Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **454,** 1012–1037 (2015) doi:10.1093/mnras/stv1896

EMMA: an adaptive mesh refinement cosmological simulation code with radiative transfer

Dominique Aubert,* Nicolas Deparis and Pierre Ocvirk

Observatoire Astronomique de Strasbourg, CNRS UMR 7550, Universite de Strasbourg, Strasbourg, France

Accepted 2015 August 13. Received 2015 August 13; in original form 2015 April 13

ABSTRACT

EMMA is a cosmological simulation code aimed at investigating the reionization epoch. It handles simultaneously collisionless and gas dynamics, as well as radiative transfer physics using a moment-based description with the M1 approximation. Field quantities are stored and computed on an adaptive three-dimensional mesh and the spatial resolution can be dynamically modified based on physically motivated criteria. Physical processes can be coupled at all spatial and temporal scales. We also introduce a new and optional approximation to handle radiation: the light is transported at the resolution of the non-refined grid and only once the dynamics has been fully updated, whereas thermo-chemical processes are still tracked on the refined elements. Such an approximation reduces the overheads induced by the treatment of radiation physics. A suite of standard tests are presented and passed by EMMA, providing a validation for its future use in studies of the reionization epoch. The code is parallel and is able to use graphics processing units (GPUs) to accelerate hydrodynamics and radiative transfer calculations. Depending on the optimizations and the compilers used to generate the CPU reference, global GPU acceleration factors between $\times 3.9$ and $\times 16.9$ can be obtained. Vectorization and transfer operations currently prevent better GPU performance and we expect that future optimizations and hardware evolution will lead to greater accelerations.

Key words: methods: numerical – dark ages, reionization, first stars.

1 INTRODUCTION

Starting in the 1970s, numerical simulations became part of the astrophysicist's toolkit to investigate regimes where non-linearities, strong coupling between different physics and multi-scale processes are dominant. The study of structure formation in a cosmological context is an archetypal example of such intricacies and has thus driven the rise and the growth of the so-called cosmological simulations for decades now. These simulations have successively included collisionless dynamics and hydrodynamics, and routinely implement sub-resolution physics, such as star formation or feedback from supernovae, to model the galaxy formation process.

Recently, the challenges encountered in galaxy formation theory and the near advent of new observatories capable of investigating the Universe at redshifts $z > 6$ (such as the *James Webb Space Telescope* and the Square Kilometre Array) have produced interest in the study of the reionization epoch. Interesting in its own sake, as a great cosmic transition produced by the first sources and capable of reionizing the Universe, this epoch will be studied in situ and will provide insights on the initial stages of the structure formation process (see, e.g., reviews by Barkana & Loeb 2001; Pritchard & Loeb 2012). Reionization is also thought to provide keys to understanding the current state of galaxies, with the rise of a UV background capable of suppressing star formation in light objects (see, e.g., Gnedin 2000; Hoeft et al. 2006; Finlator, Davé & Özel 2011; Wise et al. 2012). Consequently, numerical simulations of the reionization started to appear 15 years ago to assess this process and prepare for the advent of observational constraints (see Trac & Gnedin 2011 for a review).

The common feature of such simulations is the inclusion of radiative transfer (RT) physics, to model the impact of the UV photons emitted by the first sources. Quite demanding in terms of computing resources, RT is often included as a post-processing step on outputs of simulations to reduce this cost: absorbents are static and ad hoc recipes are included to model the negative feedback of radiation on sources (see, e.g., Ciardi, Stoehr & White 2003; Iliev et al. 2006; Mellema et al. 2006; McQuinn et al. 2007; Zahn et al. 2007; Baek et al. 2010; Chardin, Aubert & Ocvirk 2012; Paardekooper, Khochfar & Dalla Vecchia 2013; Ocvirk et al. 2013, 2014; Zawada et al. 2014). Nevertheless, radiative hydrodynamics codes started to be implemented (like recently, e.g., Finlator et al. 2011; Rosdahl et al. 2013; Pawlik, Schaye & Dalla Vecchia 2015). They cope with the reduced timescales (and therefore increased CPU cost) produced by radiation physics using massive parallelism (see, e.g., Trac & Cen 2007; Norman et al. 2015), methodological solutions (such as

* E-mail: dominique.aubert@astro.unistra.fr

implicit solvers like, e.g., González, Audit & Huynh 2007; Norman et al. 2015) or specific physical regimes (such as a reduced speed of light like, e.g., Rosdahl et al. 2013; Gnedin 2014). Whatever will be the solution, radiation will surely become an additional standard component of future cosmological simulation codes. This evolution is greatly illustrated by the number of participating codes in comparison projects such as Iliev et al. (2006) and Iliev et al. (2009).

In this context, a new cosmological simulation code is presented here, EMMA,[1] which includes collisionless physics, hydrodynamics and RT. It builds upon the experience gathered with previous codes such as ATON (Aubert & Teyssier 2008, 2010) and a particle-mesh *N*-body only code (Aubert, Amini & David 2009). The aim is to tackle structure formation experiments with a special emphasis on the influence of radiation during the reionization epoch. EMMA is an adaptive mesh refinement code written in C. It runs in parallel with graphics processing unit (GPU)-driven acceleration on a selection of its sub-modules, mainly the physics engines. As such, it shares a number of features with ATON, extending its field of application to coupled radiative hydrodynamics at high resolution through mesh refinement.

The full methodology of EMMA is first described: details on how data are managed, how the physics is solved and how the parallelization is implemented are presented. Then we present a selection of validation tests, from pure dark matter (DM) experiments to more complex situations with coupled physics. Finally, we discuss the performance and the potential future developments of EMMA.

2 METHODOLOGY

2.1 Adaptive mesh implementation

EMMA relies on a grid-based description of the physical quantities and implements adaptive mesh refinement (AMR) techniques to increase dynamically its spatial resolution. The latter is typically of the order of a few cells, for all the different physics presented here, whether it is set by the smoothing of the gravitational force or the smearing of hydrodynamic shocks or ionization fronts. AMR permits us to increase this resolution with a moderate cost in terms of memory consumption by providing finer meshes only at selected locations.

Several AMR implementations exist and EMMA adopts a fully threaded tree description (Khokhlov 1998) of the data, sharing this core feature with other codes such as ART (Kravtsov, Klypin & Khokhlov 1997) and RAMSES (Teyssier 2002). In this framework, a fundamental grid divides a three-dimensional (3D) space (usually cubic but not necessarily) into $2^{3\ell_c}$ cells where ℓ_c designates its refinement level and where the cells are arranged in a Cartesian manner. For instance $\ell_c = 7$ corresponds to a fundamental 128^3 grid where each direction is sampled along 128 points. Hereafter, this fundamental coarsest grid will be referred to as the *base* grid and ℓ_c is the *base* level.

These base level cells are the roots of oct trees that fully describe the refined geometry of space. If a cell is refined, it points towards an octal structure (or *oct*) that in turns points towards eight additional cells belonging to the $\ell_c + 1$ level. This hierarchy can be recursively maintained until a satisfying resolution (i.e., refinement level) is achieved. Conversely, base level cells can also be grouped into octs that belong to coarser cells of level $\ell_c - 1$. Recursively, this process can be repeated until a single $\ell = 0$ cell is obtained. This specific cell is therefore the root of the global AMR structure that samples the 3D space where the numerical experiments take place.

In practice, the data are organized using octs as the fundamental structures where a level ℓ oct O^ℓ contains the following information:

(i) the level ℓ
(ii) eight cells c_i^ℓ with $i \in [0, 7]$
(iii) a pointer towards its parent cell $c^{\ell-1}$
(iv) six pointers towards the six Cartesian neighbouring cells of $c^{\ell-1}$
(v) two pointers towards the next and the previous octs in memory that belong to the same level ℓ. Note that the arrangement of octs in memory is not related to the spatial distribution of the octs in the computational domain. With this feature, octs are stored as a *doubly linked list*.

A cell c_i^ℓ contains the following information:

(i) its index $i \in [0, 7]$
(ii) the physical data at this location (density, potential, pressure, ionized fraction, etc.)
(iii) a pointer towards an oct $O^{\ell+1}$ if it is refined.

Octs are mainly used for data exploration and management. They are the mandatory intermediates in probing the neighbours of a given cell. They are also the intermediate structure to scan all the cells at a given ℓ (i.e., at a given spatial resolution), as they store the pointers towards the previous and next members of the set of all the O^ℓ. These scanning operations are performed through the doubly linked list.

Cells are mainly used to store physical data. They also store a pointer towards an oct if they are refined but that is all the relational information that they possess. Any query on a neighbour cell must be passed through the parent oct. This description reduces greatly (by a factor of 8, typically) the number of relational pointers that would be required if each cell possessed its own set of neighbours: such an organization takes advantage of the fact that all the cells of an oct share a significant fraction of neighbours, at the cost of some operations to retrieve them via the octs.

Since geometrical relations between neighbours are explicit, the mesh can be arbitrarily refined without any constraint on the geometrical strategy of this refinement. Also there is no simple mapping between the position of an oct in memory and its geometrical location. As a consequence, the fully threaded tree can be extremely versatile and follow closely any physical feature that requires high resolution at the cost of storing additional information.

In refined grids, resolution jumps require special care. In particular, these interfaces are a source of errors and inaccuracies that must be controlled as much as possible. One standard requirement that must be enforced in the current description of AMR data is that resolution jumps cannot be greater than unity for two neighbouring cells. Hence, the six neighbouring pointers of an oct necessarily exist. However, it also puts constraints on the way cells are refined and the procedure takes place as follows, in this order:

(i) An oct containing a cell marked for refinement or already refined must be maintained to ensure that resolution jumps are smaller than two. Its parent cell is therefore marked for refinement.
(ii) If a cell is marked for refinement, so must its 26 neighbours. This ensures a smooth transition between low- and high-resolution regions by creating a buffer of high-resolution cells at the interface and also prevents large-resolution jumps.
(iii) If a cell satisfies some user-defined criterion, it is marked for refinement. This criterion can be physically motivated (based

[1] *Electromagnetisme et Mecanique sur Maille Adaptative*

on physical values or gradients) or based on location (for a zoom simulation for instance).

In practice, step (i) is applied to all the cells that belong to a given level in a single pass. Afterwards the step (ii) criterion is likewise applied to all the cells of that level, then step (iii), resulting in three successive passes on all the cells of a given level. The marking procedure can be repeated an arbitrary number of times, especially to increase the thickness of the buffer regions at the transition between regions at two different resolutions. In practice, this procedure is applied twice, similarly to RAMSES or ART. This ensures that a cell that is refined on a physical basis (criterion 3 above) will be produced with an outer layer of two neighbouring cells at the same resolution. Once this marking has been completed, cells can be refined by creating a new oct to be attached to it or coarsened by suppressing its child oct. When refined, all the relations must be computed while the physical quantities are straightforwardly injected from coarse values.

2.2 Solvers

EMMA tracks the evolution of three fluids coupled to each other. The first is a collisionless fluid, sampled by particles in a standard particle-in-cell description. It aims to model the dynamics of stars and non-baryonic DM. Second, the code solves the Euler equations to describe the gas dynamics. We chose to follow RAMSES by using a piecewise linear method à la MUSCL[2]-Hancock driven by HLLC[3] Riemann solvers. Third, EMMA models the propagation of radiation using a moment description of the RT equation with the M1 closure relation. Preliminary atomic processes are also included to describe the cooling and ionization that take place on top of the adiabatic evolution of the gas. Finally, EMMA deals with cosmological settings through super-comoving coordinates, which are briefly discussed.

2.2.1 Collisionless dynamics

Collisionless dynamics are handled through a Monte Carlo sampling of phase space using particles [see, e.g., Hockney & Eastwood (1981) for a reference]. Each particle has a data structure that contains:

(i) its fundamental properties: mass, 3D position $\boldsymbol{r} = x, y, z$ and velocity $\boldsymbol{v} = v_x, v_y, v_z$
(ii) the level of the cell it belongs to
(iii) two pointers towards the next and the previous particles that belong to the same cell.

Additionally, if collisionless dynamics is included, a non-refined cell contains a pointer towards the first particle it contains. If a cell is split, the particles of the coarse cell are passed to the newly created cells. If an oct is destroyed, all the particles are assigned to the parent cell. Using linked lists, the code can therefore scan all the particles of a given cell.

Each particle is evolved in phase space thanks to the usual Newton's equations:

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} = -\nabla\phi(\boldsymbol{r}), \tag{1}$$

$$\Delta\phi(\boldsymbol{r}) = 4\pi G\rho(\boldsymbol{r}), \tag{2}$$

where $\rho(\boldsymbol{r})$ stands for the 3D total matter density and $\phi(\boldsymbol{r})$ is the associated gravitational potential. Updating the phase space coordinates of a particle therefore implies an evaluation of the density and the potential fields on the refined mesh.

The density is computed in all cells of the AMR grid using a standard cloud-in-cell (CIC) interpolation scheme. In practice, it is performed level by level, according to the time stepping procedure described in Section 2.3. If an unsplit cell c_i^ℓ contains particles, they are assigned to cells of the same level ℓ according to the CIC scheme. Furthermore, if a neighbour cell is split, the contribution of the particles in its $\ell + 1$ cells is also taken into account in computing the density of c_i^ℓ. Conversely, if a neighbour cell is coarser and unsplit, its particles will also contribute to the density of c_i^ℓ. In these two situations, the CIC extent of the particles will correspond to the level ℓ resolution, resulting in a CIC density equal to the one that would be obtained from a uniform grid at level ℓ. Finally, if c_i^ℓ is split, its density is computed by averaging the eight values of its child cells.

The density being available, the potential is obtained by solving the Poisson equation. For EMMA, the solution is obtained with relaxation techniques. They have great flexibility (notably compared to Fast Fourier Transform-based approaches) in dealing with complex domain geometries, arbitrary boundary conditions and grids with multiple resolutions. One can use the simple Jacobi iteration formula where the estimation $p + 1$ for the potential in the cell of Cartesian coordinates (i, j, k) can be computed from a previous estimation p via:

$$\phi_{i,j,k}^{p+1} = \frac{\phi_{i+1}^p + \phi_{i-1}^p + \phi_{j+1}^p + \phi_{j-1}^p + \phi_{k+1}^p + \phi_{k-1}^p}{6} + \frac{4\pi G\Delta x^2\rho}{6} \tag{3}$$

The convergence rate of this process is slow, especially to achieve convergence on scales comparable to the total volume. As can be seen in equation (3), information can only be propagated from one cell to the next, making such a formula inefficient at determining the low-frequency modes of the solution. It can be marginally accelerated by over-relaxation or by using the Gauss–Seidel iteration technique, analogous to the Jacobi method but where new estimates of the potential ϕ^{p+1} are used on the right-hand side of equation (3) as soon as they are available, instead of ϕ^p.

Greater acceleration rates can be obtained by using a multi-grid algorithm. The first stage is a smoothing operation where a simple relaxation formula such as equation (3) is used to remove spurious high-frequency features in the current estimation of the potential, i.e., in the regime of scales it is the most efficient. Then this estimate of ϕ^p is coarsened through a restriction operation and applied as a first guess for a new relaxation stage, which is more efficient at low resolution for two reasons: first, the number of sampling points is reduced (typically by a factor of 8 in 3D) and secondly, the convergence of low-frequency modes is increased as their effective scale is reduced relative to the influence radius of equation (3). Once this low-resolution solution is converged, it is interpolated back (also called the prolongation step), and after a few relaxation steps at full resolution a converged solution is obtained.

This two-level process (between ℓ and $\ell - 1$) can be recursively applied to $\ell - 1$ and $\ell - 2$ and so on. In EMMA, we typically proceed until $l = 2$ (i.e., using a 4^3 field) to achieve a 10^{-3} convergence of the solution with only about five iterations at full resolution. We take advantage of the AMR structure to store residuals and perform the calculations on coarse levels. Prolongation is performed via linear

[2] Monotonic Upstream-Centered Scheme for Conservation Laws.
[3] Harten-Lax-van Leer-Contact.

interpolation, whereas restriction is done by averaging on the eight cells of an oct.

For fine levels with $\ell > \ell_c$, the potential is computed by simple relaxation, using a red-black Gauss–Seidel smoother, with boundary conditions provided by the $\ell - 1$ cells that surround high-resolution patches. However, the initial evaluation of the potential is obtained by interpolation of the coarse solution. Therefore, convergence is quickly obtained from this first value that is already close to the correct one. For EMMA, convergence at the 10^{-3} level is obtained after 10 iterations in a typical cosmological simulation. Fine cells at resolution jumps must compute the potential using high-resolution values that do not exist on the coarse side of the jumps. In this case, the potential is interpolated at the requested location from its coarse value.

The potential being available, the acceleration field $\boldsymbol{f} = -\nabla\phi$ is computed by simple derivation and is interpolated back at the particle positions again according to the CIC scheme. If a particle lies close to the interface between two regions at different levels ℓ and $\ell - 1$, the force field is interpolated from the coarsest $\ell - 1$ level. As such, it implies that a particle must penetrate significantly within high-resolution regions to be sensitive to their force field, otherwise they will be driven by a lower-resolution description of the potential.

Particles are advanced thanks to a mid-point scheme. The $p + 1$ value of the position and the velocities are computed from their p value as:

$$\boldsymbol{r}_{p+1} = \boldsymbol{r}_{\rm p} + \boldsymbol{v}_{p+1/2}\Delta t^p \tag{4}$$

where the mid-point velocity is provided by

$$\boldsymbol{v}_{p+1/2} = \boldsymbol{v}_{\rm p} + \boldsymbol{f}_{\rm p}\frac{\Delta t^p}{2} \tag{5}$$

and finally corrected as:

$$\boldsymbol{v}_{p+1} = \boldsymbol{v}_{p+1/2} + \boldsymbol{f}_{p+1}\frac{\Delta t^p}{2} \tag{6}$$

2.2.2 *Hydrodynamics*

The hydrodynamics is solved through an Eulerian description of conserved fluid quantities, which obey the set of Euler equations (see, e.g., Toro 1997):

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}(\boldsymbol{U})}{\partial x} + \frac{\partial \boldsymbol{G}(\boldsymbol{U})}{\partial y} + \frac{\partial \boldsymbol{H}(\boldsymbol{U})}{\partial z} = \boldsymbol{S}, \tag{7}$$

where $\boldsymbol{U} = (\rho, \rho u, \rho v, \rho w, E)$ is the array of conserved quantities: the density, the three components of momentum and the energy. $\boldsymbol{F}$, $\boldsymbol{G}$ and $\boldsymbol{H}$ are the three flux functions with

$$\boldsymbol{F}(\boldsymbol{U}) = (\rho u, \rho u^2 + p, \rho uv, \rho uw, u(E + p)) \tag{8}$$

$$\boldsymbol{G}(\boldsymbol{U}) = (\rho v, \rho uv, \rho v^2 + p, \rho vw, v(E + p)) \tag{9}$$

$$\boldsymbol{H}(\boldsymbol{U}) = (\rho w, \rho uw, \rho vw, \rho w^2 + p, w(E + p)) \tag{10}$$

In association with conserved quantities, it is usual also to consider the primitive quantities, namely the density ρ, the velocities $\boldsymbol{v} = (u, v, w)$ and the pressure p. The total energy is given by

$$E = \frac{\rho}{2}(\boldsymbol{v}^2) + \frac{p}{\gamma - 1}, \tag{11}$$

with contributions for the kinetic and the internal energies. Here γ is the usual adiabatic exponent, equal to 5/3 for an ideal monoatomic gas. The high-Mach flow regime is taken into account using the recipe described in Rasera & Teyssier (2006).

