2	1.6e − 02	0.1	2
64	1.8e − 02	1.2	7.8e − 02	0.7	3.5e − 03	0.5	1.5e − 02	2.3	2
128	8.1e − 03	1.4	4.6e − 02	1.0	2.5e − 03	1.6	3.0e − 02	1.3	3
256	3.1e − 03	–	2.2e − 02	–	8.2e − 04	–	1.2e − 02	–	3

[a] We use equal number of cells and particles, $N_{\text{part}} = N_{\text{cell}}$, a cell-centered force scheme, a variable timestep, $\frac{\Delta t}{\Delta x} = C_{\exp}(\frac{a}{\dot{a}}), C_{\exp} = 10^{-2}$, and $a = 0.479$.

Finally, we test the performance of the AMR code. We use a constant refinement ratio, $n_{\text{ref}} = 2$, and refine cells enclosing a mass larger than 1.5 the average value. A maximum of three levels of refinement were allowed. All runs used a first level of refinement for about 30% of the calculation, except for the lowest resolution case for which the percentage was 12%. The second level of refinement was only used by the three higher resolution runs and only for about 5% of the time. Finally, a third level of refinement was employed only by the two finest runs and only for less than 1% of the time. Similarly, finer grids cover a progressively smaller fraction of the computational domain. The L_1 and L_∞ error norms and convergence rates for the AMR runs are reported on the right hand side of Table 3 while the L-2 errors are shown as spur symbols in the left hand side panels of Fig. 2. These results show that employing AMR during the non-linear evolution improves the convergence rate of the solution in such a way that they resemble the values in the linear stage. This is true in this example for the errors in the position and velocity and to a lesser extent, the force, which is more affected by coarse/fine boundary effects.

Fig. 3 compares the errors in position, velocity and force for a fixed grid (left) and an AMR grid (right) calculations. It focuses on the region where, and the times when, the caustic forms and AMR operates. Thus only one quarter of the total grid is shown (with the cell boundaries of the base grid indicated by vertical lines and marked by integer labels), and the various errors are plotted only for the last one third of the simulation run. At the beginning of that time span (when $a \sim 0.2$), the particles have clustered sufficiently at the grid center and a first level of refinement is created. Six grid meshes are refined (only three between boundaries, 12–16, are shown in Fig. 3). The second level of refinement is created much later and only affects one base (or two refined) grid mesh(es) for the last ten per cent of the simulation. Fig. 3 shows that the errors in position, velocity and force of the particles close to the point where the caustic forms are much reduced when AMR is employed. The generation of a level of refinement is accompanied by a change in the mass distribution and the potential field. When this happens, a particle may experience a sudden change in terms of the force field acting upon it. These effects are responsible for the somewhat 'errant' behavior of the force error, as illustrated in the bottom-left panel of Fig. 3. Overall, though, for each particle the force fluctuations in the AMR case are significantly smaller than the force errors in the uniform grid case.
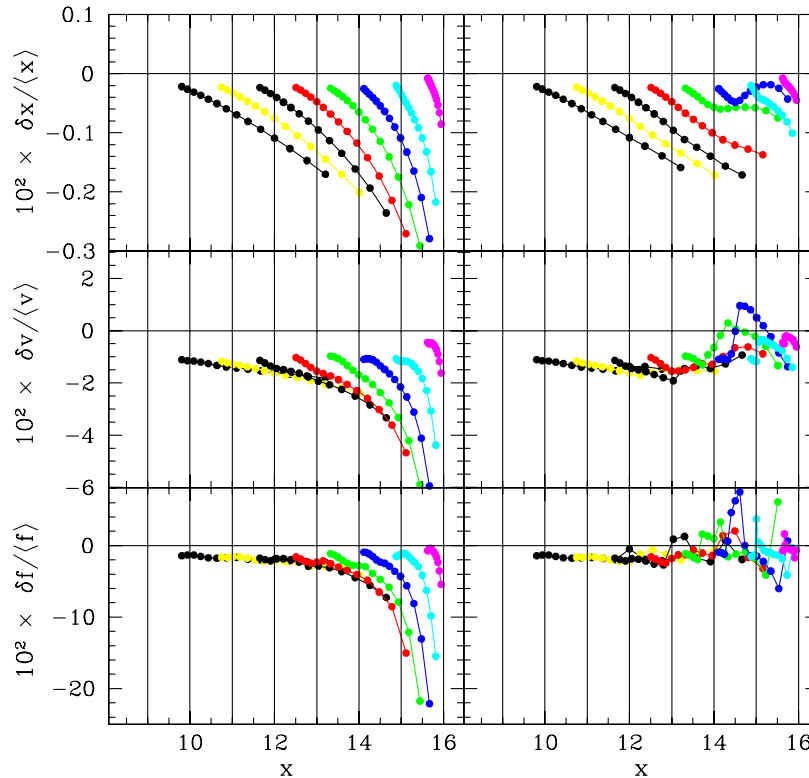
Fig. 3. Comparison between two numerical experiments, employing a uniform grid (left) and two levels of refinement (right), in terms of error in the particle position (top), velocity (middle) and force (bottom). The initial set up includes 32 grid cells (bound by vertical lines) and 32 particles uniformly distributed. Only one quarter of the grid is shown, focusing on the critical region where the caustic forms. Vertical lines indicate cells' boundaries.

As a last result for this section, in Fig. 4 we plot L_2 convergence errors analogous to Fig. 1 but for the case in which the force was computed with a staggered scheme. Comparison of the two figures shows that when $N_{part} = N_{cell}$ the solution obtained with the staggered scheme also converges with second order accuracy, while being characterized by smaller errors. This is in agreement with previous findings [44]. However, when the number of particles is halved ($N_{part} = \frac{1}{2}N_{cell}$) the results obtained with a staggered force scheme worsen more dramatically than for the cell centered case, showing a very poor convergence rate. Closer inspection shows that when the particles are sparse on the grid, oscillations appear in the potential due to the discrete character of the matter distribution as reproduced on the grid. Since this affects only the quality of the staggered scheme the problem may be related to the inconsistency of this scheme with the centering of the stencil used for the discretization of the Laplacian operator. We shall return to this issue in the next test case where the problem reappears with more dramatic effects.

