Tree-based solvers for adaptive mesh refinement code FLASH – II: radiation transport module TreeRay

Richard Wünsch [1]*, Stefanie Walch [2], František Dinnbier [1,2,3], Daniel Seifried [2], Sebastian Haid [2], Andre Klepitko [2], Anthony P. Whitworth[4] and Jan Palouš[1]

[1]*Astronomical Institute, Academy of Sciences of the Czech Republic, Bocni II 1401, CZ-141 31 Prague, Czech Republic*
[2]*I. Physikalisches Institut, Universität zu Köln, Zülpicher Str. 77, D-50937 Köln, Germany*
[3]*Charles University in Prague, Faculty of Mathematics and Physics, Astronomical Institute, V Holešovičkách 2, CZ-18000 Praha 8, Czech Republic*
[4]*School of Physics & Astronomy, Cardiff University, The Parade, Cardiff CF24 3AQ, UK*

ABSTRACT

The treatment of radiative transfer with multiple radiation sources is a critical challenge in simulations of star formation and the interstellar medium (ISM). In this paper, we present the novel TREERAY method for solving general radiative transfer problems, based on reverse ray-tracing combined with tree-based accelerated integration. We implement TREERAY in the adaptive mesh refinement code FLASH, as a module of the tree solver developed by Wünsch et al. However, the method itself is independent of the host code and can be implemented in any grid-based or particle-based hydrodynamics code. A key advantage of TREERAY is that its computational cost is independent of the number of sources, making it suitable for simulations with many point sources (e.g. massive star clusters) as well as simulations where diffuse emission is important. A very efficient communication and tree-walk strategy enable TREERAY to achieve almost ideal parallel scalings. TREERAY can easily be extended with sub-modules to treat radiative transfer at different wavelengths and to implement related physical processes. Here, we focus on ionizing radiation and use the on-the-spot approximation to test the method and its parameters. The ability to set the tree solver time-step independently enables the speedy calculation of radiative transfer in a multiphase ISM, where the hydrodynamic time-step is typically limited by the sound speed of the hot gas produced in stellar wind bubbles or supernova remnants. We show that complicated simulations of star clusters with feedback from multiple massive stars become feasible with TREERAY.

Key words: gravitation – hydrodynamics – radiative transfer – ISM: evolution – galaxies: ISM.

1 INTRODUCTION

NEED FOR FAST RT. The turbulent, multiphase structure of the interstellar medium (ISM) is shaped by the complex and non-linear interplay between gravity, magnetic fields, heating and cooling, and the radiation and momentum input from stars, in particular, massive stars (see e.g. Agertz et al. 2013; Walch et al. 2015; Kim & Ostriker 2017; Peters et al. 2017). Therefore, an efficient treatment of radiation transport in different energy bands (from the submillimeter to X-rays), and of the associated heating and cooling processes, is essential to simulate the structure and evolution of the ISM in detail, and to compare theoretical and numerical models with observations. A fundamental consideration is that radiation is emitted from different types of sources: point sources such as stars, extended sources like cooling shock fronts, diffuse sources like dust, as well as an ambient background radiation field. Hence, a modern radiative transfer algorithm must be able to handle multiple energy bands and multiple sources in an efficient manner.

OVERVIEW OF ALGORITHM. There are many ways to treat the radiation from point sources in 3D: Ray tracing (Mellema et al. 2006; Gritschneder et al. 2009) with HEALPIX schemes (Bisbas et al. 2009; Wise & Abel 2011; Baczynski, Glover & Klessen 2015; Kim

et al. 2017; Rosen et al. 2017), long- and short-range characteristics (e.g. Rijkhorst et al. 2006); flux-limited diffusion (FLD) (Krumholz, Klein & McKee 2007; Skinner & Ostriker 2013); combined schemes which work in optically thin and thick regions (Kuiper et al. 2010; Paardekooper, Kruip & Icke 2010; Klassen et al. 2014); moment methods (Petkova & Maio 2012; Rosdahl et al. 2013; Kannan et al. 2019); and backward radiative transfer schemes (e.g. Kessel-Deynet & Burkert 2003a; Altay & Theuns 2013; Grond et al. 2019), like the one developed in this work. In comparison with ray-tracing, FLD- and moment-based methods tend to be computationally less expensive and their cost does not depend on the number of sources. However, typically they do not capture certain features of the radiation field, for example shadowing (although see Rosdahl et al. 2013). Several code comparison projects have highlighted the advantages and short-comings of different radiative transfer methods (e.g. Iliev & et al. 2006; Iliev et al. 2009; Bisbas et al. 2015). Ultimately, even simple radiative transfer methods are expensive, at least as expensive as all the other elements of a simulation – (magneto-)hydrodynamics ((M)HD), self-gravity, chemistry, heating and cooling – together. Consequently, the numerical overhead of radiative transfer severely limits the astrophysical problems that can be addressed realistically in state-of-the-art 3D simulations, even on today's largest super-computers.

TREERAY BASICS. In response to this challenge, we have devised TREERAY, a new tree-based, backward radiative transfer scheme,

* E-mail: richard@wunsch.cz

which can handle multiple sources at acceptable extra cost, when running simulations that include self-gravity anyway. The computational cost of TREERAY is independent of the number of emitting sources (as demonstrated in Section 5.2), and indeed every cell can be a source. Therefore, TREERAY can readily treat, for example, the ionizing radiation from multiple massive stars in a molecular cloud (Haid et al. 2019), as well as radiating shocks or other extended sources, on-the-fly in complex 3D (M)HD simulations. TREERAY is an extension of the octtree-based solver for gravity and diffuse radiative transfer described in detail in Wünsch et al. (2018, hereafter Paper I), which has been developed for the Eulerian, AMR code FLASH 4[1] (Fryxell et al. 2000). The tree-solver of Paper I is available with the official FLASH release. Due to its efficiency, the TREERAY scheme allows us to treat all dynamically relevant radiative processes in full three-dimensional simulations of the multiphase ISM with an acceptably small error. TREERAY evolved from the original TREECOL method (Clark, Glover & Klessen 2012) for treating the shielding of a diffuse background radiation field in smoothed particle hydrodynamics (SPH). The shielding method has been presented as an OPTICALDEPTH module of the tree solver in Paper I.