The set of Euler equations is solved in a split fashion, dealing first with the pure transport part with a null right-hand side in equation (7) then updating the solution by adding the contribution of source terms. The transport update of $\boldsymbol{U}^p$ into $\boldsymbol{U}^{p+1}$ is done explicitly (in 1D here for simplicity) via:

$$\boldsymbol{U}_i^{p+1} = \boldsymbol{U}_i^p + \frac{\Delta t}{\Delta x}\left(\mathcal{F}_{i-1/2}^p - \mathcal{F}_{i+1/2}^p\right). \tag{12}$$

This solution requires a knowledge of intercell fluxes $\mathcal{F}_{i\pm1/2}^p$ at instant p between cells i and $i \pm 1/2$, through the resolution of Riemann problems. Here we use a MUSCL scheme coupled to a HLLC Riemann solver (see, e.g., Toro, Spruce & Speares 1994; Toro 1997). First, conserved quantities are linearly reconstructed at the cell boundaries (in 1D for simplicity):

$$\boldsymbol{U}_i^{\rm L/R} = \boldsymbol{U}_i^p \pm \frac{\Delta_i}{2}. \tag{13}$$

Here L/R designates the left and right reconstructed states at the cell i boundaries at $x = 0$ and $x = \Delta x$, the cell centre being in $\Delta x/2$. Δ_i is the slope vector of the conserved quantities and is computed using neighbouring cell values and a MinMod limiter to ensure a monotonic solution. Before solving the Riemann problem, the boundary extrapolated values are also evolved by a time $\Delta t/2$:

$$\bar{\boldsymbol{U}}_i^{\rm L/R} = \boldsymbol{U}_i^{\rm L/R} + \frac{\Delta t}{2\Delta x}\left[\boldsymbol{F}\left(\boldsymbol{U}_i^{\rm L}\right) - \boldsymbol{F}\left(\boldsymbol{U}_i^{\rm R}\right)\right]. \tag{14}$$

The Riemann problem at intercell positions is solved using these evolved states, for instance $\mathcal{F}_{i+1/2}^p$ is obtained from states $\bar{\boldsymbol{U}}_i^{\rm R}$ and $\bar{\boldsymbol{U}}_{i+1}^{\rm L}$. The MUSCL scheme achieves second-order accuracy (Toro et al. 1994; Toro 1997). As discussed previously, when dealing with cells at the interface between two regions with different resolutions, coarse quantities are interpolated in a conservative manner at the locations of the virtual fine cells. At such interfaces, the low-resolution values are assumed to be constant in time and accuracy reduces to first order (Khokhlov 1998; Teyssier 2002). The multi-dimensionality of the problem is dealt with through an unsplit approach with a flux contribution from the three directions included at once in the update.

The stability of the solution is obtained by enforcing the Courant condition on the time step

$$\Delta t = C\frac{\Delta x}{3V_h}, \tag{15}$$

where V_h is an estimate of the largest wave speed present throughout the computational domain and $C < 1$. We follow the simple suggestion of Toro with

$$V_h = \max\{|\boldsymbol{v}|_i + a_i\} \tag{16}$$

where a_i stands for the sound speed at location i. It usually overestimates the limiting velocity and therefore underestimates the corresponding time step, but provides a robust estimation.

The source term in equation (7) models deviation from pure conservation. In our case, $\boldsymbol{S}$ holds the contribution of the gravitational force to momentum and energy variation and expresses the coupling of gravitation with hydrodynamics:

$$\boldsymbol{S} = (0, -\rho\vec{\nabla}\phi, -\rho\boldsymbol{v}\vec{\nabla}\phi), \tag{17}$$

where ϕ is the gravitational potential. Its inclusion in the solution [equation (12)] is done in an explicit manner with

$$\boldsymbol{S} = (0, -\rho^p\vec{\nabla}\phi^p, -\rho^p\boldsymbol{v}^p\vec{\nabla}\phi^p). \tag{18}$$

This contribution modifies the conservative quantities after the pure transport update. However, they are taken into account within the MUSCL scheme to compute the interpolated left/right states before solving the Riemann problems. Of course, the hydrodynamics is also coupled to radiative physics and thermo-chemistry via the internal energy (or equivalently the pressure) of the gas: photo-heating and cooling act as source and sink terms of this quantity and are treated by the RT engine. Finally, the hydro-engine is able to handle passive scalars that are being advected with the fluid. Among such scalars, currently there is the hydrogen neutral fraction and in forthcoming developments one can think of the metallicity.

2.2.3 Radiative transfer

Propagation of radiation is dealt with using a moment-based description: light is described as a fluid, where its phase space distribution is averaged over velocities to focus on spatial fields. Among this family of radiation descriptions, EMMA relies on the M1 approximation (Levermore 1984; González et al. 2007; Aubert & Teyssier 2008).

Taking the first two moments of the Liouville equation leads to the conservation equations of the ionizing photon density $N_\nu(\boldsymbol{r}, t)$ and flux $\boldsymbol{F}_\nu(\boldsymbol{r}, t)$, in each bin of frequency ν:

$$\frac{\partial N_\nu}{\partial t} + \frac{\partial \boldsymbol{F}_\nu}{\partial \boldsymbol{r}} = S_\nu - \kappa_N N_\nu, \tag{19}$$

$$\frac{\partial \boldsymbol{F}_\nu}{\partial t} + c^2 \frac{\partial \boldsymbol{P}_\nu}{\partial \boldsymbol{r}} = -\kappa_F \boldsymbol{F}_\nu. \tag{20}$$

Here, $\boldsymbol{P}_\nu(\boldsymbol{r}, t)$ stands for the radiative pressure tensor. This system of equations being open, a closure relation is required to solve it. The M1 closure relation is given by:

$$\boldsymbol{P}_\nu = \boldsymbol{D}_\nu N_\nu \tag{21}$$

$$\boldsymbol{D}_\nu = \frac{3\chi - 1}{2}\boldsymbol{I} + \frac{1-\chi}{2}\boldsymbol{n} \times \boldsymbol{n}. \tag{22}$$

Here, $\boldsymbol{D}_\nu$ stands for the Eddington tensor and its expression is set by the value of the χ quantity that varies between $1/3$ for a diffusive regime (i.e., $F_\nu \ll cN_\nu$) and 1 for a pure transport regime (i.e., $F_\nu \sim cN_\nu$).

The quantities on the right-hand side of the conservation equations are the source of photons $S_\nu(\boldsymbol{r}, t)$ (expressed in photons per unit time per unit volume) and the two absorption terms, driven by the absorption coefficients κ_N and κ_F, considered equal in EMMA with $\kappa_N = c\sigma_\nu n_{\rm H}$. These terms couple the hydrodynamics and the RT with gas absorption with $n_{\rm H}$ being the density number of neutral gas and σ_ν the photo-ionization cross-section at frequency ν.

For multi-frequency transfer, photons are gathered in so-called groups of frequencies and the flux and number densities of a given group satisfy the above conservative equations. In practice, equations (19) and (20) can be integrated between two frequencies:

$$\frac{\partial N}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{r}} = S - c\sigma_N n_{\rm H} N, \tag{23}$$

$$\frac{\partial \boldsymbol{F}}{\partial t} + c^2 \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{r}} = -c\sigma_N n_{\rm H} \boldsymbol{F}, \tag{24}$$

with

$$N = \int_{\nu_1}^{\nu_2} N_\nu \mathrm{d}\nu \tag{25}$$

$$\boldsymbol{F} = \int_{\nu_1}^{\nu_2} \boldsymbol{F}_\nu \mathrm{d}\nu \tag{26}$$

$$S = \int_{\nu_1}^{\nu_2} S_\nu \mathrm{d}\nu \tag{27}$$

$$\sigma_N = \frac{1}{N}\int_{\nu_1}^{\nu_2} \sigma_\nu N_\nu \mathrm{d}\nu. \tag{28}$$

Typical frequency groups have limits set by the ionization levels of hydrogen and helium (even though EMMA does not currently handle helium chemistry), i.e., [13.6, 24.6, 54.4] eV or are chosen to represent broad classes of different types of radiation such as UV, X- and hard X-rays.

In practice, the updating of the radiative quantities is a two-stage process: first, a conservative transport is performed, then non-conservative contributions (source and sinks) are added within a subsequent thermo-chemical solver (see Section 2.2.4). The set of homogeneous (with zero right-hand side) coupled equations is solved for $\boldsymbol{U} = (N, \boldsymbol{F})$ using fluxes $\tilde{F} = (\boldsymbol{F}, c^2\boldsymbol{P})$ with a simple explicit finite difference scheme (in 1D for simplicity):

$$\boldsymbol{U}_i^{p+1} = \boldsymbol{U}_i^p + \frac{\Delta t}{\Delta x}\left(\mathcal{F}_{i-1/2}^p - \mathcal{F}_{i+1/2}^p\right), \tag{29}$$

which must satisfy the usual Courant condition:

$$c \leq \frac{\Delta x}{\Delta t}. \tag{30}$$

Here $\mathcal{F}_{i-1/2}^p(\boldsymbol{U})$ represents the flux function at instant p measured at the interface between cells i and $i-1$. This intercell flux is obtained by solving a typical Riemann problem at this interface: in EMMA, this flux is given by the Lax–Friedrich formula:

$$\mathcal{F}_{i+1/2}(\boldsymbol{U}) = \frac{\tilde{F}_i + \tilde{F}_{i+1}}{2} - c\frac{\boldsymbol{U}_{i+1} - \boldsymbol{U}_i}{2}. \tag{31}$$

As in ATON, an implementation of the less diffusive HLL flux is also on the way. At this stage, conservative transport is completed and radiative quantities $(N, \boldsymbol{F})$ are in an intermediate state, waiting for the contribution of source and sinks to be taken in account, as explained in the next section.

First, note that the Courant condition ensures the stability of the scheme but imposes that the numerical sampling velocity $\Delta x/\Delta t$ must be greater than the speed of light. The cost of simplicity provided by the explicit solver is therefore a very fine temporal sampling, greatly enhancing the CPU cost of the RT. This stringent constraint on the time step imposed by the Courant condition can be circumvented in two ways. The first is simply by taking advantage of hardware acceleration (as in ATON). In EMMA, this route is explored with GPUs as accelerating devices and described in Section 4. The second is to reduce the speed of light (Gnedin & Abel 2001; Rosdahl et al. 2013), taking advantage that in a large number of situations, the effective propagation of radiative information is performed through ionization fronts that propagate at smaller pace. This option can also be set in EMMA, as done for instance in Section 6.6.1.

2.2.4 Thermal and chemical processes

The thermal and chemical processes encompass the atomic physics that will affect the hydrodynamics and radiative quantities. Currently, EMMA handles only atomic hydrogen processes. They contribute to changing the number density and flux of photons (source and sinks), the ionization state of the gas and finally its internal energy:

$$\frac{\mathrm{d}N}{\mathrm{d}t} = S - c\sigma_N n_{\rm H} N + (\alpha_A(T) - \alpha_B(T))x^2 n_0^2, \tag{32}$$

$$\frac{\mathrm{d}\boldsymbol{F}}{\mathrm{d}t} = -c\sigma_N n_{\rm H}\boldsymbol{F}, \tag{33}$$

$$\frac{\mathrm{d}n_{\rm H}}{\mathrm{d}t} = (\alpha_A(T)x^2 - \beta(T)x(1-x))n_0^2 - cn_{\rm H}\sigma_N N, \tag{34}$$

$$\frac{\mathrm{d}e}{\mathrm{d}t} = cn_{\rm H}\Sigma_E N - \Lambda(n_0, x, T), \tag{35}$$

where $n_{\rm H} = (1-x)\rho/m_{\rm p} = (1-x)n_0$ is the number density of hydrogen atoms (with individual mass $m_{\rm p}$), x being their ionized fraction and n_0 being the total number of protons (i.e., neutral plus ionized hydrogen). The quantities $\alpha_{A/B}$ and β are, respectively, the case A/B recombination and collisional ionization rates. Equations (32) and (33) describe the influence of source and sinks on the radiative quantities: in particular the last term in equation (32) refers to the recombining ionizing radiation. Note that the on-the-spot approximation can easily be applied by setting $\alpha_A = \alpha_B$ in equations (32) and (34). Equation (34) details the competing effects of recombination and ionizations on the number of neutral hydrogen atoms. Equation (35) encodes the evolution of the internal energy density of the gas $e = p/(\gamma - 1)$ due to atomic cooling (given by the cooling rate Λ) and to the photo-heating above the ionization threshold $\mathcal{H} = cn_{\rm H}N\Sigma_E$. Here $\Sigma_E = (\sigma_E\langle E\rangle - \sigma_N E_{13.6})$ where the quantity σ_E is the energy averaged cross-section over the group of frequencies of interest:

$$\sigma_E = \frac{1}{N\langle E\rangle}\int_{\nu_1}^{\nu_2}\sigma_\nu N_\nu h\nu \mathrm{d}\nu, \tag{36}$$

$$\langle E\rangle = \frac{1}{N}\int_{\nu_1}^{\nu_2} N_\nu h\nu \mathrm{d}\nu \tag{37}$$

where the latter quantity $\langle E\rangle$ is the average photon energy in the same group.

For multi-frequency transfer, equations (32)–(35) are solved for each frequency interval (i.e., group) $[\nu_i, \nu_{i+1}]$ where i stands for a group label. Within each frequency group i, the corresponding photon density and fluxes $(N_i, \boldsymbol{F}_i)$ satisfy:

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = S_i - c\sigma_{N,i}n_{\rm H}N_i + (\alpha_A(T) - \alpha_B(T))\delta_{i,1}x^2 n_0^2 \tag{38}$$

$$\frac{\mathrm{d}\boldsymbol{F}_i}{\mathrm{d}t} = -c\sigma_{N,i}n_{\rm H}\boldsymbol{F}_i. \tag{39}$$

These equations depend on the group cross-section $\sigma_{N,i}$ and source function S_i obtained using equations (27) and (28) on the $[\nu_i, \nu_{i+1}]$ interval. The number of radiative conservative updates therefore scales as the number of frequency groups. The Kronecker symbol $\delta_{i,1}$ in equation (38) implies that the recombining radiation only contributes to the first group of ionizing photons, the closest to the hydrogen ionization frequency (see, e.g., Rosdahl et al. 2013). Again, if the on-the-spot approximation is used, this contribution is set to zero for all the groups. The thermo-chemical equations (34) and (35) are also modified and the following expressions must be replaced:

$$cn_{\rm H}\sigma_N N \rightarrow cn_{\rm H}\sum_{i=1}^{\rm Ngroups}\sigma_{N,i}N_i, \tag{40}$$

$$cn_{\rm H}\Sigma_E N \rightarrow cn_{\rm H}\sum_{i=1}^{\rm Ngroups}\Sigma_{E,i}N_i. \tag{41}$$

Ngroups stands for the total number of frequency intervals considered. Equations (40) and (41) depend on the group cross-sections and photon densities and effectively couple photons of different frequencies that are otherwise transported without any interactions.

In practice, we solve the equations in the same order as above by sub-cycling the dynamical step Δt with a chemical time step $\Delta\tau$. During the latter, all quantities are updated from state p to $p+1$. $\Delta\tau$ is first evaluated from the p state of the internal energy, with $\delta\tau = e^p/(\mathcal{H}^p - \Lambda^p)$. Then all the quantities are updated in the following manner and in this order:

$$N^{p+1} = \frac{N^p + \left(S + (\alpha_A(T) - \alpha_B(T))x^2 n_0^2\right)\Delta\tau}{1 + c\sigma_N(1-x^p)n_0} \tag{42}$$

$$\boldsymbol{F}^{p+1} = \frac{\boldsymbol{F}^p}{1 + c\sigma_N(1-x^p)n_0} \tag{43}$$

$$x^{p+1} = 1 - \frac{\alpha_A(T^p)x^{p2}n_0\Delta\tau + 1 - x^p}{1 + \Delta\tau(\beta(T^p)x^p n_0 + c\sigma_N N^{p+1})} \tag{44}$$

$$\begin{aligned} e^{p+1} = {} & e^p + \Delta\tau c(1-x^{p+1})n_0 N^{p+1}\Sigma_E \\ & - \Delta\tau\Lambda(n_0, x^{p+1}, T^p)). \end{aligned} \tag{45}$$

In this sequence, new information on a given quantity is immediately used to compute a subsequent one. We follow Rosdahl et al. (2013) and enforce that the internal energy must at most vary by 10 per cent (relatively) during $\Delta\tau$, otherwise the set of equations is recomputed with a time step divided by 2 until this condition is satisfied. In all our experiments, this procedure has been found to be accurate and robust enough. All the rates required to describe the atomic processes, such as the recombination, the collisional ionization and the cooling, are taken from the compilation of Theuns et al. (1998). Photo-ionization cross-sections are taken from Hui & Gnedin (1997).

2.3 Time stepping

The organization of a time step is intimately constrained by the multi-level structure of the data. A single-level time step is organized in quite a usual fashion and described in Fig. 1. Coupling between the different physics occurs at different levels, the most explicit ones being:

(i) between gravity and hydrodynamics through the total matter density and the gravitational force it creates
(ii) between radiation and hydrodynamics through the density distribution or the gas temperature.

On top of this explicit coupling, detailed in the next section, implicit ones also occur where, e.g., a photo-evaporating halo could in principle affect the underlying DM distribution (in the same way that supernova feedback could affect it, even though it is not explicitly coupled, e.g., Pontzen & Governato 2012).

When mesh refinement is enabled, coupling between levels must be taken into account with special care. In EMMA, updates are performed level by level, each level being updated with its own time step Δt_ℓ. Typically, $\Delta t_\ell \sim \Delta t_{\ell+1} \times 2$ (ℓ corresponding to a level coarser than $\ell + 1$). Advancing the solution on a level ℓ can be expressed as the following recursive expression:

$$A_\ell = R_{\ell+1}P_\ell A_{\ell+1}U_\ell M_\ell, \tag{46}$$

where A_ℓ stands for advancing the solution at level ℓ. R_ℓ is the refinement operator (i.e., creating/destroying $\ell + 1$ octs from ℓ cells). P_l and U_l are the operators for the Poisson resolution and the

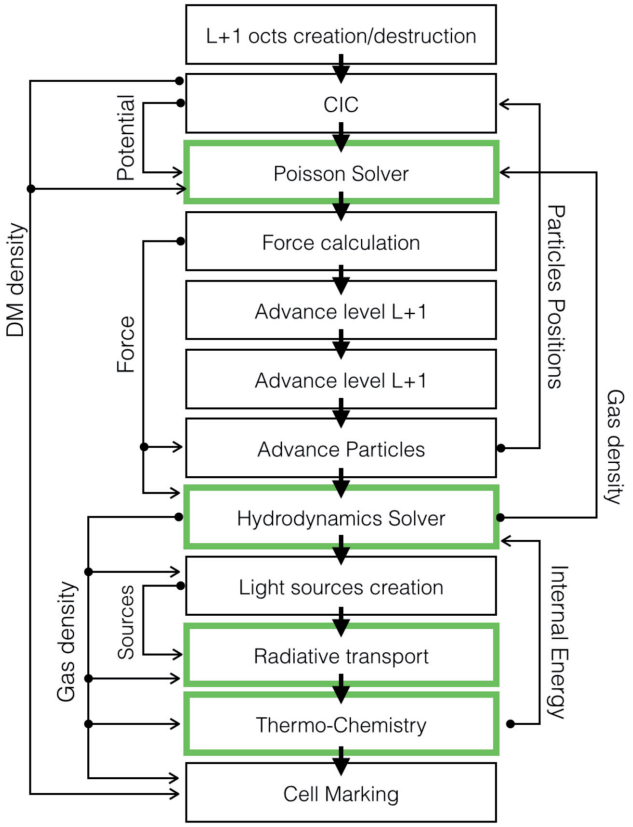

Figure 1. Flow of sequence for a time step at a given level (from top to bottom). Side arrows describe the exchange of physical quantities between different modules to emphasize the most important couplings. Green boxes stand for modules that have been ported onto the GPU using CUDA. It is assumed that $\Delta t_\ell = 2\Delta t_{\ell+1}$ and the $\ell + 1$ level is advanced twice. This sequence is repeated recursively for all the finer levels.

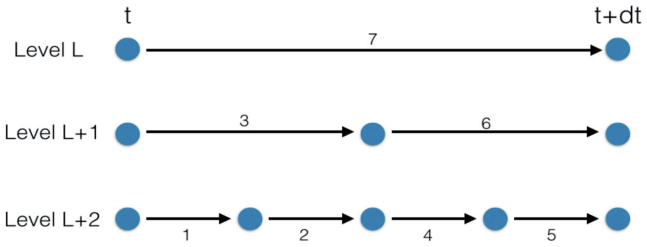

Figure 2. Time-step hierarchy of three levels of the AMR structure. The coarse level ℓ is advanced from t to $t + \mathrm{d}t$, which implies partial advance and temporary updates of fine levels ℓ+1 and ℓ+2 until they are synchronized with ℓ. Arrows and numbers indicate the sequence of partial updates at different levels to perform this full update of the three nested levels.

update (i.e., moving particles and updating Eulerian fields) at level ℓ. Finally, M_ℓ corresponds to the marking of the level ℓ cells for future refinements. The recursion is stopped at the maximal allowed level with $R_{\ell\max} = M_{\ell\max} = A_{\ell\max+1} = 1$. For instance, a simulation with three resolution levels (e.g., $\ell = 5, 6, 7$) will be fully updated on Δt_5 according to the following operations (Fig. 2 details this step in a schematic manner):

$$A_5 = R_5 P_5 [R_6 P_6 [A_7] U_6 M_6] [R_6 P_6 [A_7] U_6 M_6] U_5 M_5$$
$$A_7 = P_7 U_7 P_7 U_7 \quad (47)$$

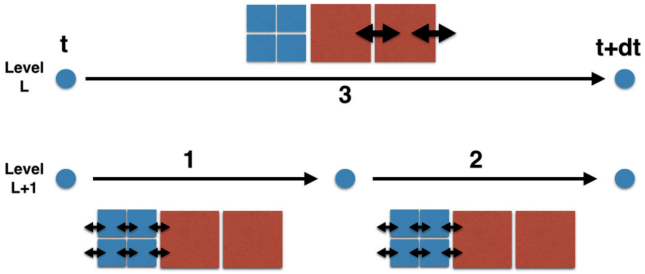

Figure 3. Multi-level update of fluxes at the interface between two levels. To perform an update from t to $t + \mathrm{d}t$, fine level $\ell + 1$ must perform two steps sub-cycled within a large step of the coarse level ℓ. Updates are performed according to the sequence 1, 2, 3: note how the middle coarse cell is updated during the three steps with fine fluxes first and a final coarse flux, during the level ℓ update.

where we assumed that $\Delta t_5 = 2\Delta t_6 = 4\Delta t_7$. More generally, the time steps are constrained by:

$$\Delta t^1_{\ell+1} + \Delta t^2_{\ell+1} \leq \Delta t_\ell. \quad (48)$$

Once the level $\ell + 1$ has been updated by $\Delta t^1_{\ell+1} + \Delta t^2_{\ell+1}$, the value Δt_ℓ is updated to synchronize the coarse level on the finer one.