4.1.2. Collisional component

We now turn to the performance of the hydrodynamic part of the code. We use the same tests employed in the previous section for the the collisionless component. However, since in this case the solution refers to an Eulerian grid, in order to have the velocity and density at a given grid location $x(q, t)$ from Eqs. (78) and (79) we must invert Eq. (77). We first consider the errors in the linear regime, in analogy to Tables 1 and 2 for the collisionless component. In Table 4 we report the L_1, L_2 and L_∞ norms of the error for the gas density, velocity and force and the corresponding convergence rates, for the case of fixed time step, $\frac{\Delta t}{\Delta x} = 1.6 \times 10^{-4}$. Similarly, the left hand side of Table 5 reports the L_1, L_2 and L_∞ errors for the case in which the time step is set by the background expansion rate. The L_2 errors for the fixed and varying timestep cases are also shown by the filled symbols in the left and right hand side panels of Fig. 5, respectively. These tests show the second order
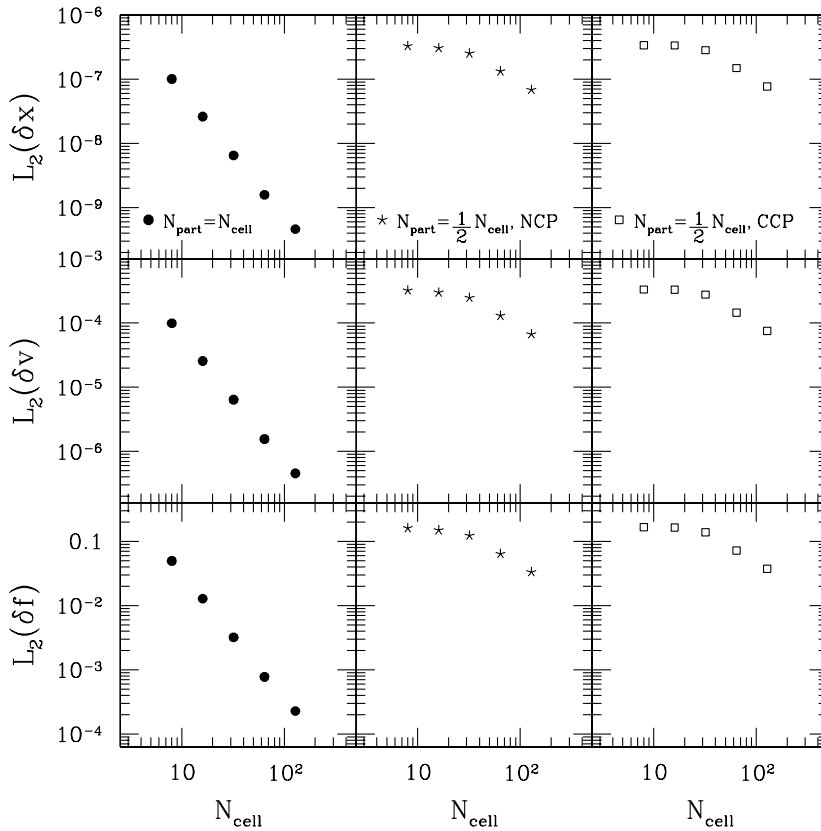
Fig. 4. Same as Fig. 1 but for the case in which a staggered scheme is used for the force calculation.

Table 4
Convergence tests: Godunov's scheme[a]

N_{part}	L_1	R_1	L_2	R_2	L_∞	R_∞
Density						
8	2.7e − 05	2.2	2.9e − 05	1.9	4.2e − 05	1.2
16	5.7e − 06	2.2	7.6e − 06	2.2	1.8e − 05	1.8
32	1.2e − 06	2.0	1.7e − 06	2.2	5.3e − 06	1.9
64	3.0e − 07	2.0	3.8e − 07	2.1	1.4e − 06	1.9
128	7.4e − 08	–	8.8e − 08	–	3.7e − 07	–
Velocity						
8	3.3e − 05	2.0	3.6e − 05	2.0	5.2e − 05	2.0
16	8.1e − 06	2.0	9.0e − 06	2.0	1.3e − 05	2.0
32	2.0e − 06	2.0	2.2e − 06	2.0	3.2e − 06	2.0
64	5.0e − 07	2.0	5.6e − 07	2.0	7.9e − 07	2.0
128	1.2e − 07	–	1.4e − 07	–	2.0e − 07	–
Force						
8	1.6e − 02	2.0	1.7e − 02	2.0	2.4e − 02	2.0
16	3.9e − 03	2.0	4.3e − 03	2.0	6.1e − 03	2.0
32	9.7e − 04	2.0	1.1e − 03	2.0	1.5e − 03	2.0
64	2.4e − 04	2.0	2.7e − 04	2.0	3.8e − 04	2.0
128	6.0e − 05	–	6.7e − 05	–	9.5e − 05	–

[a] We use a cell-centered force scheme and a constant timestep, $\frac{\Delta t}{\Delta x} = 1.6 \times 10^{-4}$.

Table 5
Convergence tests: Godunov's scheme, linear regime[a]

N_{part}	L_1	R_1	L_2	R_2	L_∞	R_∞
Density						
8	2.0e − 04	2.2	2.1e − 04	1.9	3.0e − 04	1.1
16	4.2e − 05	2.1	5.7e − 05	2.1	1.4e − 04	1.7
32	9.5e − 06	2.0	1.3e − 05	1.9	4.3e − 05	1.4
64	2.4e − 06	1.9	3.5e − 06	1.5	1.6e − 05	0.9
128	6.5e − 07	–	1.3e − 06	–	8.4e − 06	–
Velocity						
8	2.1e − 04	2.0	2.3e − 04	2.0	3.3e − 04	2.0
16	5.1e − 05	2.0	5.7e − 05	2.0	8.2e − 05	2.0
32	1.3e − 05	2.0	1.4e − 05	2.0	2.0e − 05	2.1
64	3.2e − 06	2.0	3.5e − 06	2.0	5.0e − 06	1.9
128	7.9e − 07	–	8.9e − 07	–	1.3e − 06	–
Force						
8	1.5e − 02	2.0	1.7e − 02	2.1	2.3e − 02	2.0
16	3.6e − 03	2.0	4.0e − 03	2.0	5.7e − 03	2.0
32	9.0e − 04	2.0	1.0e − 04	2.0	1.4e − 03	2.0
64	2.2e − 04	2.0	2.5e − 04	2.0	3.5e − 04	2.0
128	5.6e − 05	–	6.2e − 05	–	8.9e − 05	–