As originally described by Khokhlov (1998), this nested hierarchy of time steps has some strong implications for the flux update at interfaces between two levels. When conservative quantities are updated on level $\ell + 1$, adjacent ℓ cells must take into account the fluxes produced during the sub-cycling of level $\ell + 1$. For instance, if a conservative quantity U_ℓ and $U_{\ell+1}$ in two adjacent cells must be updated on Δt_ℓ, the following sequence must be obeyed (see also Fig. 3):

(i) $U^p_{\ell+1}$ is updated to $U^{p+1/2}_{\ell+1}$ using a flux $F(U^p_{\ell+1}, U^p_\ell)$. Meanwhile U^{p+1}_ℓ is temporarily updated using the same flux.
(ii) $U^{p+1/2}_{\ell+1}$ is updated to $U^{p+1}_{\ell+1}$ using a flux $F(U^{p+1/2}_{\ell+1}, U^p_\ell)$. Again U^{p+1}_ℓ is temporarily updated using the same flux.
(iii) Finally, U^{p+1}_ℓ is fully updated taking into account the flux created by adjacent cells of level ℓ.

This partial update of coarse cells during fine time steps is one of the reason for enforcing a maximal jump in resolution of unity between two adjacent cells. Larger jumps would create a complex hierarchy of partial updates of coarse quantities, which would be difficult to handle properly.

The amplitude of time steps is set by physical temporal scales that must be resolved to track properly their impact on the evolution of quantities. EMMA being a multi-physics code, a whole list of timescales is computed for all the cells of a given level and the smallest sets its global time step:

(i) We follow Teyssier (2002), and limit the rate of change of the cosmological expansion factor $\delta a(\Delta t_{\mathrm{cosmo}})/a < \epsilon$.
(ii) A particle cannot move on a scale larger than the size of a cell: $\Delta t_{\mathrm{pic}} = \epsilon \Delta x_\ell / v_{\max}$.
(iii) The local dynamical time must be resolved: $\Delta t_{\mathrm{dyn}} = \epsilon/\sqrt{G\rho}$.
(iv) The hydrodynamical Courant condition must be satisfied: $\Delta t_{\mathrm{hyd}} < \Delta x_\ell / V_h$ [see equation (16)].
(v) If light sources are present, the radiation Courant condition must be satisfied: $\Delta t_{\mathrm{rad}} < \Delta x_\ell / c$. It ensures that light propagation is performed in a stable manner.

When full physics are included and effective, the most stringent condition is usually provided by Δt_{rad} and by orders of magnitude

since usually $c \gg V_h$. This dominance can be reduced by setting a reduced speed of light. Furthermore, as non-linearities increase (and even more when strong shocks, such as induced by supernova feedback, are included) the ratio $\Delta t_{\rm hyd}/\Delta t_{\rm rad}$ tends to decrease.

3 COSMOLOGICAL SETTING

Cosmological experiments are implemented using the set of 'super-comoving' variables suggested by Martel & Shapiro (1998). The transformation from physical to super-comoving variables is given by:

$$\tilde{\boldsymbol{r}} = \frac{\boldsymbol{r}}{a r_*}, \tag{49}$$

$$\tilde{\boldsymbol{v}} = \frac{a\boldsymbol{v}}{v_*}, \tag{50}$$

$$\tilde{\rho} = \frac{\rho}{a^3 \rho_*}, \tag{51}$$

$$\tilde{p} = \frac{a^5 p}{p_*}, \tag{52}$$

$$\tilde{\phi} = \frac{a^2 \phi}{\phi_*}, \tag{53}$$

$$\tilde{dt} = \frac{dt}{a^2 t_*}, \tag{54}$$

$$\tilde{N} = a^3 N r_*^3 \tag{55}$$

$$\tilde{\boldsymbol{F}} = a^4 r_*^2 t_* \boldsymbol{F}, \tag{56}$$

where starred quantities stand for normalization units with $r_* = L$ (the box length), $t_* = 2/(H_0\sqrt{\Omega_m})$, $v_* = r_*/t_*$, $\rho_* = 3H_0^2\Omega_m/(8\pi G)$ and $p_* = \rho_* v_*^2$. H_0 and $a(t)$ stand for the usual current Hubble parameter and the time-dependent expansion factor.

With this set of transformations, it can be shown that almost all the differential equations to be solved keep their standard expression for a $\gamma = 5/3$ gas. The only notable exception is the Poisson equation, which becomes:

$$\tilde{\Delta}\tilde{\phi} = 6a\delta, \tag{57}$$

where $\delta = \tilde{\rho}/\langle\tilde{\rho}\rangle - 1$. Still, this equation remains typical of an elliptic equation that can be solved by all the methods already in place for the Newtonian field equation. Overall, the use of such a transformation greatly simplifies the implementation of cosmological settings in this kind of simulation code.

4 PARALLELIZATION AND VECTORIZATION

EMMA is a parallel code, which includes two levels of multi-tasking. The first is the standard multi-CPU mode, where the computational domain is distributed among several processes that communicate with each other via the MPI protocol. The second level of parallelism resides within an MPI process where the local load is distributed among several threads of execution. For EMMA, this local parallelization is performed on GPUs but could in principle be extended to other modes of multi-threading such as local shared-memory parallelism among multiple CPU cores or other hardware accelerators. The second level of parallelism can be understood as a vectorization, where arrays of data are processed in parallel through the same set of instructions with minimal communications.

With this two-level parallelism in mind, EMMA has been designed to decouple as much as possible instructions that deal with the logistics of data from the ones that actually perform calculations. Logistics operations are defined as operating directly on the AMR tree, e.g., cell marking and refinement, tree management and inter-process communications. These operations are handled by CPUs. Computing functions on the other hand expect arrays of data to be crunched, without any mention of tree-organized data or inter-process parallelism and return, likewise, an array of results. The physics solvers belong to this second category and are meant to be processed by vector-based hardware, such as multi-core processors, GPUs or any other kind of co-processor. In between, a set of interface functions must be developed to perform gather/scatter operations from/to the AMR tree to/from the calculations arrays. These aspects are developed in the next sections.

4.1 Distributed parallelism on multiple CPUs

Distributed parallelism is handled through a space-filling curve domain decomposition. Such a curve provides a 1D mapping of a 3D grid by assigning a unique key to each oct as a function of its Cartesian position. The number $n_{\rm p}$ of parallel processes being defined, the curve is split into $n_{\rm p}$ successive parts with equal loads, thus assigning a set of octs to each process. The number $n_{\rm p}$ can thus be arbitrary and the 1D mapping alleviates the need to deal with multiple boundaries along multiple directions. EMMA has been implemented with both a Peano–Hilbert space-filling curve and a slab-based key ordering for problems with unidirectional variations (such as the shock tube or the Zeldovich pancake). Currently, the domain decomposition is performed at the level ℓ_c corresponding to the base resolution of the simulation: all the ℓ_c octs are distributed among the processes, in such a way that each process possesses at least one such oct. All octs created from a level ℓ_c cell are assigned to the same process. At the current stage, EMMA does not perform any kind of load-balancing that could be obtained by sliding the limits of the 1D domains along the space-filling curve to optimize the distribution of work among the processes.

It should be emphasized that the AMR structure can only be fully exploited if it remains consistent: no holes, no level jumps greater than 1 from one cell to another, pointers towards neighbouring cells must exist, etc. Furthermore, a given process should be aware, at least partially, of the AMR tree structure of the neighbour processes. EMMA copes with these issues by employing the local essential tree decomposition (Warren & Salmon 1993; Dubinski 1996): each process, even though it has been assigned only a subset of the total volume, is aware of the whole hierarchy of nested octs but only at the levels relevant to its tasks (see Fig. 4). Neighbour cells directly in contact with its domain bring their whole tree from $\ell = 1$ to $\ell_{\rm max}$, those being second-order neighbours are one level coarser and so on. In practice, this local hierarchy is obtained at the initial building of the oct tree: from the root $\ell = 1$ oct, cells are refined down to the coarse level if they belong to the sub-volume assigned to the current processor or if they are direct neighbours of this sub-volume. This produces naturally a local tree for each processor, individually fully consistent and yet aware of the structure of the direct neighbours.

Communications are handled using the MPI protocol and if a given process P_0 requires data present on other processes, it must be performed explicitly by specifying which octs should come from which other MPI process. This communication protocol in EMMA has been written in the following fashion:

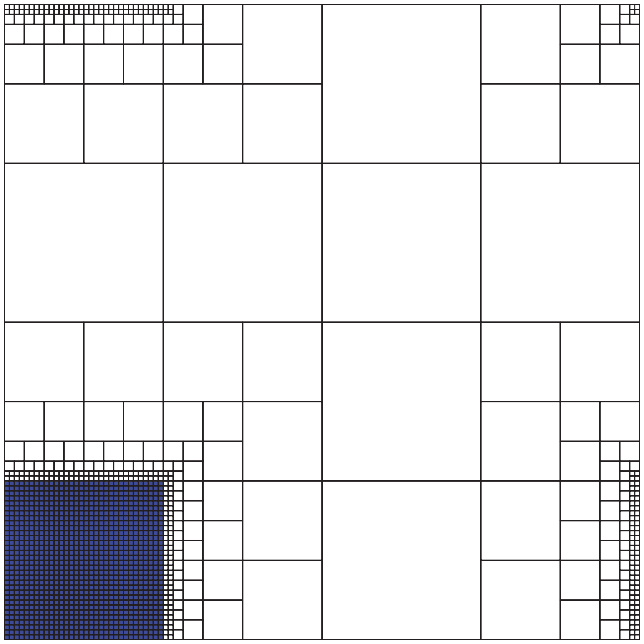

Figure 4. An example of a sub-domain seen by a processor in the essential tree paradigm. The processor has been assigned the lower left corner (shaded) but is aware of the whole computational domain. Distant regions are coarsened relative to close ones.

(i) First, all processes P_i build their own list of neighbour MPI processes $\{P_j\}$ with $i \neq j$.

(ii) For each member of $\{P_j\}$, a list of requests (i.e., of neighbour octs) is established by P_i by storing their space-filling curve keys. This list of keys is sent from each P_i to all its $\{P_j\}$: P_i acts as a client sending requests to neighbour servers.

(iii) Likewise, each P_i receives a list of requests from the same sources: P_i acts as a server to neighbour clients.

(iv) Each client key is processed by P_i through a hash table to relate the absolute key to a local pointer to an oct. The data are gathered and sent back to the clients $\{P_j\}$.

(v) Meanwhile, the data from the servers $\{P_j\}$ are received and scattered back in the local tree by P_i.

This set of instructions is performed level by level and called by the Poisson, the hydrodynamics and the radiation solvers to update border cells that belong to other processes and that have been remotely modified: the flux of information can be considered as outside in. However, there are situations where the information flux is inside out: a process performed locally will directly affect a value outside its domain. The first example of inside-out communication is the CIC assignment, where local particles will contribute to an adjacent domain. The second example is the conservative updates due to hydrodynamics or radiative fluxes between cells at different levels and belonging to different processes. Because this update is asynchronous between levels, conservative values can be updated in a neighbouring coarser cell and must be communicated to its home process. In this case, the protocol is similar to the one described above except that there is no request stage and the data are sent directly from the server to the client.

Note that without load-balancing, the list of neighbours $\{P_j\}$ is static for each P_i. Hence, the first stage of the communication protocol has to be performed only once. However, the list of *neighbour octs* is dynamic, because of mesh refinement. Therefore, for a given pair, client/server P_i/P_j, the list of requests changes on the fly and should typically be recomputed any time octs have been created or destroyed.

4.2 Local vectorization

Local parallelism relies on a vector-based strategy, where arrays of data are processed through the same set of instructions and possibly on architecture with vectorization capability. The driver for this choice of design is the recent emergence of multi-core processors, GPUs or CPU-based co-processors, which all rely on this programming paradigm to be fully efficient. For EMMA, this kind of parallelization focuses on the physics solvers (i.e., the relaxation of the Poisson equation, the conservative update of hydrodynamics and RT and the chemistry solver), which are fed with arrays of initial states and evolve into arrays of updated values.

Relying on vectors presents several pros. First, it guarantees an optimal data layout in general by ensuring that the data are accessed in a coalesced and aligned manner: computation directly on tree-stored values would induce random and unpredictable memory accesses, whereas an array-based organization ensures the proximity of successive or concurrent calculations thus providing optimal performance. For GPUs in particular, enforcing this kind of memory access is a requirement for obtaining the maximal throughput of the devices. Furthermore, arrays are generic and simple structures of data, which can be processed in a general manner: each element of an array is computed like the others and the implementation of this single-element flow of instructions can usually easily be ported from one architecture to another or even from one language to another. The difference usually arises in the details of the scanning operation for all the elements: they can be parsed in sequence for a scalar calculator or by launching multiple threads of a single-element computation on a GPU or shared memory cores or by taking advantage of the vector abilities of languages such as FORTRAN 90 and PYTHON. Overall, vectorization provides an easy opportunity to choose a language or an architecture for the code computational modules, without any consideration of the design and layouts of the data structure. For instance, EMMA has been coded into both scalar CPU and CUDA GPU versions of the Poisson, hydrodynamics and RT solvers, both versions working in the same AMR framework. In fact, upcoming developments may lead to changes in the way AMR is handled but they would not impact the way the physical engines are designed.

However, it becomes readily apparent that the AMR oct tree being a non-vector-based way to store data, the latter must be converted back and forth from a tree-based organization to an array-based one (see Fig. 5). These *gather* and *scatter* operations are critical to the performance of the code as they constitute bottlenecks to the overall code performance. Nevertheless, if the number of calculations is large enough, the cost of these operations can be hidden by computing or by overlapping transfers and calculations. In practice in EMMA, when data are gathered from the tree to update a value in a given cell, all the necessary values from the neighbours are gathered too. For instance, equation (3) requires seven values to update the potential of a given cell (six from the cardinal neighbours and one for the density). Their related gather operations, therefore, organize the data in seven arrays of n_a values, which are required to update the potential in n_a cells. Of course, since two adjacent cells share some neighbours, the data in these arrays may be redundant. Similarly, intercell fluxes (used during hydrodynamics and RT) are computed twice for two adjacent cells. In principle, such overheads could be avoided but at the cost of coding simplicity and at the current stage the data or flux evaluation has been kept redundant. Gather

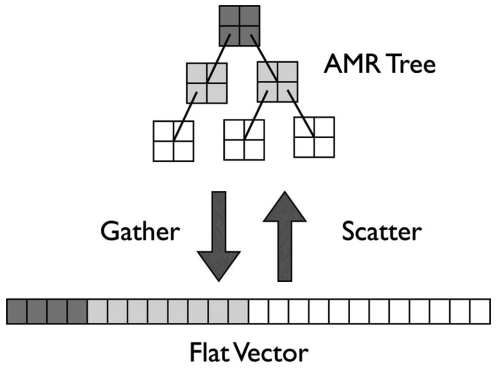

Figure 5. Schematics of the conversion between oct tree data management by the CPU for the AMR structure and array-based calculations, for instance on a GPU. A back and forth flow of data is performed through gather/scatter operations. Grey levels label different refinement levels, in both the tree and the array.

operations are also in charge of dealing with resolution jumps: if a given cell requests data from a neighbour at an unavailable resolution, it is interpolated linearly from coarse data at the position of the fine virtual cell. Global boundary conditions are also dealt with by these gathering operations. As said previously, boundary conditions are periodic by nature: if transmissive boundaries are required, the gather operation replaces the data from the periodic neighbour cell with the data of the current cell. If reflective boundaries are set up, the same operation is performed with an additional flux inversion.

5 COARSE RADIATIVE TRANSPORT APPROXIMATION

Section 2 describes the standard methodology for coupling the different physics within an AMR code with adaptive time stepping. Physical quantities and data structures are updated at the pace of the fastest evolving dynamics among the collisionless, the hydrodynamic and the radiative ones. This implementation has been proven to be both accurate and practically sustainable (in terms of required computing resources) for hydrodynamic codes in the past. However, the inclusion of explicit radiative transport and out-of-equilibrium chemistry severely impact the code's efficiency as it must track processes on timescales one or two orders of magnitude smaller than the pure hydrodynamical case. Hence, an experiment covering a given physical duration must be sampled with a number of operations one or two orders of magnitude greater, including not only physical engines but also any kind of logistics functions or overhead. As such, any deviation from a perfectly optimized set of operations will see its magnitude multiplied by a factor of 100 and reduces significantly code efficiency.

In this section, we suggest an additional level of approximation for the coupling between radiative processes and dynamics (hydrodynamics plus collisionless). It can significantly reduce the resources necessary for a simulation with radiation, at the cost of a degraded (mostly spatial) resolution. It is summarized in Fig. 6 and relies on two sets of additional approximations compared to the standard implementation of Section 2:

(i) Radiation transport and the associated thermo-chemistry are explicitly decoupled from the dynamics (collisionless plus hydrodynamics). Hence, to advance the simulation, dynamical quantities are updated first on all the AMR levels on a timescale constrained only by the dynamics (usually set by the hydrodynamics Courant-Friedrichs-Lewy (CFL) condition). Then, matter is considered as

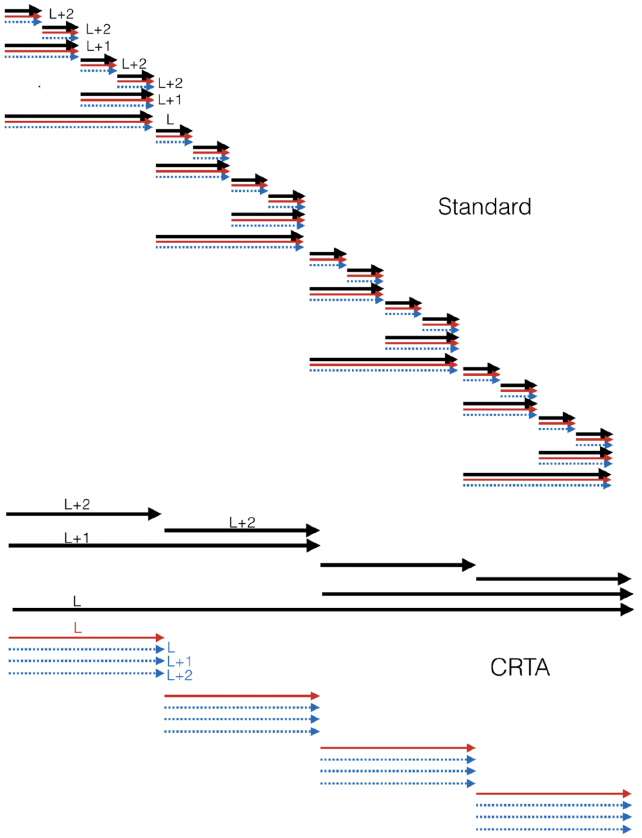

Figure 6. Comparison of the time stepping and multi-physics coupling in the standard AMR approach (top, described in Section 2) and in the coarse radiative transport approximation (CRTA) description (bottom, described in Section 5). Thick black, thin red and dashed blue arrows stand, respectively, for dynamics (collisionless plus hydrodynamics) operations, radiative transport and thermo-chemistry. The thermo-chemistry is shown as a dashed line to indicate that it is already sub-cycled with respect to the radiative transport. In both cases, the simulation is updated by a dynamical time step (from left to right), taken to be equal to 4 times the coarse radiative time step. Note how the red arrows have the same length in both schemes, corresponding to a coarse radiative transport time step. It is also assumed that two additional levels of refinement are enabled ($L+1$ and $L+2$). In the standard case, such an update takes four full updates of the AMR tree, because each update covers a radiative time step, resulting in a total of 83 engine calls. In the CRTA case, the dynamics is updated first over a dynamical time step (seven calls) and radiation plus thermo-chemistry is treated in a second stage (16 calls) for a total of 23 calls. Note how the radiation transport is only performed at the coarse level.

frozen and radiation is propagated within this static distribution for the same duration. Since typical speeds encountered in dynamical processes are of the order of the local sound speed or free-fall velocities, which are much smaller than c, such decoupling should remain under control and provide results similar to the standard procedure. Of course, radiation is subject to a stringent CFL condition, which implies that radiative quantities are updated through an intensive sub-cycling (typically 100–1000 cycles) of the dynamical time step with small radiative timescales.

(ii) Additionally, radiative transport is only performed at the *coarse* level. However, thermo-chemistry is still computed on refined levels, but with a coarse-grained description of the radiation field that is simply injected from the coarse to the fine levels. Not only does it reduce the number of transport operations but it also reduces significantly the number of thermo-chemistry steps: this

engine is already sub-cycled even in the standard approach (see Section 2.2.4) and can therefore operate on a large radiative time step. Furthermore, if an equilibrium situation is encountered in a given cell (a frequent situation in fully ionized regions for instance), this thermo-chemistry sub-cycling can be reduced to a few cycles.

Fig. 6 provides a simplified comparison of the standard and the coarse radiative transport approximation (CRTA) schemes. We arbitrarily chose a situation where the coarse dynamical time step is 4 times larger than the radiative one at the coarse level. In the standard description, the time step is set by the radiative CFL condition at all levels. Hence, for a situation with a coarse plus two refined levels, as the one described in Fig. 6, the number of dynamics, radiative transport and thermo-chemistry engine calls are identical and equal to 28 (four on the coarse level and 24 on the refined ones) for a total of 84 calls. In the CRTA case, it reduces to seven dynamics calls (including six calls on refined levels) plus four radiative transport calls at the coarse level plus 12 thermo-chemistry calls for a total 23 engine calls, i.e., a factor of 3 fewer operations. Bear in mind that a realistic case rather involves a ratio of 100 to 1000 between dynamics and radiative time steps and 5–10 refinement levels: in such cases, the CRTA essentially reduces the cost of the hydrodynamics to zero and by neglecting transport on refined levels, it reduces the cost of a radiative time step by a few tens. Additionally, AMR logistics, communication set-up, analytics, etc. are performed only at the dynamical time step and their costs are also essentially set to zero in the CRTA approach. Incidentally, this technique also increases the relative weight of physical engines over numerical overheads and among the engines it increases drastically the weight of radiative transport plus chemistry over the others: it turns out that the latter engine is one of the most efficient in the use of vectorization (see Section 4) and therefore in the use of hardware accelerators such as GPUs. CRTA is expected to benefit more from such devices in accelerating the code.

Of course, this increased efficiency comes at a cost. The most evident one is the decreased spatial resolution for radiative transport, even though the thermo-chemistry is performed at the highest resolution available. It could be thought of as an intermediate approach between a homogeneous radiation field (as usually assumed in non-RT cosmological simulations) and a full AMR description. In the CRTA approach, spatial UV field fluctuations exist but are coarsened. However, the impact could be limited. First, as shown in Aubert & Teyssier (2010), radiation fields (radiative density and flux) do not exhibit significant clumping factors compared to those of the matter distribution and are relatively smooth even in highly resolved simulations. In fact, the radiative densities span orders of magnitude between sources and dark voids and are therefore not very sensitive to local fluctuations. Furthermore, we use here a highly diffusive scheme (based on a Lax-Friedrichs intercell flux evaluation), which accentuates further the smooth aspect of radiative fields. One could therefore argue that having a fine spatial description of radiative density fields is not absolutely necessary. The other level of approximation is the imperfect coupling of radiation and matter, the latter being considered as still when light is being cast. Somehow, it relates to previous post-processing techniques but performed on the fly at every dynamical time step. Post-processing is known to provide satisfying results for large-scale experiments on $\sim$50+ Mpc scales, hence we can be confident that this imperfect coupling can be controlled in such cases. On the other hand, it is clear that some coupling between matter and radiation in highly refined cells will be lost and it is difficult to evaluate this loss properly. As shown in Section 6, the CRTA returns very satisfying results for the tests shown here but in general using CRTA would require an additional level of convergence study to ensure that this imperfect coupling is under control.

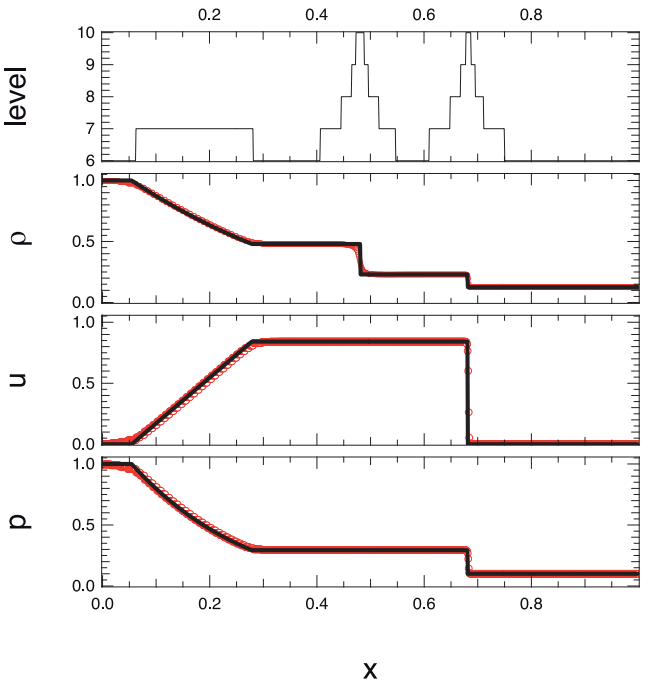

Figure 7. Shock tube experiment. From top to bottom: refinement level, density, velocity and pressure as a function of position. Points stand for the simulation results and solid lines for analytic profiles.

6 CODE VALIDATION

EMMA has been submitted to a series of tests to validate its implementation. For various combinations of simulated physics, documented experiments are described here, as well as the code results.

6.1 1D hydrodynamics: shock tube

The shock tube is a 1D test where a Riemann problem is evolved by a simulation. It focuses on the implementation of hydrodynamics and the ability of the MUSCL scheme and HLLC Riemann solvers to capture shock features. The initial conditions consist of a jump at $X_0 = 0.3125$ between two different states ($\rho_1 = 1, u_1 = 0, p_1 = 1$ and $\rho_2 = 0.125$, $u_2 = 0$, $p_2 = 0.1$, taken from Toro 1997). The solution to this Riemann problem is known and can therefore be compared to the results delivered by EMMA.

The calculation has been performed using a $\ell = 6$ coarse resolution with four additional levels of resolution, triggered by density gradients satisfying $\Delta\rho/\rho \geq 0.015$. Even though the problem is 1D, the calculation has been performed in 3D with the jump occurring along the x direction. Transmissive boundary conditions were retained along the x direction and periodic ones along the two others.

Fig. 7 shows the density ρ, the velocity along the x direction u, the pressure p and the refinement level at $t = 0.2$. Also shown is the solution of the Riemann problem, as a solid line. Clearly, the match is satisfying with shocks being resolved on a few cells, thanks to both the shock-capturing scheme and the improved resolution allowed by the on-the fly refinement. It can also be noted that the contact wave

is also present near $x = 0.5$, even though some smearing can still be present at this resolution. Overall, this standard test demonstrates the ability of EMMA to solve classic hydrodynamical problems at high resolution.

6.2 1D gravity plus hydrodynamics: Zeldovich pancake

The Zeldovich pancake test (Zel'dovich 1970) tracks the evolution of a single planar mode in an $\Omega_m = 1$ expanding Universe, where the linear stages of the evolution can be analytically predicted. The initial matter density is given by:

$$\rho(x) = 1 + \frac{1+z_\mathrm{c}}{1+z_\mathrm{i}} \cos(2\pi x) \tag{58}$$

whereas the initial velocity is given by:

$$u(x) = \frac{1}{\pi} \frac{1+z_\mathrm{c}}{(1+z_\mathrm{i})^{3/2}} \sin(2\pi x). \tag{59}$$

Here the mode oscillates along the x direction. z_i and z_c stand for the initial and collapse redshifts.

As in Section 6.1, this test case has been simulated with EMMA in 3D as a planar experiment. Both the baryons and DM were included with $\Omega_\mathrm{b} = 0.1$. The base resolution is $\ell = 6$ (i.e., 64^3 cells) and the DM field is sampled with 64^3 particles. The initial temperature is chosen to be arbitrarily small at $T = 10$ K and velocities orthogonal to the x direction are taken to be zero. The experiment has been conducted down to $z = 0$ with $z_\mathrm{i} = 100$ and $z_\mathrm{c} = 10$. Two additional refinement levels were triggered on gas density gradients $\Delta\rho/\rho > 0.1$. This tests the cosmological setting and the coupling between DM and baryons. Linear stages were compared to the analytic solution and were found to match at better than the per cent level (not shown here) until the redshift of collapse.

Fig. 8 shows the $z = 0$ baryon density, velocity and pressure as well as the DM phase diagram. Clearly, being way later than the collapse redshift ($z_\mathrm{c} = 10$), it can be noted that several plane crossings occurred with a significant number of foldings for the DM phase space diagram. Baryons fell in the DM potential, creating shocks and an inner increase of temperature (via the pressure) within the collapsed region. In particular, note how the infall velocity of the gas is strongly reduced as it enters the collapsed gas. Refinement levels were triggered as expected, providing a better resolution of the density peak and a smoother description of the phase space curve of DM in the innermost regions. Finally, direct comparisons with, e.g., Teyssier (2002), show that the results of EMMA are consistent with other codes, even at these latest stages of the pancake collapse.

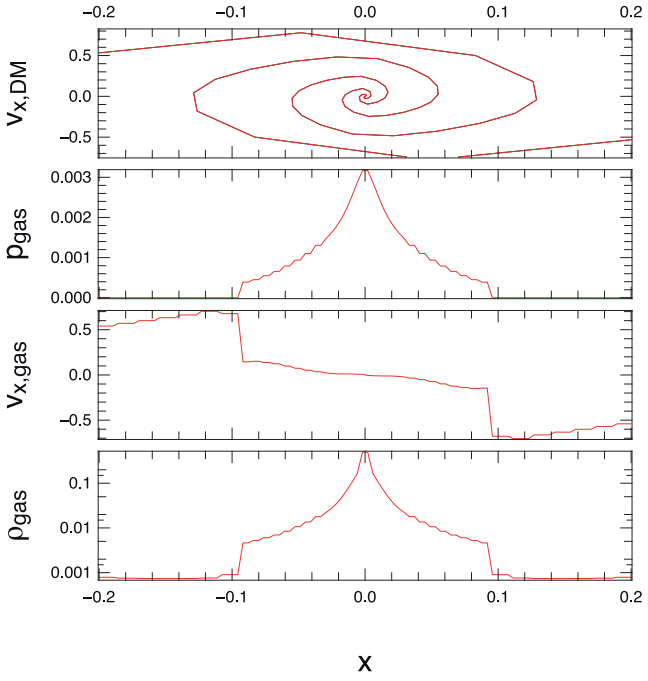

Figure 8. Zeldovich pancake experiments with baryons ($\Omega = 1$ and $\Omega_\mathrm{b} = 0.01$). From top to bottom: Dark matter velocity $v_{x,\,\mathrm{DM}}$, gas pressure p_gas, velocity $v_{x,\,\mathrm{gas}}$ and density ρ_gas. All the quantities are shown as a function of the position, at $z = 0$. The sine wave is initiated at $z = 100$ and collapsed at $z = 10$. Note the increasing resolution towards the centre of the caustic, from $\ell = 6$ to $\ell = 8$.

6.3 3D gravity plus hydrodynamics: Bertschinger's self-similar infall

This experiment aims at reproducing the calculation made by Bertschinger (1985) in which a top-hat overdensity within an expanding Einstein–De Sitter universe ($\Omega_m = 1$) collapses towards a scale-invariant density distribution in a self-similar fashion. Provided that radii are rescaled to the turnaround radius λ, this self-similarity can be fully predicted analytically. In the original paper, several configurations are explored and here we focus on the evolution of an overdensity that includes baryons in a DM-dominated potential. Compared to the test described in Section 6.2, the situation here is 3D with a spherical symmetry.

In practice, we generated a regular lattice of 128^3 DM particles starting at $z = 1000$ in a 1 Mpc box, with cosmological parameters $\Omega_m = 1$, $\Omega_\mathrm{b} = 0.01$, $H_0 = 70$ km s^{-1} Mpc^{-1}. Two types of DM particles co-exist: particles within a radius $R_i = 0.05$ Mpc from the centre were assigned a greater mass (equal to 7.89×10^4 M$_\odot$) than particles at larger distance (with a 6.37×10^4 M$_\odot$ mass), to produce a central overdensity $\delta_i = 0.2$. Also, an arbitrarily cold and motionless gas has been sampled on a coarse grid of 128^3 (i.e., $\ell = 7$), with the same central overdensity as DM: cells within a 0.05 Mpc radius were assigned a 797 M$_\odot$ mass and ones at greater radii were given a 604 M$_\odot$ mass. Mesh refinement triggers when the mass within a cell is greater than 8 times the mass of a low-mass DM particle: this criterion is similar to the one used in cosmological simulations to provide a quasi-Lagrangian strategy.

Fig. 9 shows the radial profiles of the DM and baryon densities as well as the baryon radial velocity, for the simulation and compared to the fits of the analytic solution provided by Bertschinger (1985). Also shown is the spherical average of the refinement level. As in Bertschinger (1985), the DM density D_dm, the baryonic density D_b and the baryonic radial velocity V are expressed in units of turnaround values:

$$D_\mathrm{dm} = \frac{\rho_\mathrm{dm}}{\rho_\mathrm{ta}} \tag{60}$$

$$D_\mathrm{b} = \frac{\rho_\mathrm{b}}{\rho_\mathrm{b,ta}} \tag{61}$$

$$V = \frac{v_\mathrm{b}}{V_\mathrm{ta}}, \tag{62}$$

where $\rho_\mathrm{ta} = (6\pi G t^2)^{-1}$ and $\rho_\mathrm{b,ta} = \Omega_\mathrm{b}\rho_\mathrm{ta}$. Bertschinger (1985) gives the evolution of the turnaround radius:

$$r_\mathrm{ta}(t) = \left(\frac{4\pi t}{3t_\mathrm{i}}\right)^{8/9} \delta_\mathrm{i}^{1/3} R_\mathrm{i}, \tag{63}$$

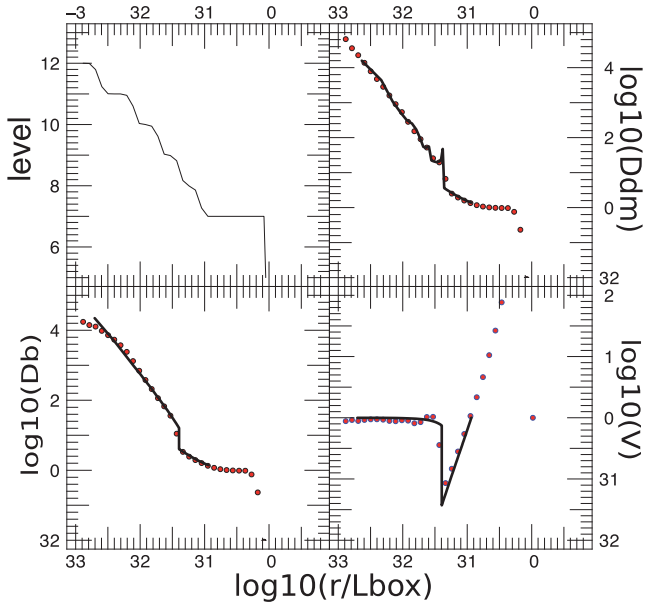

Figure 9. Self-similar 3D collapse of a top-hat density perturbation. Clockwise from top left: The radial profile of refinement level ℓ, DM density $D_{\rm dm}$, baryon radial velocity V and density $D_{\rm b}$, at $a = 0.56$. As in Bertschinger (1985), these quantities are expressed in units of turnaround quantities. Dots are for simulation results and lines stand for the analytic prediction of Bertschinger (1985). Radii are in units of the box length.

where $t_{\rm i}$, $\delta_{\rm i}$ and $R_{\rm i}$ are, respectively, the initial time, the initial overdensity and the initial overdensity radius, from which an expression for the associated velocity can be obtained:

$$V_{\rm ta} = \frac{r_{\rm ta}}{t}. \tag{64}$$

All the results are shown for $a = 0.56$. Clearly we manage to reproduce the analytic solution, in particular the predicted inner logarithmic slope of $-9/4$ for the DM density profile and the shock positions. A small amount of diffusion can be seen in the innermost regions in the baryon density. The velocity jump from the Hubble flow to the shocked region is not as sharp as the predicted one, but the overall features are well reproduced by our calculations and of the same quality described for ART and RAMSES.

6.4 3D radiative hydrodynamics: growth of a H II region

This test consists of a corner source, powered by a 100 000 K blackbody that sends photons into a surrounding homogeneous hydrogen-only medium. It belongs to the suite proposed by Iliev et al. (2006, 2009) and comes in two different versions. The first version deals with a static and uncoupled gas (Test 2, Iliev et al. 2006): we obtained a very good agreement with EMMA (not shown here), which does not come as a surprise since this test was also successfully passed by ATON and RAMSES-RT, which share a great number of details with EMMA. Here we focus on the second version, a coupled test known as Test 5. The UV photons ionize and heat up the gas, leading to the creation of ionization fronts that propagate inside out and put the gas into motion. This test couples RT and hydrodynamics and has been performed by a whole series of codes in Iliev et al. (2009).

A 100 000 K blackbody is located at the corner ($x = y = z = 0$) of a 15 kpc box, emitting 5×10^{48} UV photons per second. We sample the frequencies with the three groups of photons given in

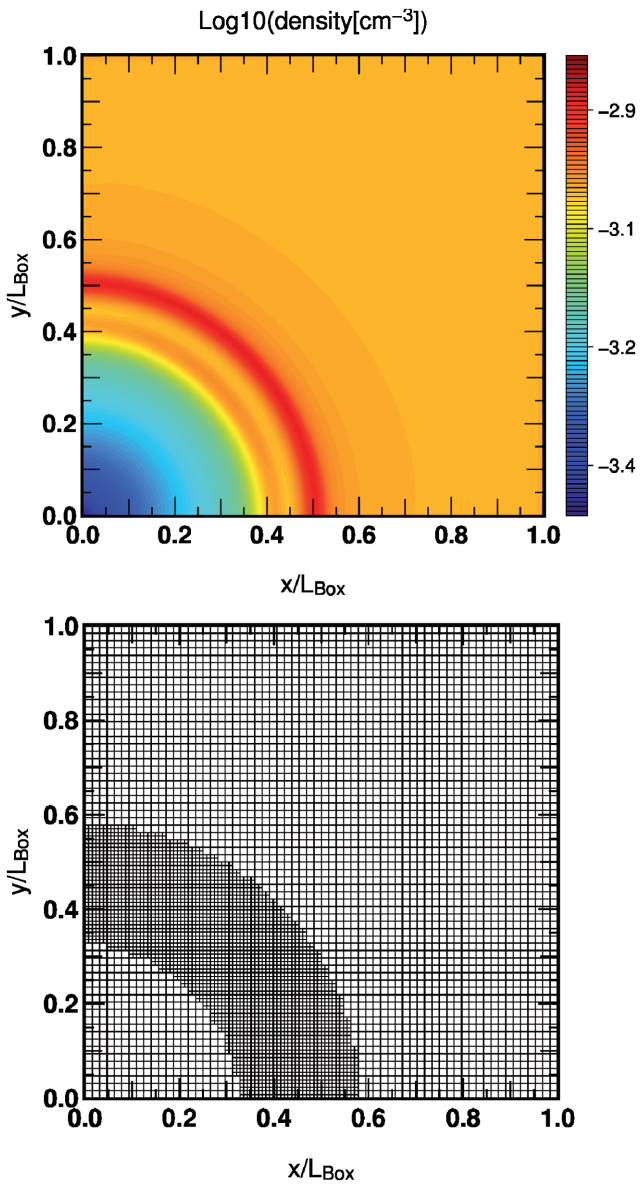

Figure 10. Expansion of a H II region. Top: $\log_{10}$ of density map (in cm^{-3}). Bottom: AMR grid used for the computation. The coordinates are expressed in units of the box length, which has a physical extent of 15 kpc. The source is located at the bottom left corner and was ignited 200 Myr ago.