[a] We use a cell-centered force scheme, a variable timestep, $\frac{\Delta t}{\Delta x} = C_{exp}(\frac{a}{\dot{a}})$, $C_{exp} = 10^{-2}$, and $a = 0.0221$.
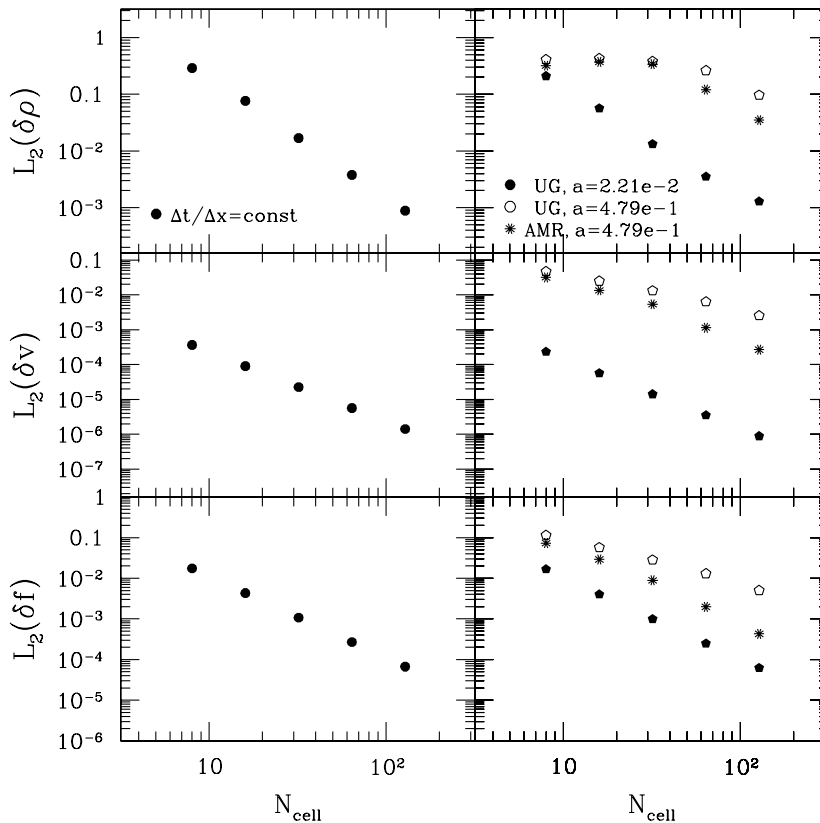
Fig. 5. L_2 norm of the error in density (top), velocity (center) and force (bottom) as a function of the number of grid cells. See legend for the meaning of the symbols (UG = Uniform Grid, a is the expansion parameter).

accuracy of the implemented scheme. We note, however, that in the case of variable time step the convergence rate of the L_∞ norm of the density error is slower. Next, the left hand side of Table 6 reports the errors and convergence rates for the case of a uniform grid calculation, well in the non-linear regime, close the formation of the caustic ($a = 0.479$). As before, the L_2 errors are also shown as open symbols in Fig. 5. Unlike the collisionless case, here is the convergence rate of the density that is mostly affected, particularly at low resolutions (cf. [31,45]). As illustrated by the L_∞ norm, the error is dominated by the contribution of a few cells, located where the caustic forms.

Finally, we test the performance of the AMR code. As for the collisionless component, we use a constant refinement ratio, $n_{ref} = 2$, refine cells enclosing a mass larger than 1.5 the average value and allow for a max of three levels of refinement. The results at $a = 0.476$ are reported in the right hand side columns of Table 6 for the L_1, L_∞ errors and convergence rates. There we also show the maximum level employed by each run. L_2 errors are reported as spur symbols in the right panels of Fig. 5.

The use of refined grids in terms of fraction of grid covered and fraction of the simulation time is very similar to the corresponding collisionless case. Similar is also the benefit of AMR, which improves the convergence of the solution to rates very similar to those characterizing the linear regime. This is indeed a powerful performance of the AMR technique.

4.2. Effect of C_{exp} on the solution quality

We have investigated how the error depends on the choice of the parameter C_{exp} for the above problem. In particular we have computed the error accumulated during an interval $\Delta a \ll a_{collapse}$, for values of the expansion parameter $a = 0.196$ and $a = 0.091$, and for values of C_{exp} ranging from 10^{-3} to 0.5. We consider both the fluid and the collisionless case. We use a uniform grid with 32 zones on a side and, for the collisionless case, we use one particle per cell. The results are rather independent of the norm type. We find that the errors introduced in the particles velocity and position is quite stable, except for the largest values of C_{exp}. On the other hand, the errors in the fluid components decrease steadily as C_{exp} is reduced, spanning a factor ~ 5 before reaching a plateau for $C_{exp} \leqslant (1–2) \times 10^{-2}$.

Table 6
Convergence tests: Godunov's scheme, non-linear regime[a]

N_{part}	Uniform grid				AMR				
	L_1	R_1	L_∞	R_∞	L_1	R_1	L_∞	R_∞	l_{max}
Density									
8	2.2e − 01	0.5	8.2e − 01	–	1.6e − 01	0.7	8.2e − 01	–	1
16	1.6e − 01	0.7	1.2e00	–	1.0e − 01	0.6	1.5e00	–	1
32	9.7e − 02	0.9	1.5e00	0.0	6.4e − 02	2.0	2.4e00	1.3	2
64	5.0e − 02	1.6	1.5e00	1.0	1.6e − 02	2.0	1.0e00	1.1	2
128	1.6e − 02	–	7.3e − 01	–	3.9e − 03	–	4.7e − 01	–	3
Velocity									
8	2.7e − 02	1.4	9.4e − 02	0.4	1.6e − 02	1.8	8.6e − 02	0.7	1
16	1.0e − 02	1.4	7.0e − 02	0.4	4.7e − 03	1.7	5.3e − 02	0.6	1
32	3.8e − 03	1.5	5.3e − 02	0.6	1.4e − 03	2.4	3.6e − 02	2.2	2
64	1.3e − 03	1.6	3.6e − 02	0.9	2.6e − 04	1.8	9.6e − 03	1.9	2
128	4.3e − 04	–	2.0e − 02	–	7.2e − 05	–	3.1e − 03	–	3
Force									
8	6.8e − 02	1.6	2.3e − 01	0.5	3.7e − 02	1.7	2.0e − 01	0.9	1
16	2.3e − 02	1.5	1.6e − 01	0.5	1.1e − 02	2.3	1.1e − 01	0.8	1
32	8.4e − 03	1.5	1.1e − 01	0.6	2.2e − 03	2.0	6.3e − 02	1.7	2
64	2.9e − 03	1.6	7.3e − 02	0.9	5.5e − 04	1.8	1.9e − 02	2.2	2
128	9.5e − 04	–	3.9e − 02	–	1.6e − 04	–	4.0e − 03	–	3

[a] We use a cell-centered force scheme, a variable timestep, $\frac{\Delta t}{\Delta x} = C_{exp}\left(\frac{a}{\dot{a}}\right)$, $C_{exp} = 10^{-2}$, and $a = 0.479$.

4.3. Homologous dust cloud spherical collapse

In this section we test the ability of the code to follow the collapse of a pressure-less (dust) sphere of matter [46]. The problem is described by the following equations

$$\left(\frac{\partial^2 r}{\partial t^2}\right)_M = -G\frac{M(r)}{r^2} \tag{80}$$

$$u = \left(\frac{\partial r}{\partial t}\right)_M \tag{81}$$

with initial conditions

$$\rho(\mathbf{i}, t = 0) = \begin{cases} f[r(\mathbf{i})] & \text{if } r(\mathbf{i}) \leqslant R \\ 0 & \text{if } r(\mathbf{i}) > R \end{cases} \tag{82}$$

$$u(\mathbf{i}, t = 0) = 0 \tag{83}$$

where $M(r)$ is the mass enclosed within a distance r from the sphere center, R is the radius of the sphere and $f(r)$ is a function (with $f'(r) \leqslant 0$) that depends solely on r. For the hydrodynamic case null density will be approximated with a value $\rho(r > R) \ll \min[\rho(r)]$. The problem admits a self similar solution in implicit form which reads

$$(1 - \xi)^{1/2}\xi^{1/2} + \sin^{-1}(1 - \xi)^{1/2} = \tau \tag{84}$$

$$r = \xi r_0, \quad u = \frac{r_0}{\tau_c}\left(\xi^{-1} - 1\right)^{1/2}, \quad \tau \equiv t\left(\frac{8\pi G\langle\rho\rangle_r}{3}\right)^{1/2} \tag{85}$$

where $\langle\rho\rangle_r = 3M(r)/4\pi r^3$ is the average density within a radius r.

From the numerical point of view, the problem is challenging in two respects: during collapse the force potential becomes progressively steeper and, therefore, more demanding for the gravity solver. In addition, since the problem has inherent radial symmetry and we are solving it on a Cartesian grid, the ability of the code at preserving that symmetry will be tested. For the collisionless component we initially set particles with null velocity at the center of cells whose distance from the cloud center is less than R (=1). This produces a homogeneous density distribution everywhere inside R, except close to the cloud edge due to the discreteness of the grid. In Fig. 6 we compare the position of each particle in phase-space (v_r, r) as given by the code (black filled triangles) with the analytic solution (cyan open circle). The left panel corresponds to the case in which a
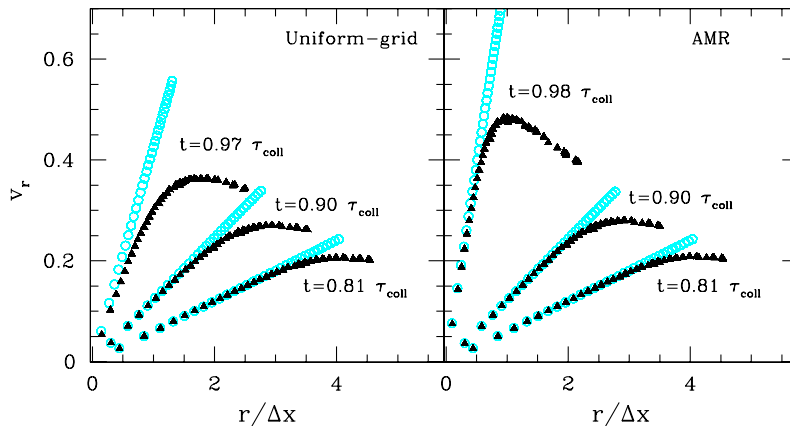

Fig. 6. Phase space distribution of particles for the spherical collapse of a pressureless cloud. Cyan open circles correspond to the analytic solution and filled triangles to the numerical simulation result. The right panel is for the case of a uniform grid whereas the left panel correspond to an AMR calculation with two levels of refinement. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

uniform grid is used whereas for the right panel solution AMR was employed. We allow for two levels of refinement and tag cells with a total mass four times as high as the initial value. The comparison between analytic and numerical solution in Fig. 6 is made for a number of evolution times, expressed in terms of the adimensional collapse time $\tau_{coll} = \pi/2$. Noticeably, the code follows the particles motions with high accuracy all the way down to the time of collapse. In particular, there is no sign of artificial asymmetries. Additional levels of refinement were dynamically generated towards the final phase of the collapse. At the latest time shown ($\tau = 0.98\tau_{coll}$), we can see the improvement due to the employment of finer grids in the collapsing region. Note that towards the edge of the cloud the particles are trailing. This is due to the inability to reproduce a perfectly homogeneous sphere near the cloud edge from the beginning. The region affected by this is about one base mesh wide.

When exploring the accuracy of cell-centered versus face-centered force schemes our tests suggest that, again, in the uniform grid case the latter perform slightly better, at the level of ca. 15%. In analogy with the analysis of Section 4.1, we have tested this further, for the case in which the ratio of cells to particles is significantly larger than one. This situation may easily occur depending on the adopted criterion for refinement and on the efficiency for the generation of the refined grid out of the tagged cells. In Fig. 7 we compare the solutions obtained with a cell-centered (left) and a staggered (right) force scheme for an initial cell-to-particle ratio of 4. The initial dust sphere is placed on a grid of 256 cells on a side, and eight particles are aligned along its radius out to 32 cells from its center. The code output is plotted at three different solution times close to the time of collapse $t = \tau_{coll}$. This test shows that when the number of cell-to-particle ratio is significantly higher than one, the staggered force scheme tends to produce spurious results. This is in line with the findings in the previous test problem in Section 4.1. On the other hand, the cell-centered scheme seems well behaved. We find that the qualitative result does not change when we use a two or four point force stencil, when the number of particles to resolve the sphere is changed or when the sphere center is shifted by a fraction of a mesh size in an arbitrary direction.

Thus the staggered force scheme, although apparently more accurate than its cell-centered counterpart when the number of cells is comparable to, or less than, the number of particles [44], it gives rise to spurious results when particles are sparse on the grid. Therefore, caution must be exercised when employing staggered schemes for force evaluation.

Finally, the results for the collisional case are illustrated in Fig. 8. The plot compares the density (top) and velocity (bottom) profiles of the numerical solution (threaded black dots) and the analytic solution (cyan dots). The latter extend only out the cloud size, whereas the numerical solution includes the region covered by the finest level. Two times during the collapse are shown: $t = 0.72\tau_{coll}$ and $t = 0.92\tau_{coll}$, corresponding to the low and high curves, respectively. At these times one and two levels of refinement have been generated, respectively. The chosen times are close to the collapse time, when the errors have accumulated and the sim-
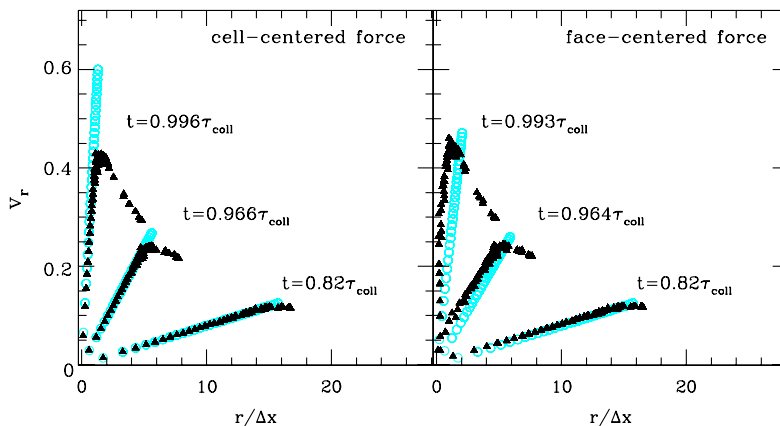