Section 2.2.3. The surrounding gas has a homogeneous number density of 0.001 hydrogen atoms per cm^3. The calculation is run on a 64^3 coarse grid ($\ell = 6$), allowing for an additional level of refinement (128^3, i.e., $\ell = 7$) to comply with the resolution requirements of Iliev et al. (2009). The refinement is simply triggered for cells with ionized fraction $0.01 < x < 0.8$: with the two-cell layer of neighbours being also refined, it is simple to track the ionization front. The boundary conditions are reflective for boundaries adjacent to the source and transmissive otherwise.

Fig. 10 shows the baryon density in the $z = 0$ plane, as well as the AMR structure that tracks the ionization front for $t = 200$ Myr. The source being in the bottom left corner, the distant cells remain at the base resolution as the front has not yet progressed to these regions. The cells close to the source are also at the base resolution: this region has returned to low resolution as it does not contain an

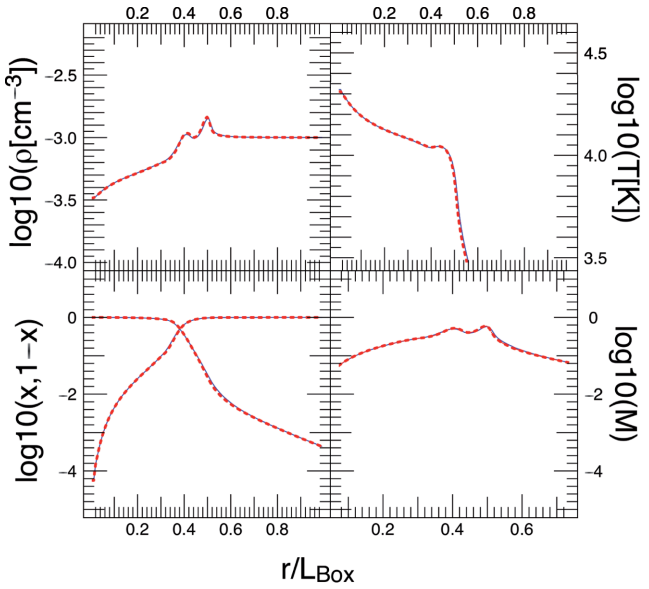

Figure 11. Expansion of a H II region. Clockwise from top left: Gas density, temperature, Mach number and ionized/neutral fraction. The coarse resolution is $\ell = 6$ and refinement is triggered on the ionization front to $\ell = 7$ in accordance with the Iliev et al. Test 5. Red dashed lines stand for the radial average taken 200 Myr after the source has been ignited. Blue lines stand for the same calculation, but performed with the CRTA.

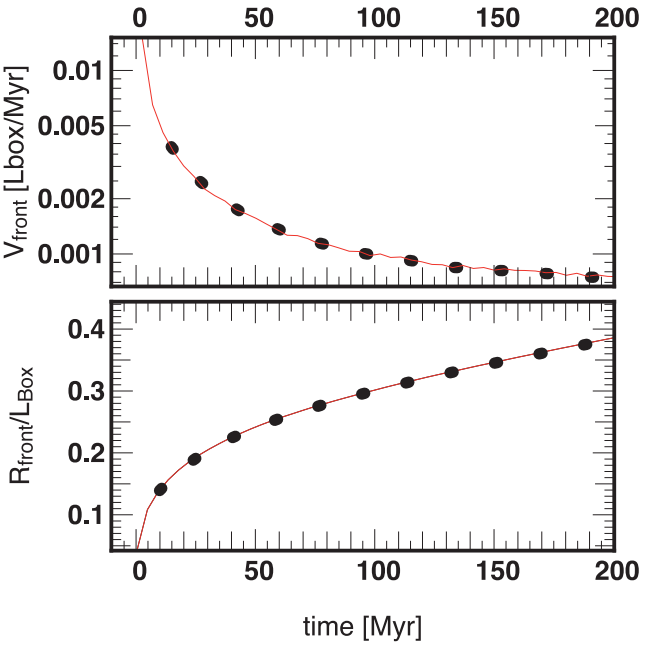

Figure 12. Expansion of a H II region. Front propagation predicted by the standard AMR RT implementation (red line) and CRTA (dots). Top: Front velocity. Bottom: Front position.

ionization fraction that satisfies the refinement criterion (chosen to track front-like features).

Finally, between these two ensembles of coarse cells, one can find the refined region at 128^3 resolution, tracking the front. On the top panel of Fig. 10, the number density field of hydrogen is also shown, presenting a typical double peak structure encompassing a void created by the energy injection by the corner source.

Fig. 11 shows the radial profile of the density at the same instant (i.e., 200 Myr after the source ignition) as well as the temperature, ionization/neutral fraction and the Mach number profiles. EMMA recovers the typical features already obtained by most of the codes of Iliev et al. 2009. The double peak is due to the presence of high-energy photons in the spectrum of the 10^5 blackbody: these photons with large mean free path can deposit energy behind the ionization front, while lower energy photons deposit their energy at the base of the front. The input of energy at larger radii is also at the origin of the moderate temperature increase seen before its drop in the neutral region. The effect of hard photons can also be seen in the extent of the ionized fraction drop at the front, which would be much sharper for a monochromatic incoming flux. Overall, that EMMA reproduces these specific features seen by all other codes indicates that the coupling between radiation and matter and the handling of multi-frequency transfer is consistent with other implementations.

Figs 11 and 12 provide the different CRTA field profiles as well as a comparison of the temporal evolution of the front position and velocity (the front being defined as having a 50 per cent ionized fraction) in both cases. Clearly CRTA provides the same results as the standard calculation: in Fig. 11 the radial profiles taken at $t = 200$ Myr are almost indistinguishable and in Fig. 12, the front position and velocities of the CRTA calculation follow the ones obtained from the standard procedure. The evolution is smooth enough both spatially (with features sampled on ∼15 fine cells or seven coarse cells) and temporally (with terminal velocities as small as 0.1 per cent box lengths per million years) to ensure a good convergence of the CRTA towards the standard case.

6.5 3D radiative hydrodynamics: photo-evaporation of a dense clump

This situation has also been suggested by Iliev et al. (2009) and consists of a dense cold clump irradiated by a planar UV front. As the ionization front encounters the cloud, the high density will slow down its progression, acting as a trap for the incoming photons. As a side effect, a shadow will also be cast in the trail region of the clump. Finally, the energy deposited in the clump will put the gas in motion, leading to photo-evaporation by the incoming photon flux.

The set-up is given by Iliev et al. (2009): a spherical clump of radius 0.8 kpc is centred on (5.4, 3.3, 3.3) kpc inside a 6.6 kpc box. Outside the clump, the gas has a 8000 K temperature with a density of 200 atoms m^{-3}. The clump itself has a density of 40 000 atoms m^{-3} and a temperature of 100 K. A UV flux with a 100 000 K blackbody spectrum is incoming from the $x = 0$ boundary at a rate of 10^{10} photons m^{-2} s^{-1}. In practice, the simulation is performed on a 64^3 grid ($\ell = 6$) with one additional refinement level to comply with the Iliev et al. (2009) recommended resolution. Mesh refinement is triggered for cells with a density greater than the background density. The $x = 0$ boundary is a source of flux of the required rate, whereas the $x = 6.6$ kpc boundary is transmissive. Boundaries in the two other directions are periodic.

Figs 13 and 14 show maps of the neutral fraction and gas density at $t = 10$ and 35 Myr. In each figure, the top and middle rows were obtained from 64^3 simulations with an additional level of refinement, the top being obtained from the standard RT implementation on AMR and the middle panel being obtained from CRTA simulations.

Globally, a simple comparison with the results of Iliev et al. (2009) demonstrates their consistency. In particular, the evaporation is obvious with the expansion of the cloud limits due to the

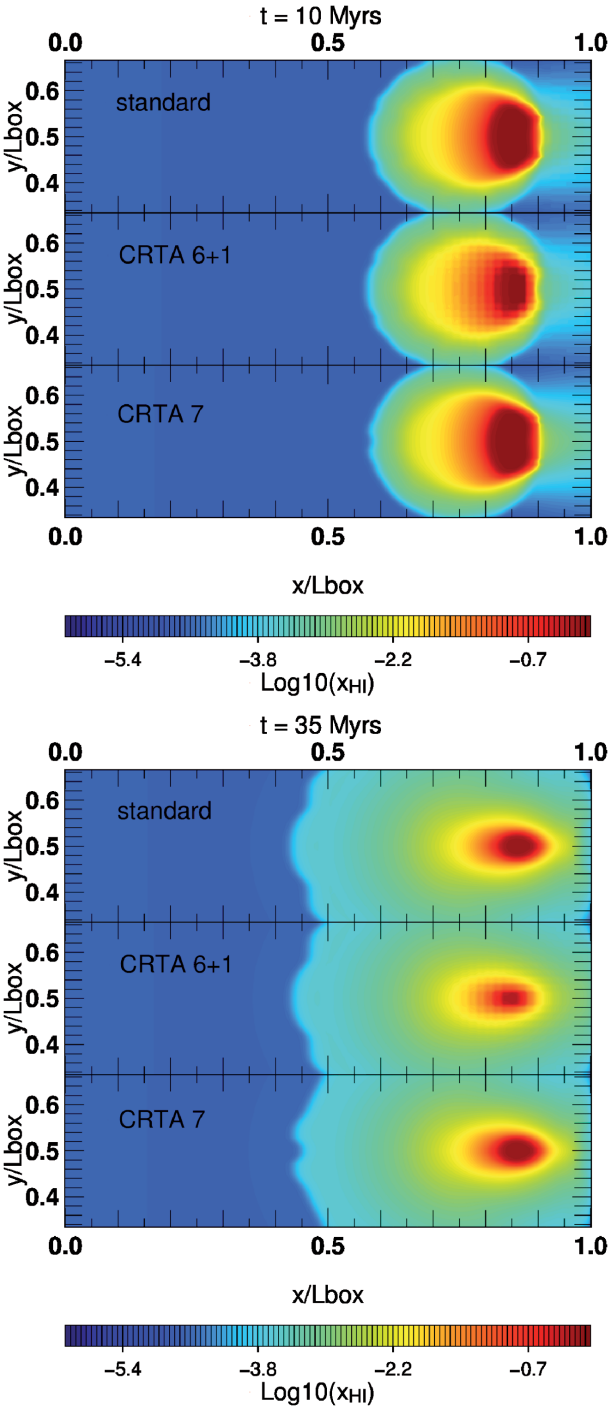

Figure 13. Photo-evaporation of a dense clump. Top: $\log_{10}$ of the neutral fraction along the plane of symmetry of the clump at $t = 10$ Myr, as predicted by the standard AMR RT implementation (top) and CRTA (middle), both assuming an $\ell = 6$ base level plus one level of refinement. The bottom row is the prediction of CRTA on a static $\ell = 7$ grid. Bottom: The same quantities at $t = 35$ Myr. Blue is for ionized and red for neutral matter.

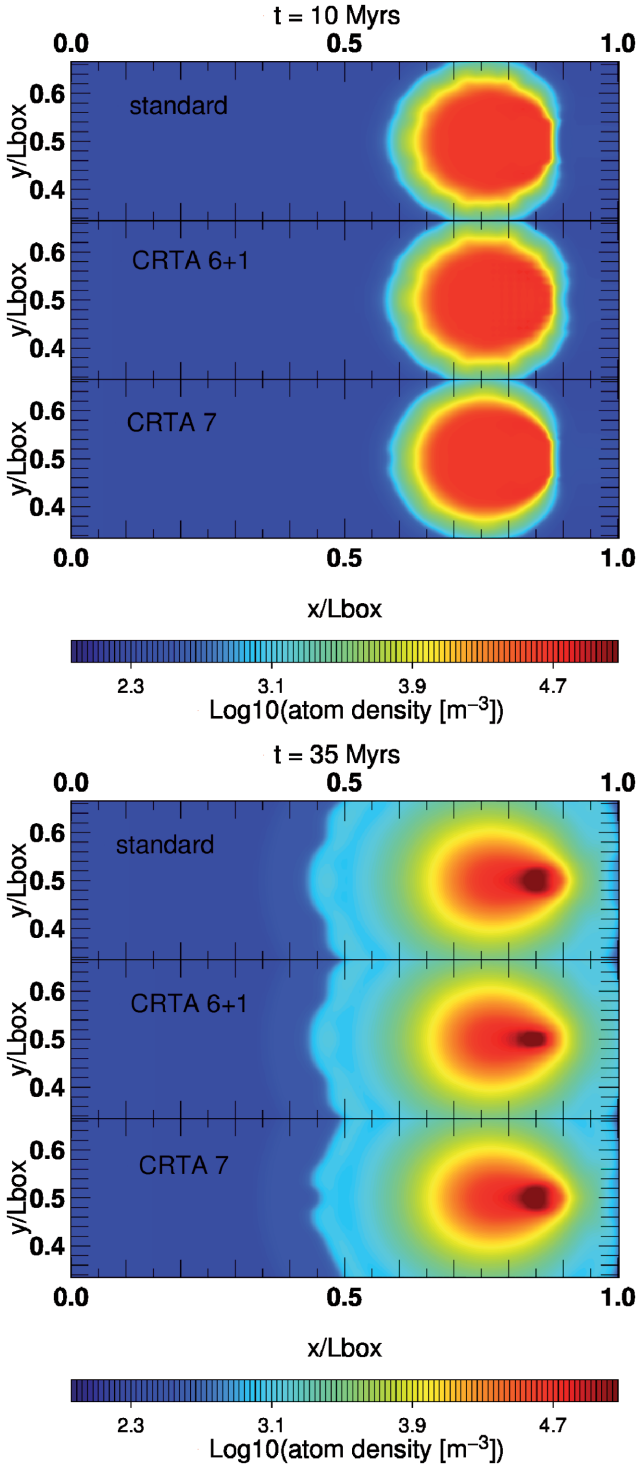

Figure 14. Same as in Fig. 13 but for the gas density. Underdense regions are blue and dense regions are red.

energy injected by the UV front. However, clear differences can also be noted: first the shadow behind the clump, albeit existent, is much weaker than in other RT codes. This is not a surprise since EMMA implements the global Lax-Friedrichs flux to compute inter-cell exchanges and is known to be very diffusive. This, therefore, prevents the creation of clear-cut shadows as a diffuse component of the flux eats the neutral gas in a direction orthogonal to the incoming direction of the UV photons. The same effect was already noted for ATON (Aubert & Teyssier 2008). Secondly, at late time, the contours of the extended cloud are not as spherical as expected and present significant fluctuations around a mean radius. These fluctuations are artefacts of the initial cloud sampling on the coarse $\ell = 6$ grid. The same artefact can be seen, e.g., in the FLASH-HC results in Iliev et al. (2009), which are also linked to the initial conditions. Comparing

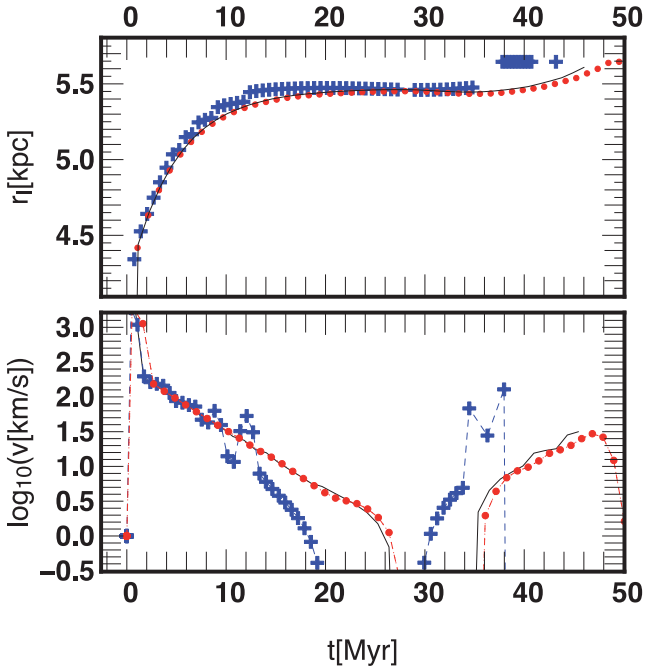

Figure 15. Photo-evaporation of a dense clump. Position (top) and velocity (bottom) of the ionization front along the x-axis as a function of time. The solid line stands for the standard AMR RT implementation results and blue crosses stand for CRTA results. Both were obtained using an $\ell = 6$ base level plus one level of refinement. Red dots stand for CRTA results with a $\ell = 7$ static grid.

standard RT and CRTA, it can be seen that the latter provides a faster ionization of the clump. Looking at the Fig. 13, the shadowed neutral clumps are systematically smaller in the CRTA regime. This is not a surprise since the RT is performed at the base level only, which increases artificially the extent of the UV flux penetration into the clump and also increases the scheme diffusion. Fig. 15 provides a more quantitative insight on this aspect, describing the front progression inside the cold clump and its velocity. The front position is defined as corresponding with the position of the cell having $x = 0.5$. At early stages ($t < 7.5$ Myr), CRTA and the standard description produce an identical front propagation. However, note that CRTA presents a step-like progression due to the coarser resolution of the RT, which translates into a coarser resolution of the ionized front. Later, the front is pushed back by the expanding cloud in both descriptions (as can be seen from the recessing velocities), but it happens earlier for CRTA. Finally, the front cannot be tracked for $t > 38$ Myr, as no cells with a neutral fraction greater than 0.5 can be found any more. Let us mention that comparisons of the standard calculations with the results presented in Iliev et al. (2009) confirm the capacity of EMMA to track correctly the front propagation within the clump. In particular, EMMA recovers, as the other codes, the phase where the front is pushed back by the expanding cloud, when $t \sim 35$ Myr. Globally a faster photo-evaporation of the cloud can be detected in CRTA compared to the standard AMR description, essentially due to the coarse description of the radiative fields.

Finally, we present in the lower panels of Figs 13 and 14 the results of a CRTA calculation on a static $\ell = 7$ grid. It allows us to probe the separate effects of incomplete coupling between the dynamics and RT and of the coarse description of the radiation. In this experiment, the CRTA is equivalent to a radiative post-processing of the dynamics but performed on the fly, at the temporal scale of dynamical times and without any impact of a coarsened RT. Compared to the standard treatment, RT is sub-cycled with respect to the dynamics, leading to a large number of RT plus thermo-chemistry calls per single gravity and hydrodynamics calculation. Clearly, CRTA greatly reproduces the standard calculation in this case, both in the neutral fraction and density maps and in the propagation of the fronts. It confirms that the coarsened resolution is indeed the reason for an accelerated photo-evaporation of the clump and that radiation sub-cycling does not induce significant deviations from the standard treatment.

6.6 Cosmological runs

6.6.1 Preliminary reionization simulations

Finally, we present the results of full simulations of cosmological reionization. The focus is on hydrodynamical simulations with radiative transport and on reionization-related quantities but additional tests on the DM halo mass function and energy conservation are presented in Appendix A. We produced a set of four simulations with four different specific emissivities for the sources. Each simulation consists of a 4 Mpc h^{-1} box sampled with 128^3 base resolution cells and 128^3 DM particles. These simulations will be referred to as X0.3, X1, X3 and X30. The X1 simulation is a fiducial run with source emissivities that produce a reasonable ionization history. The three additional cases use stars with boosted or depleted specific emissivities by the corresponding factor, X0.3 and X30 standing, respectively, for the dimmest and brightest source models. Initial conditions were produced using MPGRAFIC (Prunet et al. 2008) with a Planck cosmology (Planck Collaboration XVI et al. 2014) ($\Omega_m = 0.315$, $\Omega_\Lambda = 0.685$, $\Omega_b = 0.049$, $n_s = 0.96$ and $H_0 = 67$ km s^{-1} Mpc^{-1}) starting at $z = 80$. Each DM particle weighs 4×10^6 M$_\odot$. AMR is triggered using a quasi-Lagrangian strategy and a cell is refined if it contains more than eight DM particles. RT is run with three groups of frequencies ([13.6,24.6] eV, [24.6,54.4] eV and [54.4,1000] eV), dictated by the ionization thresholds of hydrogen and helium.