Fig. 7. Phase space distribution of particles for the spherical collapse of a pressureless cloud. The initial cell-to-particle ratio is 4. Cyan open circles correspond to the analytic solution and filled triangles to the numerical simulation result using a cell-centered (left) and face-centered (right) force scheme, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
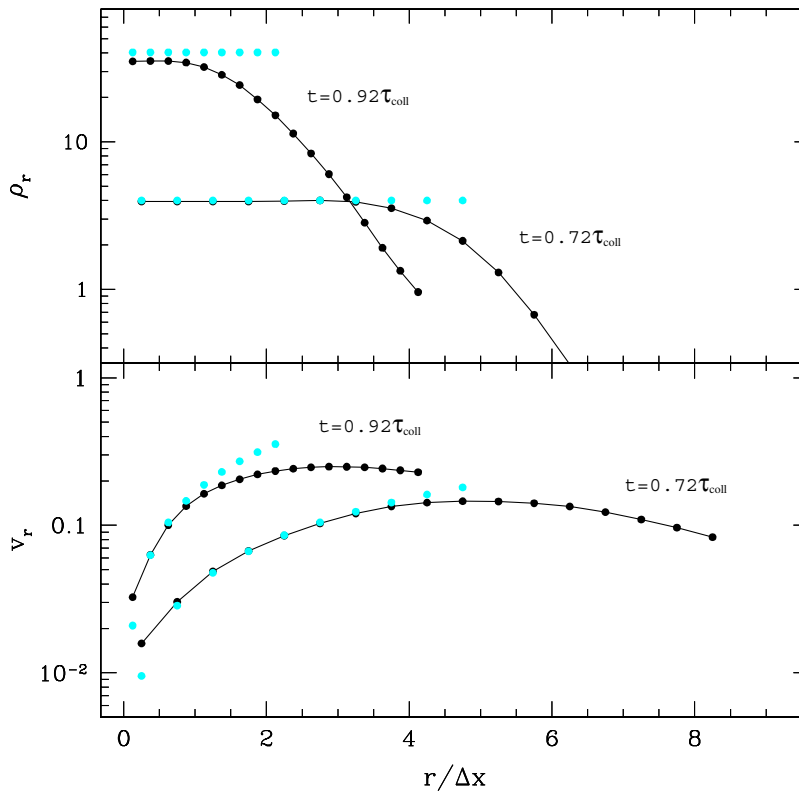
Fig. 8. Density and velocity profile for spherical collapse of a pressureless cloud of gas. Cyan dots correspond to the analytic solution and black dots to the numerical simulation results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

ulation becomes more challenging. Nevertheless, as for the collisionless component, the code follows accurately the evolution of the density and velocity profiles of the collapsing cloud. Again, close to the cloud edge the density profile is smoother and the velocity field slower than the analytic solution. The size of the region affected by this is again of order of the coarse mesh size and it is partially ascribed to the crude representation of the cloud edge on the grid.

4.4. Energy conservation: Layzer–Irvine equation

For pure hydrodynamics conservation of the total (kinetic + thermal) energy is enforced by our conservative Godunov's method. When gravity is added, energy conservation should still hold, but is not explicitly enforced in our scheme. Finally, with an expanding background energy is not conserved. For a collection of particles that interact only gravitationally, the evolution of the energy of the system is regulated by the Layzer–Irvine equation, which reads

$$\frac{\mathrm{d}}{\mathrm{d}t}[a(t)(\mathcal{E} + \mathbf{W})] = -\dot{a}\mathcal{E} \tag{86}$$

where $\mathcal{E}$ is the kinetic energy associated to the peculiar motions of the particles and $\mathcal{W}$ their gravitational potential energy due to the overdensity produced by their mass distribution. Clearly in absence of expansion ($a = 1, \dot{a} = 0$) Eq. (86) reduces to the ordinary energy conservation equation. Otherwise it describes the change in the total energy of the system due to the adiabatic expansion of the background. The derivation and physical meaning of Eq. (86) is reviewed in Ref. [27]. Its applicability to hydrodynamic simulations is discussed in, e.g. Ref. [32], in which case a monoatomic gas is assumed and $\mathcal{E}$ includes both the kinetic and thermal energy of the gas. Eq. (86) can be integrated in time giving

$$a[\mathcal{E}(a) + \mathcal{W}(a)] - a_0[\mathcal{E}(a_0) + \mathcal{W}(a_0)] = -\int_{a_0}^{a} \mathcal{E}\, \mathrm{d}a. \tag{87}$$

We can evaluate the integral on the RHS of the above equation numerically with the trapezoidal rule and assess the accuracy of the code at tracking the energy of the system through the quantity

$$\delta\varepsilon = \frac{a[\mathcal{E}(a) + \mathcal{W}(a)] - a_0[\mathcal{E}(a_0) + \mathcal{W}(a_0)] + \int_{a_0}^{a} \mathcal{E}\, \mathrm{d}a}{[a_0\mathcal{W}(a_0) - a\mathcal{W}(a)]}. \tag{88}$$

We first test the energy conservation accuracy of the code for the case of the collapse of a pressureless cloud. This is the problem studied in the previous section. We carry out three AMR calculation with different base grid sizes, namely 16, 32, 64 corresponding to 4, 8, 16 cells per cloud radius, respectively. In these runs, cells enclosing more then four times the initial mass content were tagged for refinement and a maximum of two 2 refinement levels were allowed. The results of the test are reported in the left hand side panel of Fig. 9 where we plot the error in the total energy, $\delta\varepsilon$, as a function of resolution, for both the particle (open) and the gas-dynamic (filled) case. The plots show that with 16 zones per cloud radius the error in the energy is at the level of a per cent or so.

Next we test code accuracy at tracking the energy of the system in a cosmological run. For the purpose we use a Λ-Cold Dark Matter cosmology with parameters $\Omega_m = 0.3$, $\Omega_\Lambda = 0.7$, $\Omega_b = 0.04$, for the energy density in total matter, dark energy and baryonic matter, respectively; and $H_0 = 70$ km s^{-1} Mpc^{-1} for the Hubble constant. The physical domain has a size of $L = 91.43$ Mpc on a side. We execute three runs with different numerical resolution. The first two runs employ a uniform grid with 32^3 and 64^3 cells, respectively, and the same number of particles as grid cells. The third run uses a base grid with 32^3 cells and 32^3 particles, and two additional levels of refinement created in region where the total mass enclosed in a cell exceeds the initial value by a factor eight. The initial conditions where generated on a 64^3 grid and coarse-averaged to a 32^3 grid for the low-resolution-uniform and AMR runs.

The results are presented in the right panel of Fig. 9 where we plot $\delta\varepsilon$ (top), $\mathcal{E}$ (middle) and $\mathcal{W}$ (bottom) as a function of expansion parameter a for each resolution case. The plots show that when using a uniform grid the total energy of the system is evolved with an accuracy at the percent level ($\sim 2\%$ and 1% for the 32^3 (dot) and 64^3 (dash) cases, respectively), with most of the error generated at startup. In the AMR case (solid line), however, our error parameter $\delta\varepsilon$ increases visibly when refinement levels are created (at $a \sim 0.15$ and $a \sim 0.2$ for the first and second level, respectively). The reason for this is simple. When a level of refinement is created the potential energy of the system changes suddenly (see solid and dot lines in the bottom panel) throwing off the balance between particle/gas velocities and their potential energy. As a result a large error in the sense of Eq. (88) is generated. This is so, even though with the additional level of refinement the potential energy of the system is more accurate as it gets closer the value from the higher resolution run (see solid and dash lines
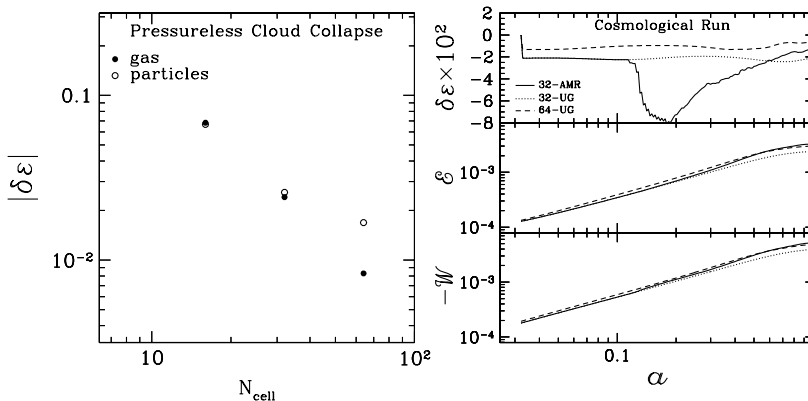

Fig. 9. Energy conservation error based on Eq. (88) for the collapse of a pressureless spherical cloud (left) and a cosmological run (right). See text for details.

in the bottom panel). Over time the kinetic energy readjusts to the new potential (middle panel; notice that the internal energy is negligible) and a balance between the two forms of energy is reestablished. Because the potential energy associated with the newly formed structures is larger than that of the system at the time when the refinement levels were first generated, the new balance between kinetic and potential energy reduces substantially the error as time progresses. At simulation end the AMR run (with a base grid of 32^3 cells) produces estimates of $\delta\varepsilon$, $\mathcal{E}$ and $\mathcal{W}$ very close to the high resolution run.

4.5. Santa Barbara galaxy cluster

In this section we carry out the calculation defined by the 'Santa Barbara Cluster Comparison Project' [47] and compare the results of our code with those from different codes implemented independently by other authors and based either on similar or different techniques.

The problem consists of simulating the formation of a galaxy cluster in a Standard Cold Dark Matter universe. The cosmological parameters assumed were $\Omega_m = 1$ and $\Omega_b = 0.1$ for the total and baryonic mean mass density in units of the critical density, respectively; $H_0 = 50$ km s^{-1} Mpc^{-1} for the Hubble constant; $\sigma_8 = 0.9$ for the present-day linear rms mass fluctuation in spherical top hat spheres of radius 16 Mpc. The computational domain has a size of $L = 64$ Mpc on a side. The initial matter fluctuation are characterized by a power spectrum with an asymptotic spectral index, $n = 1$, and are 'constrained' so that at simulation end a massive structure has formed at the center of the computational box.

The simulation was initialized at $z = 40$ with two grids already in place: a base grid covering the entire 64 Mpc^3 domain with 64^3 cells and 64^3 particles; and a second grid, also with 64^3 cells and 64^3 particles, but only 32 Mpc on a side and placed in the central region of the base grid, thus yielding an initial cell size of 0.5 Mpc. Refinement is applied only in this central, higher resolution region and is based on a local density criterion: cells with a total mass of 6.4×10^{10} M$_\odot$ or more were refined. We allowed for five levels of refinement (for a total of a six levels hierarchy), with a constant refinement ratio $n_{ref} = 2$. The size of the finest mesh is about 15 comoving kpc. We use the following CFL coefficients for the time step: $C_{hydro} = 0.8$, $C_{part} = 0.5$ and $C_{exp} = 0.02$.

At simulation end a halo finder based on the spherical overdensity method [48] was run in order to define the center of the galaxy cluster. The radial profiles for six quantities of interest are presented in Fig. 10 together with results from two other simulation codes: ENZO, which is an Eulerian AMR code similar to ours, and HYDRA (as run by Jenkins & Pearce), which combines smoothed particle hydrodynamics (SPH) and adaptive particle–particle–particle–mesh (AP^{3}M) method for the N-body part [49]. These two codes are meant to be representative of the high resolution grid based and SPH approaches, respectively. Results are shown down to scales of about 30 kpc which is just above the nominal resolution at finest level of refinement. The plot shows that there is good agreement among the results of the different codes particularly with ENZO (even though we used one less level of refinement and use a different refinement criterion). The discrepancies, when significant, are consistent with those already found in the extensive comparison paper in Ref. [47]. In fact, there is very good agreement in terms of dark matter density distribution (top left) which is well fit by the analytical form proposed in Ref. [50] with parameters specified in the caption. Similarly, there is good matching of the solution in terms of gas density distribution, except in the inner regions within 100 kpc, where the two AMR solutions flatten and the SPH solution keeps on increasing. More significant is the difference in temperature distribution (middle left), which drops in the SPH solution for the inner regions and stays constant for the AMR case. These differences were already found and discussed in Ref. [47]. (See also Ref. [19] for similar findings.) Their origin is not fully clear, although as suggested in Ref. [47], it could be ascribed to the different way in which SPH and grid based methods treat shocks.

Next panel (middle right) shows the profile for the ratio of gas to dark matter mass, normalized to the global value. Our solution is in good agreement with ENZO's but deviates from HYDRA's up to a few Mpc or so. There is significant scatter in the results from the full set of codes found in Ref. [47], at the level of 0.1–0.2. Nevertheless, it is pointed out in Ref. [47] that when integrated out to the virial radius of the system (2.7 Mpc), the SPH codes seem to predict a systematically smaller values for this quantity than high resolution grid based codes. The reason for this is still not clear.
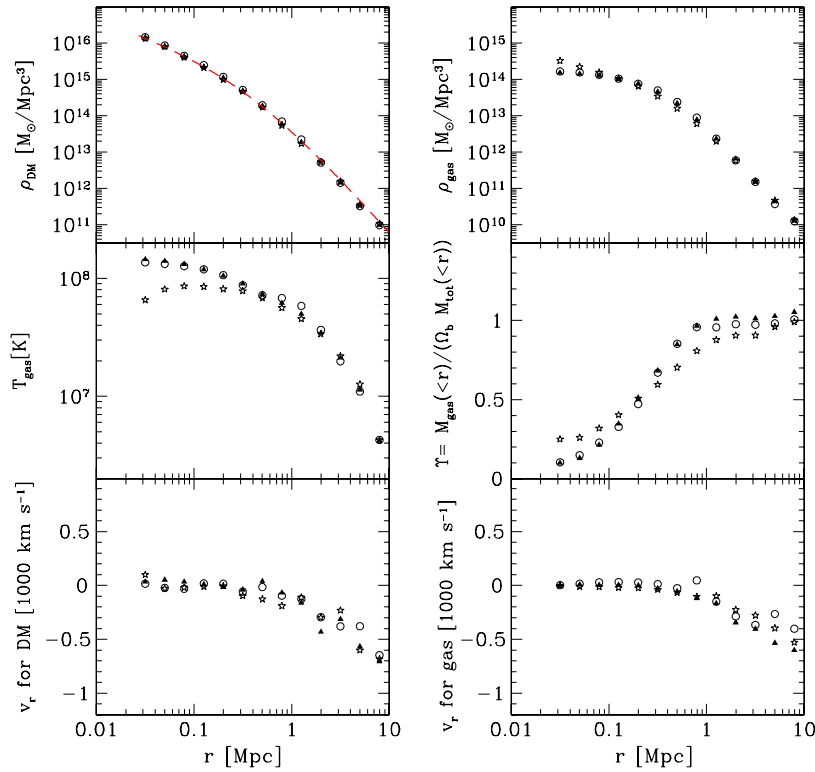
Fig. 10. Radial profile of dark matter (top left), baryonic gas (top right) temperature (middle left), baryonic fraction (middle right), radial velocity for dark matter (bottom left) and gas (bottom right). In addition to the results from CHARM (open circles), for comparison we also show those from the ENZO AMR code (filled triangles) [21] as well as those from the HYDRA SPH code (open stars) [49].