In addition to the hydrodynamics and RT, we had to implement a simple star formation recipe to populate the simulated volume with the ionizing sources that drive the reionization process. This star formation model is briefly described here and will be the subject of a dedicated paper in the near future: its is widely inspired by Katz, Weinberg & Hernquist (1996), Kay et al. (2002), Rasera & Teyssier (2006) and Dubois & Teyssier (2008). A cell is said to be prone to star formation if either its gas comoving density (n_*) or its gas density contrast (δ_*) are greater than user-set thresholds. Once a cell is flagged to form stars, the number of stellar particles to be created is drawn from a Poisson law with the λ parameter given by:

$$\lambda = \epsilon \frac{m_{\rm cell}}{m_\ell} \frac{\Delta t}{t_*}. \tag{65}$$

λ corresponds to the average number of stars created during a time step Δt within a cell that contains a mass of gas given by $m_{\rm cell}$. The star formation process is controlled by a typical star formation timescale t_* and an efficiency parameter ϵ. The mass of a stellar particle is given by m_ℓ, which depends on the level of the cell and is equal to

$$m_\ell = \bar{\rho}_b \delta_* \Delta x^3. \tag{66}$$

The following results were obtained with $\delta_* = 150$ and $t_*/\epsilon = 2$ Gyr. These values are considered as standard even though we do not discuss them here. We will explore thoroughly the dependence of the results on these values in a forthcoming paper. As shown hereafter,

they nevertheless lead to a star formation and a reionization process in reasonable agreement with the constraints. Each stellar particle emits photons for 20 Myr, with a constant emissivity and assuming a 50 000 K blackbody spectrum (see Baek et al. 2009). In practice, the source emissivity has been tuned by trial and error to produce a reasonable reionization history, complete at $z \sim 6$ and for the fiducial X1 model, it results in an emissivity of 1.5×10^{16} ionizing photons per second per stellar kilogram. Taking the calculation of Baek et al. (2009) as a reference, which assumes a Salpeter initial mass function and 1–100 $M_{\odot}$ mass range, it corresponds to a 15 per cent escape fraction. Again, X0.3, X3 and X30 use emissivities multiplied by the corresponding factor, the X30 model being clearly over-powered and merely used to probe the qualitative behaviour of the code in the regime of strong radiation.

At the current stage we restrict ourselves to this simple model that obviously lacks important ingredients. For instance, supernova and active galactic nucleus feedback has not been implemented yet, chemistry is limited to simple hydrogen and no modification of the equation of state is assumed at very high densities. As a consequence, the star formation rate (SFR) is essentially not regulated in this cosmological toy model. Hence, the following results should not be considered as definitive for what the code can do but should be seen rather as tests of experimental configurations close to production runs.

Figs 16 and 17 present the distribution of matter, AMR levels, temperature and ionized hydrogen fraction in a 320 kpc h^{-1} thick slab of the X1 run at $z = 6.8$. Clearly, the matter on these scales is already highly structured at $z = 6.8$, with regions having a density contrast greater than 1000. These regions are effectively tracked by the AMR grid and the overall distribution of high-resolution grids follows the main features of the filamentary structure in this simulated volume. Sources are created in the overdensities and their radiation leads to large H II regions. Note how the fronts are locally prevented to progress into the inter-galactic medium by filaments and dense clumps, leading to complex features in their geometry. It can also be seen that the ionization fronts present a certain extent induced by the larger mean free path of high-energy photons. It also leads to a preheating of the gas, behind the ionization fronts, to temperatures close to a few thousand kelvins. Within the ionized regions, a quasi-homogeneous temperature close to 10 000 K is set by the UV radiation with local fluctuations correlated with the density field. Some shock-heated gas (located at $\sim$[2, 1.7] Mpc h^{-1}) with temperatures greater than 100 000 K can also be seen.

The fiducial model X1 presents a reasonable reionization history and SFR, in broad agreement with observational constraints (see Fig. 18). Compared to Fan et al. (2006), the ionization happens slightly earlier than observed and correspondingly the photoionization rate at $z \sim 6$ is overestimated compared to Calverley et al. (2011). The cosmic star formation history is also in excess compared to the observationally deduced rates given by Bouwens et al. (2015). This fiducial model could have benefited from a slightly improved calibration to reproduce the observed data points; however, we consider that the current level of agreement is good enough at this stage: recall, for instance, that these simulations lack supernova feedback and the small simulated volume could also be inadequate for making quantitative predictions on cosmic averaged quantities. At this stage, we merely aim at looking for qualitative and not quantitative clues of the impact of radiation within a cosmological setting.

These clues can be obtained by comparing this fiducial simulation with the three other models, X0.3, X3 and X30. As expected, these models result in different reionization histories at earlier (respectively, later) times for larger (respectively, lower) emissivities (see Fig. 18). The SFR, however, remains essentially unaffected by the change in emissivity, except at later times ($z > 10$) for the models with the brightest sources (X30), leading to a depleted SFR.

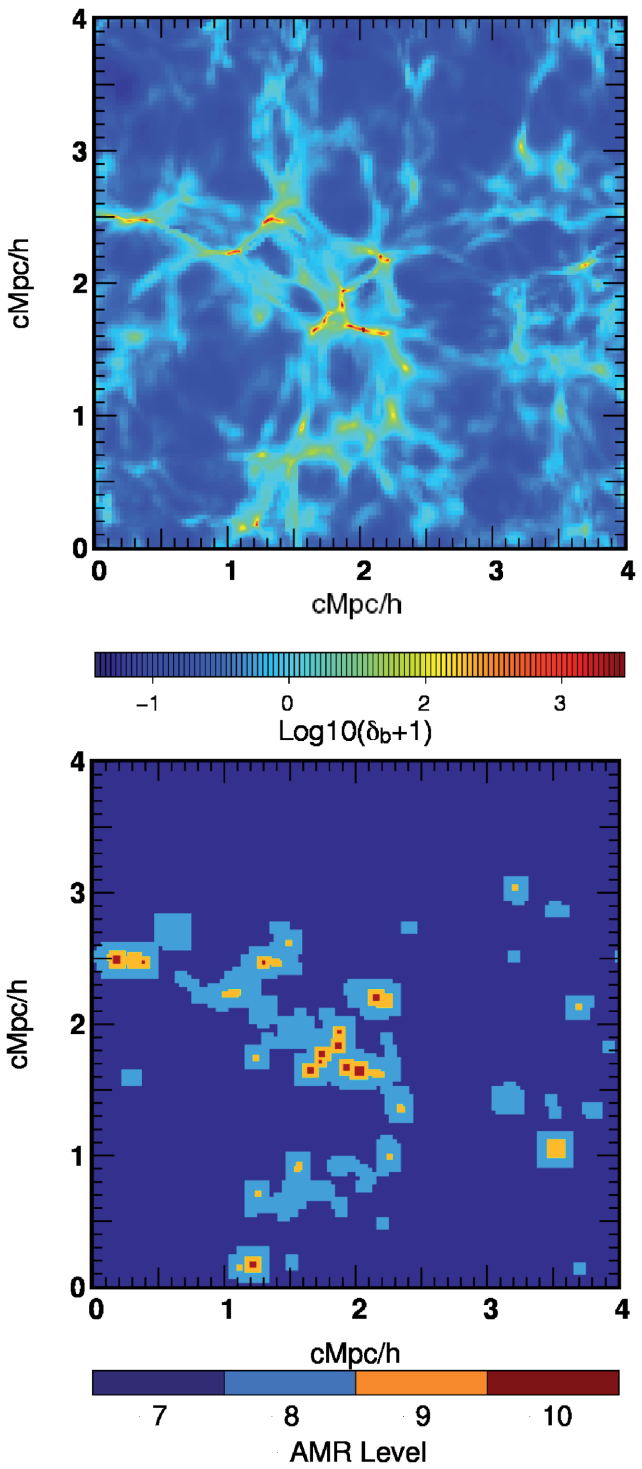

Figure 16. Structuration of matter in a cosmological RT run of a comoving 4 Mpc $h^{-1}/128^3$ box, taken at $z = 6.8$. Top: Baryon overdensity map. Bottom: AMR levels. The region shown has a thickness of 320 comoving kpc h^{-1}.

Fig. 19 presents the baryon fraction and the instantaneous SFR measured in the DM haloes found at $z = 5.5$ in the different models.

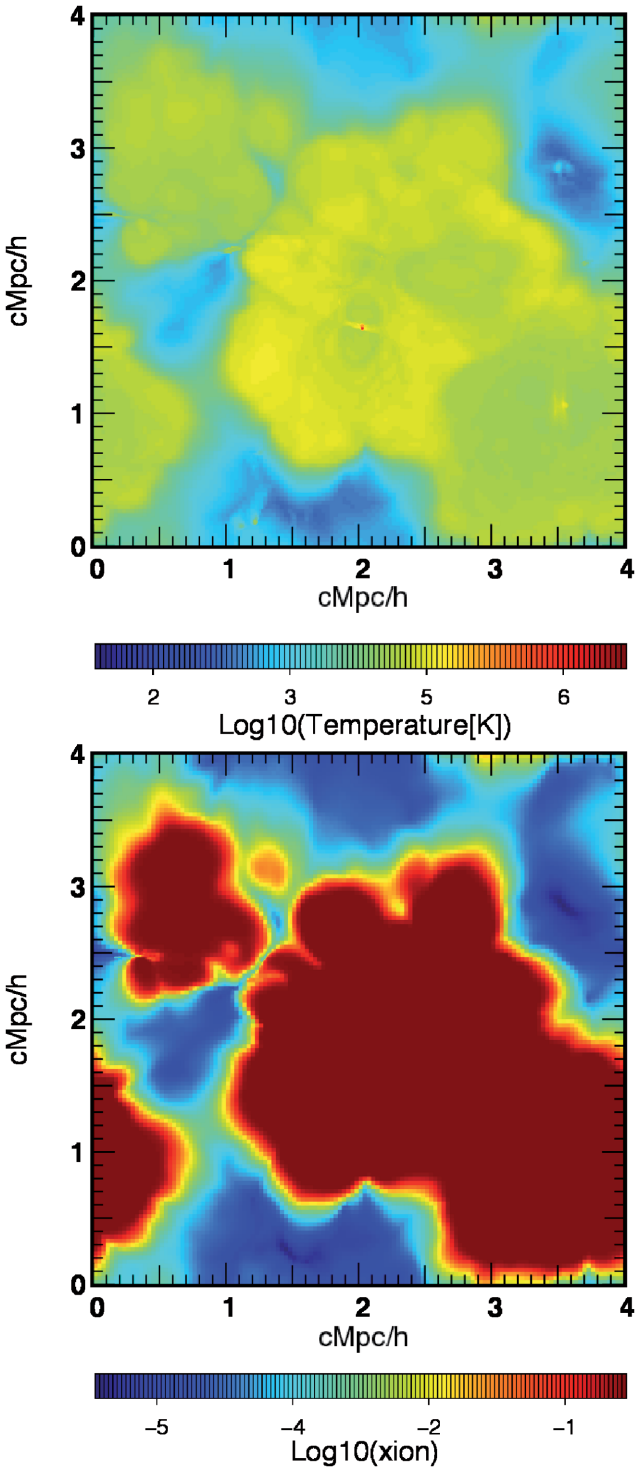

Figure 17. Same experiment and region as in Fig. 16. Top: Temperature map. Bottom: Map of fraction of ionized hydrogen.

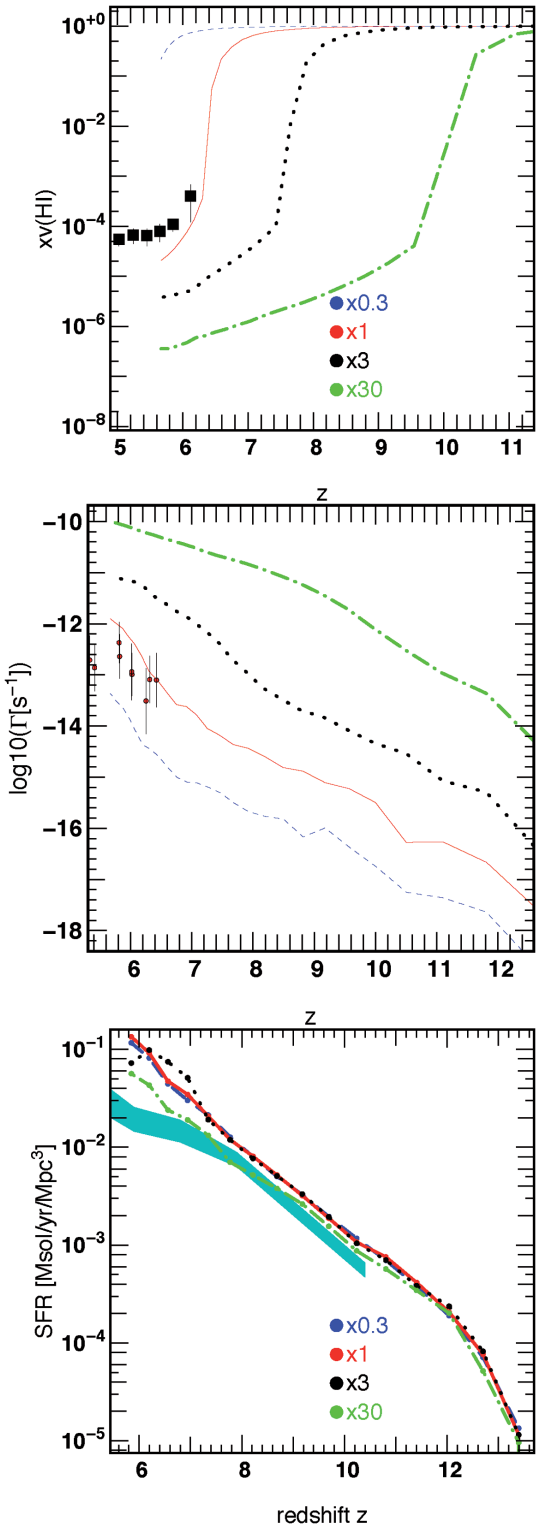

Figure 18. Global evolution of the average volume weighted neutral fraction (top), photoionization rate (middle) and star formation history (bottom) in 4 Mpc/128^3 reionization simulations. The fiducial model is shown in solid red (labelled as having an X1 emissivity) while simulations with different emissivities are shown in dashed blue (with an emissivity equal to 30 per cent of the fiducial one, X0.3), dotted black (X3) and dash-dotted green (X30). Observational constraints from Fan et al. (2006) (top panel, squares), Calverley et al. (2011) (middle panel, points) and Bouwens et al. (2015) (bottom panel, blue shaded area) are also given.

At this redshift, the reionization is well advanced in most models except in X0.3 where a 75 per cent ionization level is only achieved. Haloes have been detected using the HOP halo finder (Eisenstein & Hu 1998) and baryons are counted within R_{200}, i.e., the radius of the spherical region around each halo with an average density 200 times greater than the average cosmic matter density. So far, ∼450 haloes with a mass greater than 10^8 h^{-1} M$_\odot$ (corresponding to 45 particles) have been found. Clearly a significant scatter can be

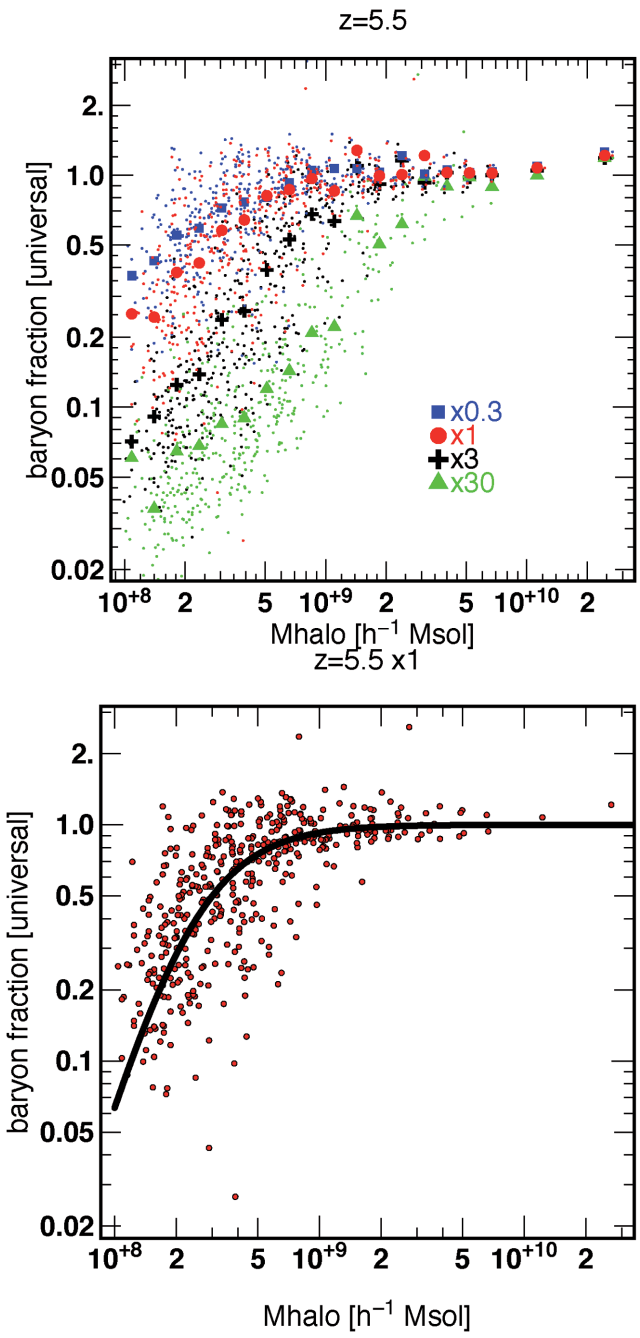

Figure 19. Top: Baryon fraction in DM haloes as a function of their mass at $z = 5.5$ computed in the four models of emissivity. Small dots stand for the values for each individual halo whereas large symbols stand for the average baryon fraction within a bin of halo mass. Bottom: The same quantity but for the fiducial model only (dots) compared to the Okamoto et al. (2008) fit.

found in the distribution of the baryon fraction but general trends can nevertheless be observed in the data: haloes with a mass greater than $10^9\,M_\odot$ basically present a universal fraction whereas lighter objects are more dominated by DM as expected. A comparison of the fiducial model distribution to the fit provided by Okamoto, Gao & Theuns (2008) shows a reasonable agreement with a correct transition mass at $M \sim 3 \times 10^8\,M_\odot$, even though a significant scatter is obtained. If all the models are considered, a clear trend can be noted: the dimmest models (X0.3 and X1) share the same global behaviour (or qualitative functional form) even though the

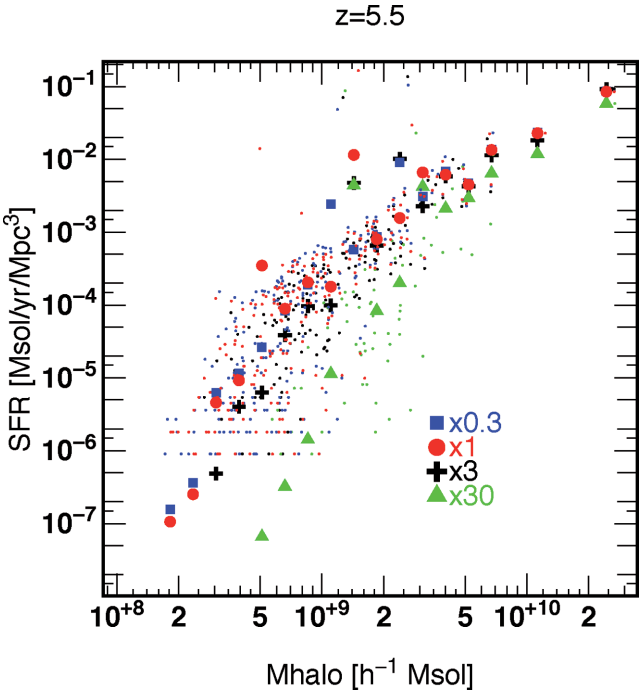

Figure 20. Instantaneous SFR per DM halo mass for the four emissivity models. Small dots stand for the values for each individual halo whereas large symbols stand for the average baryon fraction within a bin of halo mass.

fiducial model presents baryon-poor low-mass haloes. Meanwhile, the brightest models can produce baryon fractions 10 times smaller than the fiducial case with a different functional relation between the baryon fraction and the halo mass. The impact of radiation on this quantity seems, therefore, well established in this series of models, where brighter sources have a strong impact on the gas within shallow potentials.

Interestingly, this impact does not directly translate into a modified SFR inside the haloes (see Fig. 20). Again the scatter is quite important and finite mass effects can be seen in haloes with small formation rates and the interpretation can therefore be difficult. Still, it appears that the three dimmest models (X0.3, the fiducial X1 and X3) are not significantly different and present the same mass dependence of the SFR within their haloes. Since we found that the global baryon quantity is indeed affected, it seems to imply that the *star-forming* baryons are unaffected by the source emissivity and the presence of radiation. Only the most extreme case of source emission shows a significant dip in the SFR of low-mass haloes: in our simplistic model of star formation, a certain level of gas depletion must be achieved to impact the production of stellar particles.