Finally, both the gas and dark matter radial velocity profiles agree quite well at all radii. Some differences may arise due to slight differences in the simulation timing, as pointed out in Ref. [47]. Note that at larger radii (last few points), typically characterized by wider scatter, both ENZO's and HYDRA's results tend to be below the average value defined in Ref. [47].

5. Conclusions

We have presented a new code based on AMR technique for systems comprising collisional and collisionless components coupled through a long range force. We have thus extended the scheme in [2] to include collisionless particle dynamics and gravity arising from the mass distribution of the two components. For the hydrodynamics we use a slightly modified directionally unsplit Godunov's method based on Ref. [28]. As for the collisionless component we have implemented various time centered modified symplectic schemes based on both the kick–drift–kick and drift–kick–drift sequence. Our implementations of these schemes appear to perform comparably. We have also used different types of stencil to calculate the force from the potential. We find that while the staggered schemes appear more accurate when the number (density) of particles is at least as large as the number of grid cells, it produces spurious results when the particles are sparse on the grid. Cell centered stencils thus seem more reliable, especially when a five point cell centered discretization of the Laplacian operator is used.

Due to the time refinement character of the AMR technique the solution on different levels is advanced with different timesteps. Synchronization issues then arise as the multilevel solution to the elliptic equation needs to be solved simultaneously on all levels. In particular, the density field represented by the particles evolved on finer levels may not be available on coarser levels when they are not synchronized. Similarly, one cannot account for the effects of the mass distribution on finer levels on the multilevel solution of the potential, unless

all levels are synchronized. Among other features of the code, we have thus introduced an *aggregation* procedure to cost-effectively represent on the coarser levels the particles on finer levels without compromising the code accuracy and performance. We have also implemented a procedure for estimating (when the coarse and finer levels are not synchronized) the effects on the coarse potential produced by the matching conditions at refinement boundaries between coarse and fine solution, as they would arise in a full multilevel calculation. We performed several standard tests which illustrate the code accuracy as well as the advantages of the AMR technique for the study of both self gravitating hyperbolic systems, collisionless system and hybrid systems.

Acknowledgments

Appendix A. Charge assignment schemes

In one dimension the following Charge Assignment Schemes are commonly used. The first order accurate nearest grid point (NGP) scheme, defined as

$$W(x) = \begin{cases} 1 & \text{if } |x| \leqslant \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{A.1}$$

in which the assigned charge distribution is discontinuous as the particle crosses the cell boundary. The second order accurate cloud in cell (CIC) scheme, defined as

$$W(x) = \begin{cases} 1 - |x| & \text{if } |x| \leqslant 1 \\ 0 & \text{otherwise} \end{cases} \tag{A.2}$$

in which the assigned charge distribution is continuous but the first derivative is not. And the third order accurate triangular shape cloud (TSC) scheme, defined as

$$W(x) = \begin{cases} \frac{3}{4} - |x|^2 & \text{if } |x| \leqslant \frac{1}{2} \\ \frac{1}{2} - \left(\frac{3}{2} - |x|\right)^2 & \text{if } \frac{1}{2} \leqslant |x| \leqslant \frac{3}{2} \\ 0 & \text{otherwise} \end{cases} \tag{A.3}$$

in which both the assigned charge distribution and first derivative are continuous. These schemes retain their properties when they are extended to more than one dimension in the form

$$W(\mathbf{x}) = \prod_{j=1}^{\mathbf{D}} W_j(x_j). \tag{A.4}$$

References

[1] M.J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comp. Phys. 53 (1984) 484–512.
[2] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comp. Phys. 82 (1989) 64–84.
[3] F.X. Timmes, M. Zingale, K. Olson, B. Fryxell, P. Ricker, A.C. Calder, L.J. Dursi, H. Tufo, P. MacNeice, J.W. Truran, R. Rosner, On the cellular structure of carbon detonations, Astrophys. J. 543 (2000) 938–954.
[4] K. Kifonidis, T. Plewa, H.-T. Janka, E. Müller, Nucleosynthesis and clump formation in a core-collapse supernova, Astrophys. J. Lett. 531 (2000) L123–L126.
[5] A.S. Almgren, J.B. Bell, C.A. Rendelman, M. Zingale, Low Mach number modeling of type Ia supernovae. I. Hydrodynamics, Astrophys. J. 637 (2006) 922–936.
[6] A.S. Almgren, J.B. Bell, C.A. Rendelman, M. Zingale, Low Mach number modeling of type Ia supernovae. II. Energy evolution, Astrophys. J. 649 (2006) 927–938.

[7] R.I. Klein, C.F. McKee, P. Colella, On the hydrodynamic interaction of shock waves with interstellar clouds. 1: Nonradiative shocks in small clouds, Astrophys. J. 420 (1994) 213–236.

[8] R. Walder, D. Folini, Knots, filaments, and turbulence in radiative shocks, Astr. Astrophys. 330 (1998) L21–L24.

[9] R.I. Klein, R.T. Fisher, M.R. Krumholz, C.F. McKee, Recent advances in the collapse and fragmentation of turbulent molecular cloud cores, in: Revista Mexicana de Astronomia y Astrofisica Conference Series, 2003, pp. 92–96.

[10] J.K. Truelove, R.I. Klein, C.F. McKee, J.H. Holliman, L.H. Howell, J.A. Greenough, The Jeans condition: a new constraint on spatial resolution in simulations of isothermal self-gravitational hydrodynamics, Astrophys. J. Lett. 489 (1997) L179.

[11] J.K. Truelove, R.I. Klein, C.F. McKee, J.H. Holliman, L.H. Howell, J.A. Greenough, D.T. Woods, Self-gravitational hydrodynamics with three-dimensional adaptive mesh refinement: methodology and applications to molecular cloud collapse and fragmentation, Astrophys. J. 495 (1998) 821.