As a final note, we present in Fig. 21 the halo baryon fraction in the four models at the same cosmic average ionization fraction, $x = 0.75$. Of course, this level of ionization is achieved at high redshift ($\sim$10) for the brightest model and corresponds to the last snapshot at $z = 5.5$ for the dimmest one. It can be easily seen that the baryon fraction mass distribution is essentially identical in the four models, taken at four different redshifts but at the same ionization level. It could hint that an essential ingredient of the baryon depletion is not only the source intensity but also the exposition duration to the UV background. In the previous analysis at $z = 5.5$, not only does the brightest model contain the brightest sources but it also provided the longest duration over which haloes are in an optically

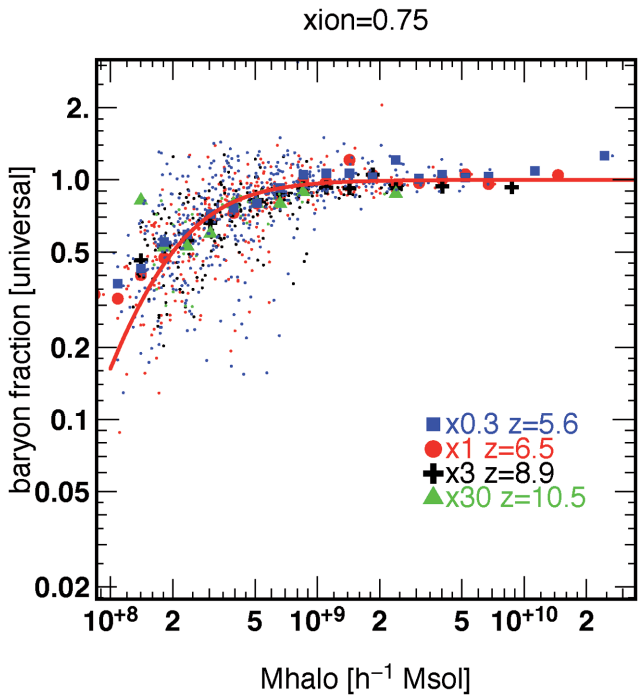

Figure 21. Baryon fraction as a function of halo mass for the four different models measured at the same ionization fraction $x = 0.75$. The corresponding redshifts are given in the labels. The red line stands for the Okamoto et al. (2008) fit taken at the redshift of the fiducial model (X1).

thin universe since such models provide an early reionization. Conversely, dimmer models produce a late reionization and therefore a shorter exposition duration to the UV flux in a transparent universe. It could impact the baryon fraction in low-mass halo measured at a given time. At a given average ionization fraction, we somehow get rid of the scatter in flux exposition and look at haloes from different simulations at a similar stage of their universe ionization history, with a similar structure for the UV field. And indeed, in our model, it significantly reduces the differences observed previously.

Recall that several important ingredients are missing in our models, like supernova feedback, which may enhance the SFR suppression in low-mass haloes, and the presence of H_2, the fraction of which can greatly differ from the baryon fraction. Hence, the results presented in this section indicate in a qualitative manner that EMMA is able to handle cosmological reionization simulations. Further investigations and implementations are necessary to assess quantitatively the subjects discussed here.

7 PERFORMANCE

7.1 Preamble

As a closing chapter to this description of EMMA, we now discuss the performance of the code. As shown hereafter, the comparison of performance on different architectures is a complex matter as it depends on how architecture-dependent optimizations are implemented. Such a comparison also depends on the context in which it has been performed: as such it will evolve in time (as hardware improves for instance) or in space (from one computer to another at a given time). We nevertheless think that the following will shed some light on how the code behaves and how this behaviour can vary significantly depending on the compiler or the architecture. More generally, it is also an opportunity to demonstrate that code performance should be carefully considered, and not only for EMMA.

In the following, the calculations were for a 4 Mpc $h^{-1}/128^3$ cosmological simulation with full physics and the same parameters and simple star formation recipe as the ones described in Section 6.6.1. They used CRTA with speed of light $c/10$, which should not affect the discussion on raw performance and scaling issues. We compared three types of EMMA binaries on the Curie-CCRT supercomputer hybrid nodes, compiled using single-precision arithmetic:

(i) A GPU binary produced by the NVCC compiler from the CUDA 5.5 SDK with O2 optimization level, called *gpu-O2*. This version runs the vectorized physical engines on M2090 Nvidia GPU devices while still relying on a single core to perform the other tasks, such as AMR logistics, vectorization, particle operations, etc. For a multi-GPU run, each MPI process is attached to a single CPU core associated with a single distinct GPU.

(ii) A CPU binary produced by GCC 4.4.7, called *gcc-O2*. This version wholly runs on 2.7 GHz Sandybridge Westemere processors and uses the standard O2 optimization level.

(iii) A CPU binary produced by ICC 14.0.3. with the same O2 optimization level, called *icc-O2*. Such EMMA binaries are usually faster than those provided by GCC by a factor close to four: this difference is essentially the result of optimizations on floating-point operations that are enabled by default. Such optimizations can be disabled by setting an additional `fp-model=strict` flag to produce EMMA binaries with performance reduced to the level of those produced by gcc (not shown here).

ICC is available in most supercomputing and institutional facilities. GCC, on the other hand, is widely distributed and could be the only option on small configurations (e.g., on laptops, desktop machines or local shared memory calculators). Since they produce binaries with different performance, the resulting GPU acceleration will also depend on the CPU version taken as a reference.

Comparisons of GPUs and CPUs are done by considering one graphics device against one CPU *core*. Obviously, this biases performance in favour of GPUs, which are essentially parallel devices. Nevertheless, we argue that this is the simplest way to do the comparison, since a given GPU can be associated with a variety of different CPU nodes with different numbers of cores. However, some care must be taken when considering acceleration rates. If an acceleration factor of $\times$ 80 is found, it should be seen as considerable since 80 CPU cores per GPU is already a significant configuration and codes do not usually follow strong scaling laws at such levels of acceleration (i.e., a $\times$ 80 acceleration cannot be obtained using 80 cores). On the other hand, if a $\times$ 4 acceleration factor is found, it should be considered as low since four cores are easily obtained and $\times$ 4 strong scaling factors can usually be achieved. For Curie hybrid nodes, one GPU is associated with four cores but other configurations exist (e.g., Titan-ORNL associates 16 cores with one GPU).

7.2 Computing time consumption

Fig. 22 presents the time spent by a calculation to reach a given expansion factor in a cosmological simulation. Whichever code version is considered, two major phases can be distinguished: for an expansion factor $a = (1 + z)^{-1} < 0.065$, the code achieves a stable regime with small and regular time steps (given by the slope of the Fig. 22 curves). At this stage, no source has been created yet and no light has to be propagated: the radiative engine (which also includes thermo-chemistry modules such as cooling

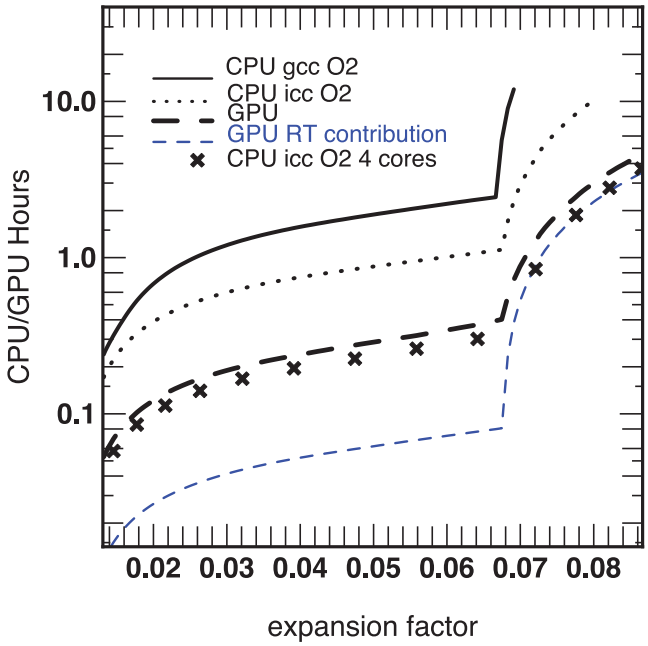

Figure 22. Comparison of the cumulative time spent to reach a given expansion factor for a 4 Mpc $h^{-1}/128^3$ cosmological simulation of the reionization. Times are given for a single computing device (i.e., one GPU or one CPU core). The thick black dashed line stands for the GPU run performed on a M2090 Nvidia GPU whereas the thin dashed blue line stands for the contribution of RT to this cost. The black solid (respectively, dotted) line stands for a single CPU core (2.7 GHz Sandybridge Westemere) using the *gcc-O2* (respectively, *icc-O2*) binary. The symbols stand for a four-core CPU calculation using *icc-O2* on a Curie node.

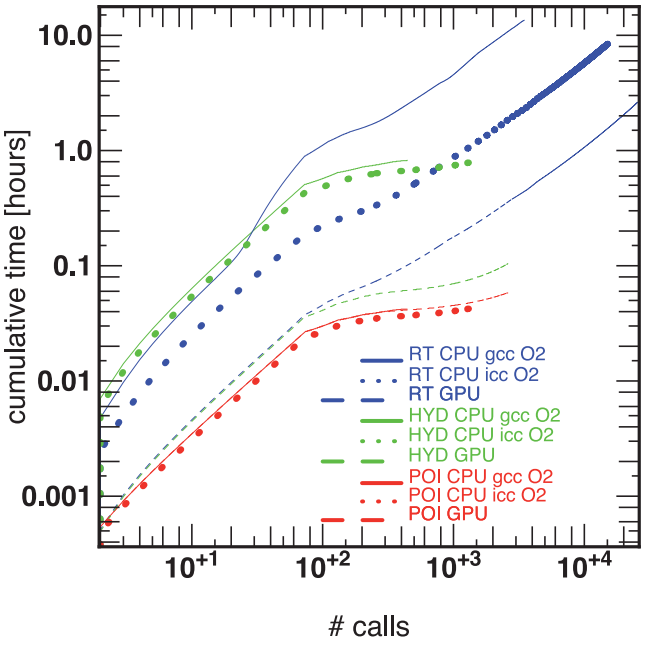

Figure 23. Cumulative time spent in the Poisson (red), hydrodynamics (green) and RT (blue) physical engines as a function of the number of calls. Solid lines stand for single-core CPU calculations on a 4 Mpc/128^3 simulation using *gcc-O2*, dotted lines stand for single-core CPU calculations using *icc-O2* and dashed lines for calculations driven by a single M2090 GPU using *gpu-O2*.

processes) is not limited by the CFL condition and is called once per dynamical time step. Furthermore, the non-linearities are small and AMR has not been deployed yet, hence the work per coarse cell is naturally close to balance. At $a \sim 0.065$, the first source appears and radiative transport must be computed while satisfying the stringent CFL condition. The number of RT calls per dynamical time step increases to typical levels of 150 calls per step. In Fig. 22, the time spent increases by orders of magnitude with a much greater slope, i.e., a much greater time spent per time step. This contribution of RT to the computing time is further emphasized by the dashed blue line in Fig. 22, which stands for the RT-only contribution in the *gpu-O2* calculation (similar curves are obtained for the CPU calculation albeit not shown here): clearly the dramatic increase in the computing time is driven by this specific module.

In the same plot, solid lines stand for the computing time required for EMMA to reach a given expansion factor using a M2090 Nvidia GPU device with *gpu-O2* (black dashed line) and using a single CPU core with *gcc-O2* (black solid line) and *icc-O2* (black dotted line) binaries. Comparing these different versions of EMMA, it can be seen in Fig. 22 that for the *gpu-O2* version, $a = 0.07$ is achieved in 50 min, whereas 16 h are required for the *gcc-O2* version, providing a $\times$ 16.9 acceleration factor. In the pre-source regime (for $a < 0.065$), this acceleration factor drops to $\times$ 6.4: in this regime the contribution of the RT engine is much smaller and so is the level of potential acceleration. In the very first stages of the calculation, this acceleration rate even drops further (to a factor close to $\times 4$) as the cooling induced by dynamical effects is small and hence the need for associated calculations that could have benefited from hardware acceleration. If the GPU version is compared to the *icc-O2* run, the maximal acceleration rate of the *gpu-O2* code drops to $\times 3.9$. Clearly the removal of strict value-safe floating-point operations (which allows greater optimization from the CPU compiler) results in a more competitive CPU code performance-wise. Moreover, note that the current comparisons are for a GPU device against a single CPU core as we argue that this provides the simplest comparison. However, cores are usually part of multi-core nodes, connected to one or two GPU devices. Hence, an acceleration rate of a few times can be seen as not sufficient if it does not exceed the core per GPU ratio. For instance, we show in Fig. 22 the time required for the four cores of a hybrid Curie *node* to run the same test (symbols). As can be seen here, the strong scaling behaviour of EMMA is sufficient to improve further the CPU consumption by a factor of almost 4 and the parallel *icc-O2* binary slightly outperforms *gpu-O2*. As described in the next section, this diversity of performance results from the impact of CPU optimization and GPU acceleration, which are not uniformly distributed among the different modules.

7.3 Detailed computing cost breakdown

For the same experiments, Fig. 23 presents the cumulative time spent in the three principal modules (Poisson, hydrodynamics and radiative solver) of EMMA as a function of the number of successive calls made to these modules. Solid lines stand for the single-core CPU calculation obtained with *gcc-O2*, dotted lines stand for single-core CPU calculations produced by *icc-O2* and dashed lines stand for the GPU-driven experiments.

Focusing first on the *gcc-02* results, it clearly appears that hydrodynamics and RT calculations dominate the overall time budget of EMMA. Unsurprisingly, the Poisson solver only contributes marginally to the overall cost: first, the amount of calculation involved in this stage is small compared to the complex

Table 1. Typical time spent (in seconds) in the vectorization plus transfer operations (VT) and in calculations (Cal) for the hydrodynamics (HYD) and radiative transfer (RT) modules. Times are given for time step 10 of the benchmark simulation described in Section 7, corresponding to a regime without sources and without AMR.

	HYD VT	HYD Cal	RT VT	RT Cal
icc-O2	1.2	26.4	0.9	13.1
gpu-O2	1.76	0.69	1.66	0.85

hydrodynamical solvers or thermo-chemistry calculations. Secondly, it relies on an iterative solver, where the solution does not evolve quickly from a time step to another or only in a few hyper-refined cells, ensuring a rapid convergence and hence a low computational cost. Also note that the hydrodynamics are the dominant stage at early times, being overtaken by RT only as thermo-chemical computations start to contribute and obviously at later time when CRTA executes ∼150 RT calls per hydrodynamics call. This effect due to CRTA is also evident in the number of RT calls, which is much greater than the identical number of hydrodynamics and Poisson calls.

Looking at the performance of the GPU-driven binaries *gpu-O2*, the time spent in the hydrodynamics and RT is reduced to the levels of the Poisson solver: compared to *gcc-O2*, the RT module is accelerated by a factor ×32 and the hydrodynamics by a factor ×14. Interestingly, we could not achieve any acceleration with the Poisson solver on GPU architecture. The reason is the poor computation/transfer ratio for the Poisson solver: our measurements show that gathering the data from AMR to the vector-like structure on the CPU takes ∼75 per cent of the time required by the Poisson solver. The room for acceleration is therefore extremely small, whereas in hydrodynamics and RT this gathering stage only represents ∼5–10 per cent of the computation. In general, the acceleration can be efficient on computation-dominated modules, and in our implementation, the Poisson iterative solver does not belong to this family of functions and represents, therefore, an intrinsic limit to EMMA's performance on GPUs.

Finally, dotted lines show the cumulative time per call of a given module using *icc-O2*. No differences can be noted for the hydrodynamics and Poisson solvers compared to the timings obtained for *gcc-O2*, but the time spent in the RT module is greatly reduced. This is easily explained by the important contribution of non-trivial mathematical operations in cooling rates, cross-sections, ionization rates, etc. in the thermo-chemistry operations handled by the RT module. Such operations clearly benefit from the optimizations made by the compiler. This is not the case for hydrodynamics: even though a MUSCL scheme involves a great number of operations, they essentially rely on simple arithmetic operations, which are less prone to optimization. For hydrodynamics, *gpu-O2* still provides a × 12 acceleration compared to *icc-O2* but the RT acceleration rate drops to × 5.5: since it is the dominant process, this strongly affects the overall GPU acceleration.

The current limiting factor of GPU performance is the cost of vectorization and data transfer to and from the device. In fact, the near identical floor performance obtained by the three GPU modules is due to the irreducible cost of these operations. In Table 1, we list the time consumption for the vectorization/transfer stages as well as for the actual computations, measured in a typical early-stage step for *icc-O2* and *gpu-O2* binaries. The results for the Poisson solver are not discussed as they are already dominated by vectorization on the CPU. We find that for the hydrodynamics and RT, the cost

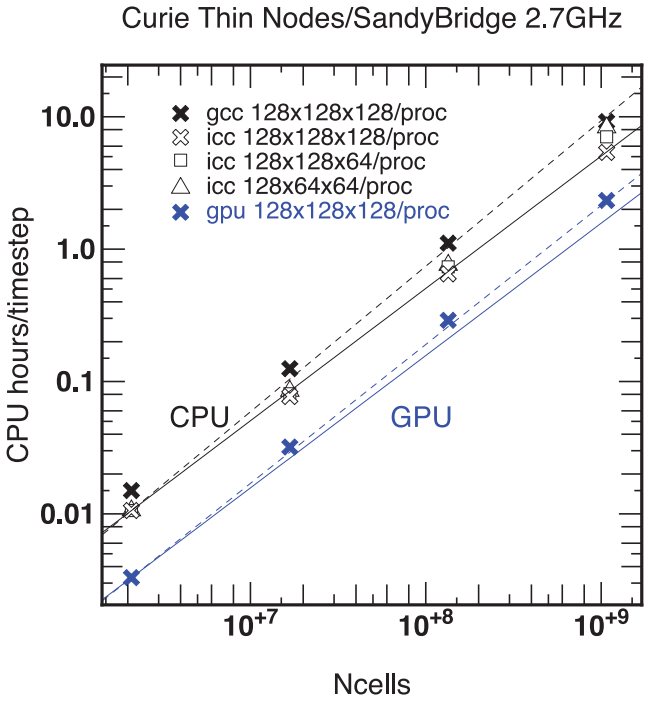

Figure 24. The empty symbols are scaling curves for different loads per process configuration using the *icc-O2* binary, given as the CPU hours per time step as a function of the total number of coarse cells. The black solid line represents perfect CPU scaling (i.e., $t \sim N_c^{\alpha}$ with $\alpha = 1$) while the black dashed line stands for the worst scaling law obtained here with $\alpha = 1.1$. Filled black symbols stand for the same measurements but made with the *gcc-O2* binary and a constant load of 128^3 per process. Filled blue symbols stand for the same measurements on GPUs (using *gpu-O2* binaries) and a constant load of 128^3 per process. The blue solid line stands for the perfect scaling expected for a GPU whereas the dashed blue one stands for the actually measurement with $\alpha = 1.05$.

of these operations is close to 70 per cent on GPUs (see Table 1) whereas they contribute to less than 10 per cent of the CPU calculation: the acceleration potential of *calculations* is thus almost fully exhausted. Also note from Table 1 that these vectorization/transfer steps are actually more expensive on a GPU because of the additional transfer of data from the CPU host to the GPU: the cost of transfers is broadly equivalent to the vectorization. It doubles the time spent in non-calculation operations, which end up dominating the cost of the hydrodynamic and RT modules.

7.4 Parallel scaling

Fig. 24 shows the scaling properties for EMMA, where a fixed load per process is chosen and the number of processes is increased, thus increasing the volume and total number of coarse cells or particles handled by the code. For the *gpu-O2* and *gcc-O2* scaling measurements, we stick to a load of 4 Mpc/128 × 128 × 128 coarse cells per process, similar to the one used above. For the *icc-O2* Intel CPU scaling, we varied the load per core and used non-cubical sub-domains. Configurations from 1 to 256 GPUs and from 1 to 2048 cores were used. Times were measured in the initial stages of a cosmological run, during the first 20 steps. At these early stages, the non-linearities are weak and AMR is not triggered: load imbalance is minimal and therefore this allows a better estimation of parallelism-induced deviations. It also corresponds to the epoch where the acceleration is the weakest but it should not affect our

conclusions regarding the scaling abilities of the code. Clearly the (weak) scaling trends are satisfying with a CPU/GPU computing time that scales as $t \sim N^{[1\text{-}1.1]}$ where N is the number of cells. In fact, on a CPU the scaling is almost perfect in configurations with a $128 \times 128 \times 128$ load per process and drifts away for smaller problems per process: this is a standard strong scaling issue, where a smaller local load increases the weight of the overheads for parallelism. Furthermore, as the sub-domains become non-cubical, the number of neighbours and therefore the amount of communication varies from one process to another if a Peano–Hilbert segmentation is used, leading to a small unbalance of communications between processes. Nevertheless, the scaling of EMMA running on multiple CPUs and GPUs seems satisfying, aside from load-balance issues that will inevitably arise later as structures emerge. We plan to implement balancing procedures in forthcoming developments.

7.5 Discussion

Overall, the performance achieved by GPU-driven calculations is promising and acceleration rates greater than $\times 16.9$ can be obtained, but such rates should be discussed as comparing performance is complex. First the large acceleration rates are obtained in the regime where CRTA is used and effective (i.e., after the first star has appeared): CRTA is somewhat designed to favour hardware accelerators as it increases the weight of pure and heavy calculations on the overall budget of EMMA. Furthermore it remains an approximation with a coarse description of radiative transport; hence, it stands as a lower-order approximation compared to the standard AMR coupling of radiation to matter. Note that even in the CRTA regime, this standard coupling is naturally enforced in the pre-source stages, and we demonstrated that only moderate acceleration is achieved in this regime ($\times 6.4$). Secondly, we also demonstrated that properly optimized CPU binaries may be only a few times slower than GPUs. The overall code acceleration rate drops then to $\times$ 3.9 even in the CRTA-dominated regime, as can be seen by examining Fig. 22. Finally, the performance gap can be reduced further by increasing the number of CPU cores used or by using full node capacities.

Note that these acceleration are global, i.e., they rely on global timings of EMMA where a significant number of modules remain to be ported on GPUs. This is, for instance, the case for particle-related operations, which are currently only handled by the CPU. Even if they are sub-dominant, optimizing these tasks or porting them to a GPU architecture could provide a moderate additional acceleration.

However, the most obvious way to increase the GPU acceleration is to reduce the cost of vectorization and data transfer to the device. The data transfer bottleneck is expected to evolve naturally as new standards are being developed to increase the CPU to GPU bandwidth[4] or by using architectures such as AMD's Accelerated Processor Units (APUs) in which the CPU and GPU share the same memory, thus nullifying the cost of transfers. Note that thanks to the vectorization strategy of EMMA, future ports to new architectures should be of limited complexity. Regarding the cost of the vectorization (driven by gather/scatter operations), note that this operation is currently purely sequential and dealt with by the CPU. Gather/scatter operations could in principle be parallelized to reduce their imprint on the overall costs and to lower the intrinsic floor that limits GPU performance. Such parallelization is limited by concurrent memory accesses but current CPU architectures designed with non-uniform memory access (NUMA) should, in principle, alleviate this issue. Another option would be to deport the vectorization to the parallel computing device (a GPU in our case): performance gains are expected to be limited, since gather/scatter operations rely on non-GPU-friendly tree-walk operations, but even a weak acceleration of the vectorization process would provide a welcome boost to the calculation acceleration.

The tests presented here were made on CPU and GPU architectures of the same generation (M2090+2.7 GHz Westemere on Curie), but newer hardware is and will be available and similar tests will be necessary to reassess the results shown here. However, preliminary tests on more recent devices show no significant improvement in performance (see Appendix B). This is not a complete surprise since our GPU calculations are currently limited by vectorization and transfers and the more recent hardware has not made significant progress for these.

On a broader perspective, the results presented here seem to make a strong case for the full use of hybrid installations, where EMMA could distribute its different tasks/modules simultaneously on the different types of hardware (multi-core CPU and GPU) available on a node. For instance, gather/scatter operations on a CPU represent typically 5–10 per cent of the time spent in a physics engine, hence there is room for potential acceleration on a multi-core node through OpenMP directives, probably of a factor of a few times, making it competitive with current GPU accelerations. For instance, multiple tasks could be done in parallel, such as, e.g., thermo-chemistry on a multi-core CPU and radiative transport on a GPU so that current performance could be increased by an overall factor of 2. Of course, these estimates need to be confirmed by experience.

8 CONCLUSIONS

EMMA is a cosmological simulation code, which handles simultaneously gravity, hydrodynamics and RT on an adaptive grid that can be refined on the fly (AMR). Written in C, this code is parallel (via the MPI protocol) and can deploy its physics modules on GPUs using CUDA. Designed to study the reionization epoch, EMMA is, nevertheless, a versatile code for structure formation.

The code has passed a variety of test cases and can confidently produce accurate and relevant simulations. A first comparison of cosmological reionization simulations with different source parameters presents the expected qualitative behaviour of the physics at play.

EMMA has been tested in a wide variety of parallel configurations and it demonstrates satisfying scaling properties. It is able to use GPUs to accelerate hydrodynamics and RT calculations. Depending on the optimizations and the compilers used to generate the CPU reference, global GPU acceleration factors between $\times 3.9$ and $\times 16.9$ can be obtained. Vectorization and transfer operations currently prevent better GPU performance, and we expect that future optimizations and hardware evolution will lead to greater accelerations. Overall we demonstrate that EMMA is able to cope efficiently with a variety of hardware.

Aside from optimization to improve the performance of the code and GPU-driven acceleration factors, additional features will be included in EMMA in the near future. Among them, star formation and supernova feedback are a major priority as they are an essential ingredient for galaxy formation theories and models. Their implementation is on the way and will be the described in a forthcoming paper. Molecular chemistry is envisioned too, as it is physically relevant for understanding the formation of the first and smallest objects during the reionization epoch (like mini-haloes with $M < 10^7\,\mathrm{M}_\odot$)

[4] http://www.nvidia.com/object/nvlink.html

and also numerically interesting as such calculations can be easily accelerated. Full documentation of the code is also on the way for a public release of EMMA in a finite, maybe short, term.

ACKNOWLEDGEMENTS

We are grateful to B. Semelin, N.Gillet, J. Blaizot, J. Rosdahl, C. Pichon, R. Teyssier, T. Stranex and P. Shapiro for discussions over the years that provided the basis for the EMMA code. We are also grateful to the anonymous referee who helped to improve the quality of the article. This work has been supported by Agence Nationale de la Recherche grants, ANR-12-JS05-0001 (EMMA), ANR-14-CE33-0016-03 (ORAGE) and ANR-09-BLAN-0030 (LIDAU). This research used resources at the Oak Ridge Leadership Computing Facility (INCITE 2013 Award AST031, INCITE Prep project AST105), which is a Department of the Environment Office of Science User Facility supported under Contract DE-AC05-00OR22725, at the Meso-Centre de l'Universite de Strasbourg and at the Centre de Calcul Recherche et Technologie-CCRT (Curie-CPU and Curie-GPU, DARI Grant 2015047393). The authors thank D. Munro for freely distributing his YORICK[5] programming language and its yorick-gl extension.

REFERENCES

Aubert D., Teyssier R., 2008, MNRAS, 387, 295
Aubert D., Teyssier R., 2010, ApJ, 724, 244
Aubert D., Amini M., David R., 2009, Lecture Notes in Computer Science Vol. 5544, Springer Science+Business Media, Berlin, p. 874
Baek S., Di Matteo P., Semelin B., Combes F., Revaz Y., 2009, A&A, 495, 389
Baek S., Semelin B., Di Matteo P., Revaz Y., Combes F., 2010, A&A, 523, A4
Barkana R., Loeb A., 2001, Phys. Rep., 349, 125
Bertschinger E., 1985, ApJS, 58, 39
Bouwens R. J. et al., 2015, ApJ, 803, 34
Calverley A. P., Becker G. D., Haehnelt M. G., Bolton J. S., 2011, MNRAS, 412, 2543
Chardin J., Aubert D., Ocvirk P., 2012, A&A, 548, A9
Ciardi B., Stoehr F., White S. D. M., 2003, MNRAS, 343, 1101
Dubinski J., 1996, New Astron., 1, 133
Dubois Y., Teyssier R., 2008, A&A, 477, 79
Eisenstein D. J., Hu W., 1998, ApJ, 496, 605
Eisenstein D. J., Hut P., 1998, ApJ, 498, 137
Fan X. et al., 2006, AJ, 132, 117
Finlator K., Davé R., Özel F., 2011, ApJ, 743, 169
Gnedin N. Y., 2000, ApJ, 542, 535
Gnedin N. Y., 2014, ApJ, 793, 29
Gnedin N. Y., Abel T., 2001, New Astron., 6, 437
González M., Audit E., Huynh P., 2007, A&A, 464, 429
Hockney R. W., Eastwood J. W., 1981, Computer Simulation Using Particles, McGraw-Hill, New York
Hoeft M., Yepes G., Gottlöber S., Springel V., 2006, MNRAS, 371, 401
Hui L., Gnedin N. Y., 1997, MNRAS, 292, 27
Iliev I. T. et al., 2006, MNRAS, 371, 1057
Iliev I. T. et al., 2009, MNRAS, 400, 1283
Katz N., Weinberg D. H., Hernquist L., 1996, ApJS, 105, 19
Kay S. T., Pearce F. R., Frenk C. S., Jenkins A., 2002, MNRAS, 330, 113
Khokhlov A., 1998, J. Comput. Phys., 143, 519
Kravtsov A. V., Klypin A. A., Khokhlov A. M., 1997, ApJS, 111, 73
Levermore C. D., 1984, J. Quant. Spectrosc. Radiat. Transfer, 31, 149
Martel H., Shapiro P. R., 1998, MNRAS, 297, 467
McQuinn M., Lidz A., Zahn O., Dutta S., Hernquist L., Zaldarriaga M., 2007, MNRAS, 377, 1043
Mellema G., Iliev I. T., Pen U.-L., Shapiro P. R., 2006, MNRAS, 372, 679
Norman M. L., Reynolds D. R., So G. C., Harkness R. P., Wise J. H., 2015, ApJS, 216, 16
Ocvirk P., Aubert D., Chardin J., Knebe A., Libeskind N., Gottlöber S., Yepes G., Hoffman Y., 2013, ApJ, 777, 51
Ocvirk P. et al., 2014, ApJ, 794, 20
Okamoto T., Gao L., Theuns T., 2008, MNRAS, 390, 920
Paardekooper J.-P., Khochfar S., Dalla Vecchia C., 2013, MNRAS, 429, L94
Pawlik A. H., Schaye J., Dalla Vecchia C., 2015, MNRAS, 451, 1586
Planck Collaboration XVI et al., 2014, A&A, 571, A16
Pontzen A., Governato F., 2012, MNRAS, 421, 3464
Pritchard J. R., Loeb A., 2012, Rep. Progress Phys., 75, 086901
Prunet S., Pichon C., Aubert D., Pogosyan D., Teyssier R., Gottloeber S., 2008, ApJS, 178, 179
Rasera Y., Teyssier R., 2006, A&A, 445, 1
Rosdahl J., Blaizot J., Aubert D., Stranex T., Teyssier R., 2013, MNRAS, 436, 2188
Sheth R. K., Mo H. J., Tormen G., 2001, MNRAS, 323, 1
Teyssier R., 2002, A&A, 385, 337
Theuns T., Leonard A., Efstathiou G., Pearce F. R., Thomas P. A., 1998, MNRAS, 301, 478
Toro E. F., 1997, Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction. Springer, Berlin, New York
Toro E. F., Spruce M., Speares W., 1994, Shock Waves, 4, 25
Trac H., Cen R., 2007, ApJ, 671, 1
Trac H. Y., Gnedin N. Y., 2011, Advanced Sci. Lett., 4, 228
Warren M. S., Salmon J. K., 1993, A Parallel Hashed Oct-Tree *N*-Body Algorithm, ACM, New York
Wise J. H., Abel T., Turk M. J., Norman M. L., Smith B. D., 2012, MNRAS, 427, 311
Zahn O., Lidz A., McQuinn M., Dutta S., Hernquist L., Zaldarriaga M., Furlanetto S. R., 2007, ApJ, 654, 12
Zawada K., Semelin B., Vonlanthen P., Baek S., Revaz Y., 2014, MNRAS, 439, 1615
Zel'dovich Y. B., 1970, A&A, 5, 84

[5] http://dhmunro.github.io/yorick-doc/

APPENDIX A: ADDITIONAL TESTS ON COSMOLOGICAL SIMULATIONS

A1 Mass function in pure DM simulations

First, we try to recover the halo mass function in pure DM cosmological simulations. Initial conditions were produced with the MPGrafic package (Prunet et al. 2008) for a 100 h^{-1} comoving Mpc box sampled with 256^3 particles. The gravitational potential is computed on a 256^3 coarse grid ($\ell = 8$) and refinement up to $\ell = 12$ is triggered in a quasi-Lagrangian manner when a cell contains more than eight particles. The spatial resolution is equivalent to a 4096^3 grid. Cosmological parameters were taken from Planck Collaboration XVI et al. (2014) (setting $\Omega_m = \Omega_c$) and used as inputs to the Eisenstein & Hu (1998) transfer function. The haloes were detected using the HOP halo finder (Eisenstein & Hut 1998) and their mass function is compared to the formulation of Sheth, Mo & Tormen (2001). Only haloes with more than 10 particles were kept, corresponding to a minimal mass of $5.2 \times 10^{10}\,M_\odot$.

Fig. A1 presents the halo mass function obtained at different redshifts, directly compared to Sheth et al. (2001). A good agreement is obtained at all redshifts, with massive haloes kicking in only at later time as expected. On the low-mass end of the mass function, EMMA is complete for haloes with at least ∼500 particles and a factor of 2 below full completeness for haloes with ∼100 particles. These numbers are standard for such AMR codes and overall we can conclude that EMMA tracks correctly the assembly history of DM haloes.

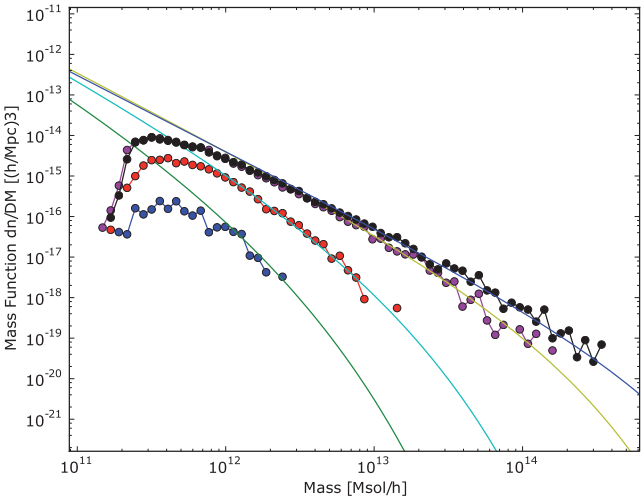

Figure A1. Halo mass function of a pure DM 100 h^{-1} Mpc/256^3 cosmological run. Dots stand for the mass functions measured in the simulation at $z = 5.4$, 3.3, 1.0 and 0.0 (from bottom to top). Lines stand for the Sheth et al. (2001) expression for the halo mass function.

A2 Energy conservation in DM plus gas adiabatic simulations

In this section, we probe the energy conservation of an adiabatic cosmological run (DM plus gas). In super-comoving variables, the cosmic energy varies between expansion factors a_1 and a_2 according to

$$(\tilde{T} + \tilde{U} + \tilde{E})|_{a_1}^{a_2} = \int_{a_1}^{a_2} \frac{\tilde{U}}{a} \mathrm{d}a \tag{A1}$$

for a $\gamma = 5/3$ gas. $\tilde{T}$, $\tilde{U}$ and $\tilde{E}$ stand for the total super-comoving kinetic energy, potential energy and internal gas energy. Three 100 Mpc $h^{-1}/128^3$ ($\ell = 7$) simulations with $\ell_{\max} = 7$, 8 and 10 were run, using the same cosmological parameters and refinement strategy as in Section A1. In the same spirit as Kravtsov et al. (1997), we check equation (A1) against the change in potential energy, i.e.:

$$\text{error} = \frac{(\tilde{T} + \tilde{U} + \tilde{E})|_{a_1}^{a_2} - \int_{a_1}^{a_2} \frac{\tilde{U}}{a} \mathrm{d}a}{\tilde{U}|_{a_1}^{a_2}}. \tag{A2}$$

In Fig. A2, the error is shown for three maximum levels of refinement: $\ell_{\max} = 7$ (i.e., no refinement), $\ell_{\max} = 8$ and $\ell_{\max} = 10$. The error is found to be under control at 2, 1.2 and 0.7 per cent, respectively. Note how the three tracks diverge as refinement levels are installed (e.g., $a = 0.12$ for $\ell = 8$ and $a = 0.2$ for $\ell = 9$), providing greater resolution and smaller energy drifts. Overall, these levels of error are consistent with, e.g., Kravtsov et al. (1997) and Teyssier (2002).

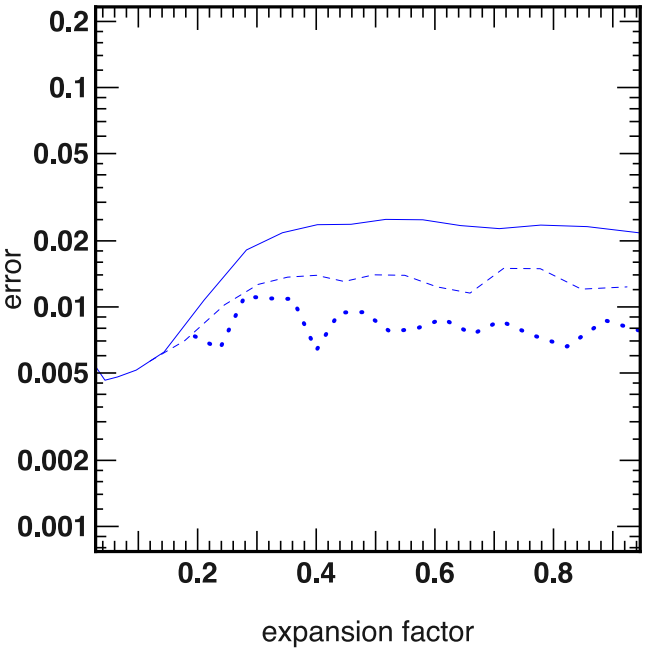

Figure A2. Error on cosmic energy variation as defined in equation (A1) for three 100 Mpc h^{-1} adiabatic DM plus gas cosmological simulations with 128^3 coarse resolutions ($\ell = 7$). From top to bottom, a simulation without AMR (solid), with $\ell_{\max} = 8$ (dashed) and $\ell_{\max} = 10$ (dotted).

APPENDIX B: PRELIMINARY COMPARISON OF EMMA'S PERFORMANCE ON DIFFERENT GPU DEVICES

We briefly describe the timings obtained by EMMA on K20c devices, which are more recent than the M2090 GPUs available on Curie. K20c devices differ by the number and type of cores (2496 Kepler cores versus 512 Fermi cores for the M2090), core clock (706 MHz versus 1.3 GHz for the M2090), memory frequency (2.6 GHz versus 1.9 GHz for the M2090) and bandwidth (208 Gb s^{-1} versus 177 Gb s^{-1} for the M2090). In terms of single-precision floating-point operations, the theoretical peak performance of K20c is a factor of 2 greater than the M2090.

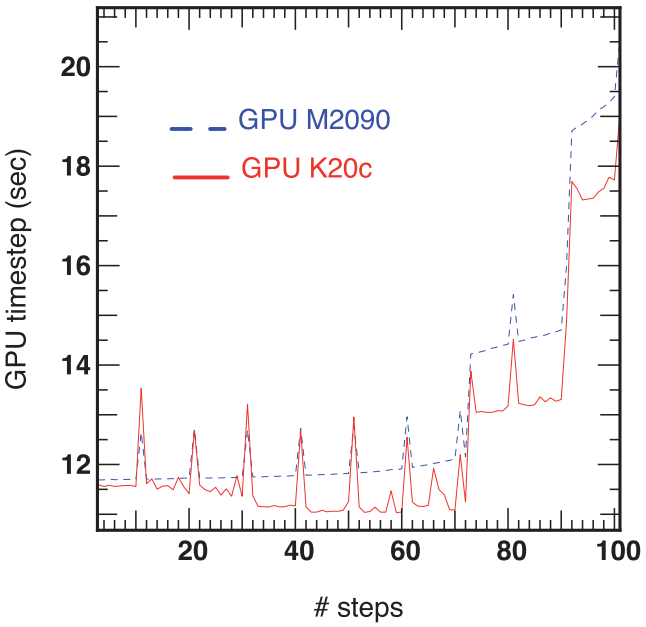

Figure B1. Comparison of the time step duration on two different kinds of GPU device. Measurements were taken on the first 100 steps of a 4 Mpc h^{-1} cosmological simulation with full physics with parameters similar to the runs described in Section 3. The blue dashed line stands for M2090 timings and the red solid line stand for timings on the more recent K20. The spikes seen in both curve are due to input/output operations.

We ran a 4 Mpc h^{-1} cosmological simulation on a single GPU with full physics over 100 time steps, with the same settings as that chosen in Sections 3 and 4. Fig. B1 compares the duration of the time steps obtained from two simulations made on these two

kinds of device. In both cases, the timings show the same global evolution with spikes due to outputs of data and large jumps due to AMR refinement. K20 performs marginally better than M2090, at the 10 per cent level, despite its greater computing power. This is expected since EMMA's calculations on a GPU are already dominated on M2090 devices by gather/scatter and transfer operations from the host to the device. Future devices with greater bandwidth could improve the situation, but in its current state, EMMA does not really benefit from the greater computing power of more recent hardware.

This paper has been typeset from a TEX/LATEX file prepared by the author.