[12] T. Abel, G.L. Bryan, M.L. Norman, The formation of the first star in the Universe, Science 295 (2002) 93–98.

[13] M. Norman, Cosmological simulations of X-ray clusters: the quest for higher resolution and essential physics, in: S. Bowyer, C.-Y. Hwang (Eds.), Astronomical Society of the Pacific Conference Series, vol. 301, 2003, pp. 419.

[14] D.F. Martin, K.L. Cartwright, Solving Poisson's equation using adaptive mesh refinement, Technical Report UCB/ERI M96/66 UC Berkeley.

[15] P.M. Ricker, K. Olson, K.M. Riley, F. Miniati, et al., Adding self-gravitation and collisionless particles to an adaptive-mesh hydrodynamics code (in preparation).

[16] A.V. Kravtsov, High-resolution simulations of structure formation in the Universe. Ph.D. Thesis.

[17] A. Knebe, A. Green, J. Binney, Multi-level adaptive particle mesh (MLAPM): a c code for cosmological simulations, Mon. Not. R. Astron. Soc. 325 (2001) 845–864.

[18] R. Teyssier, Cosmological hydrodynamics with adaptive mesh refinement. A new high resolution code called RAMSES, Astr. Astrophys. 385 (2002) 337–364.

[19] V. Quilis, A new multidimensional adaptive mesh refinement hydro + gravity cosmological code, Mon. Not. R. Astron. Soc. 352 (2004) 1426–1438.

[20] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, McGraw Hill, New York, 1981.

[21] G.L. Bryan, M.L. Norman, A hybrid amr application for cosmology and astrophysics, in: D.A. Clarke, M. Fall (Eds.), Computational Astrophysics, ASP Conference, 1997.

[22] G. Strang, On the construction and comparison of difference schemes, SIAM J. Num. Anal. 5 (1968) 506–517.

[23] J. Vay, P. Colella, P. McCorquodale, B.V. Straalen, A. Friedman, D. Grote, Mesh refinement for particle-in-cell plasma simulations: applications to and benefits for heavy ion fusion, Laser Particle Beams 20 (4) (2002) 75–596.

[24] G.H. Miller, P. Colella, A conservative three-dimensional Eulerian method for coupled solid–fluid shock capturing, J. Comp. Phys. 183 (1) (2002) 26–82.

[25] A. Almgren, J. Bell, P. Colella, L. Howell, M. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, J. Comput. Phys. 142 (1998) 1–46.

[26] D. Martin, P. Colella, A cell-centered adaptive projection method for the incompressible Euler equations, J. Comput. Phys. 163 (2000) 271–312.

[27] P.J.E. Peebles, Principles of Physical Cosmology, Princeton University Press, Princeton, NJ, 1993.

[28] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, J. Comp. Phys. 82 (1989) 64–84.

[29] J. Saltzman, An unsplit 3D upwind method for hyperbolic conservation laws, J. Comput. Phys. 115 (1994) 153–168.

[30] F. Miniati, D. Ryu, H. Kang, T.W. Jones, R. Cen, J. Ostriker, Properties of cosmic shock waves in large-scale structure formation, Astrophys. J. 542 (2000) 608–621.

[31] G.L. Bryan, M.L. Norman, J.M. Stone, R. Cen, J.P. Ostriker, A piecewise parabolic method for cosmological hydrodynamics, Comp. Phys. Commun. 89 (1995) 149–168.

[32] D. Ryu, J.P. Ostriker, H. Kang, R. Cen, A cosmological hydrodynamic code based on the total variation diminishing scheme, Astrophys. J. 414 (1993) 1.

[33] T. Quinn, N. Katz, J. Stadel, G. Lake, Time stepping N-body simulations; e-archive: astro-ph/9710043.

[34] W. Kahan, R.-C. Li, Unconventional schemes for a class of ordinary differential equations– with applications to the Korteweg–de Vries equation, J. Comp. Phys. 134 (2) (1997) 316–331.

[35] R.J. LeVeque, Numerical Methods for Conservation Laws, second ed., Springer, Verlag, 1992.

[36] M. Minion, A projection method for locally refined grids, J. Comp. Phys. 127 (1996) 158–178.

[37] D.F. Martin, P. Colella, D. Graves, A cell-centered adaptive projection method for the incompressible Navier–Stokes equations in three dimensions, J. Comp. Phys., submitted for publication.

[38] R.T. Fisher, Single and multiple star formation in turbulent molecular cloud cores, Ph.D. thesis, University of California, Berkeley, 2002.

[39] S.L.W. McMillan, S.J. Aarseth, An O($N\log N$) integration scheme for collisional stellar systems, Astrophys. J. 414 (1993) 200–212.

[40] F. Miniati, P. Colella, A modified Godunov's scheme for stiff source conservative hydrodynamics, J. Comp. Phys. 224 (2007) 519–538.

[41] F. Miniati, Cosmocr: A numerical code for cosmic ray studies in computational cosmology, Comp. Phys. Commun. 141 (2001) 17–38.

[42] F. Miniati, Glimm–Godunov's method for cosmic-ray-hydrodynamics, J. Comp. Phys. (in press) (eprint arxiv:astro-ph/0611499).

[43] Y.B. Zel'dovich, Gravitational iinstability: an approximate theory for large density perturbations, Astr. Astrophys. 5 (1970) 84–89.

[44] A.L. Mellot, Comment on "nonlinear gravitational clustering in cosmology", Phys. Rev. Lett. 56 (1986) 1992.

[45] L. Feng, C. Shu, M. Zhang, A hybrid cosmological hydrodynamic/N-body code based on a weighted essentially nonoscillatory scheme, Astrophys. J. 612 (2004) 1–13.

[46] S.A. Colgate, R.H. White, The hydrodynamic behavior of supernovae explosions, Astrophys. J. 143 (1966) 626.

[47] C.S. Frenk, S.D.M. White, P. Bode, J.R. Bond, G.L. Bryan, R. Cen, H.M.P. Couchman, A.E. Evrard, N. Gnedin, A. Jenkins, A.M. Khokhlov, A. Klypin, J.F. Navarro, M.L. Norman, J.P. Ostriker, J.M. Owen, F.R. Pearce, U.L. Pen, M. Steinmetz, P.A. Thomas, J.V. Villumsen, J.W. Wadsley, M.S. Warren, G. Xu, G. Yepes, The santa barbara cluster comparison project: A comparison of cosmological hydrodynamics solutions, Astrophys. J. 525 (1999) 554.

[48] C. Lacey, S. Cole, Merger rates in hierarchical models of galaxy formation – part two – comparison with n-body simulations, Mon. Not. R. Astron. Soc. 271 (3) (1994) 676.

[49] H.M.P. Couchman, P.A. Thomas, F.R. Pearce, Hydra: an adaptive-mesh implementation of P 3M-SPH, Astrophys. J. 452 (1995) 797.

[50] J.F. Navarro, C.S. Frenk, S.D.M. White, A universal density profile from hierarchical clustering, Astrophys. J. 490 (1997) 493.