TREVR. The recently published TREVR code (Grond et al. 2019), which uses a tree data structure and reverse ray-tracing, is, we believe, the most closely related existing code. However, the two codes differ significantly in several respects. TREVR casts rays in the directions of sources, resulting in some (albeit weak) dependence of the computational cost on source number, while TREERAY casts HEALPIX rays (strictly speaking cones) in all directions, thereby covering the whole computational domain. These two fundamentally different approaches result in different types of numerical artefacts. Whilst TREERAY uses iteration to deal with regions irradiated by multiple sources, in situations where the absorption coefficient depends on the radiation energy density, TREVR applies limits to the time-step and uses a special refinement criterion to take account of the directional dependence of absorption in tree nodes. Finally, TREVR is implemented in the SPH code Gasoline, while TREERAY is implemented in the AMR code FLASH, although both codes are general and could, in principle, be adapted to work with any hydrodynamics code.

OUTLINE. The plan of the paper is as follows. In Section 2, we present the general algorithm and describe its implementation and coupling to the tree solver. In Section 3, we perform a suite of static and dynamic tests to demonstrate the viability of the scheme, and we discuss the impact of the user-specified parameters on the accuracy. We restrict ourselves to the simple on-the-spot approximation (Osterbrock 1988) (TREERAY/ONTHESPOT module) for treating the interaction of UV radiation from massive stars with the ISM. The flexibility and performance of TREERAY are demonstrated in Section 4, where we present a more complex simulation of multiple massive stars dispersing a molecular cloud with their ionizing radiation and winds. In Section 5, we discuss the performance and parallel scaling of the algorithm, before summarizing in Section 6.

2 THE ALGORITHM

CODE STRUCTURE. TREERAY is implemented as a module for the *tree-solver* described in Paper I. The tree-solver provides general algorithms for building an octal tree, communicating the required
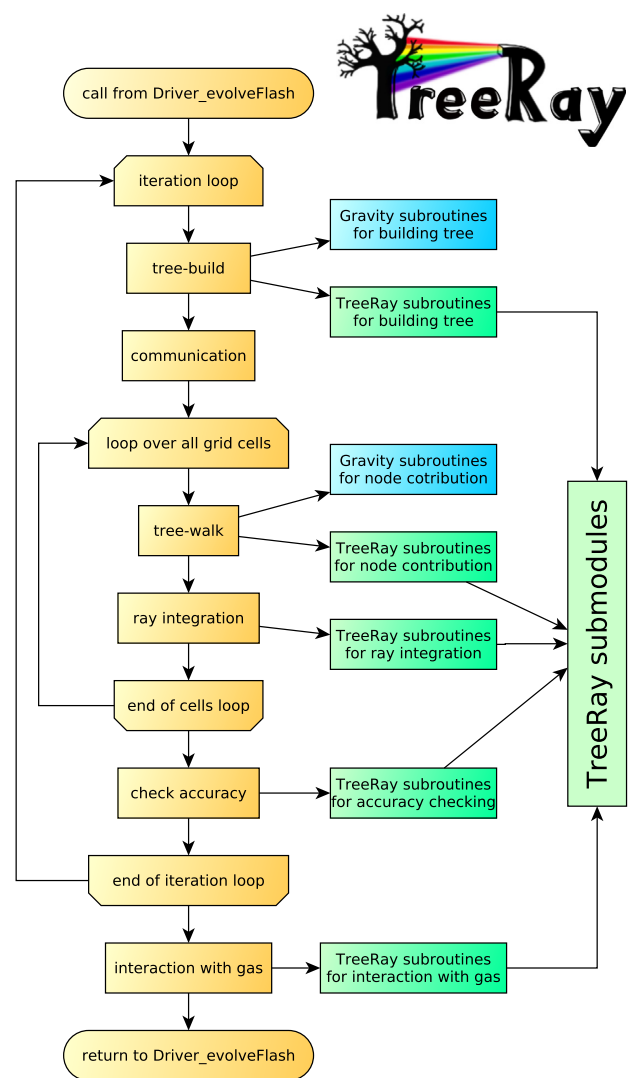
---

[1]The FLASH code is maintained by the ASC/Alliances Center for Astrophysical Thermonuclear Flashes (Flash Center for Computational Science) at the University of Chicago (http://flash.uchicago.edu/site)

Figure 1. The tree solver flowchart and its connection to the GRAVITY and TREERAY modules. All general tree solver routines are yellow. GRAVITY routines are blue. TREERAY routines are green.

parts of it to other processors, and traversing the tree to calculate quantities that require integration over the computational domain. It is a generalization of the widely used algorithm devised by Barnes & Hut (1986) to solve for self-gravity in numerical codes. Individual modules (e.g. GRAVITY or TREERAY) define which quantities should be stored on the tree, and provide subroutines to be called during the tree-walk to calculate integrated quantities like the gravitational acceleration or radiation energy density. TREERAY itself consists of a general part plus submodules that treat the different physical processes needed to solve the radiation transport equation (RTE). The TREERAY/OPTICALDEPTH submodule (described in Paper I) is the simplest TREERAY submodule; instead of solving the RTE, it simply sums contributions from different directions to obtain the corresponding optical depths. Here, we focus on the TREERAY/ONTHESPOT submodule, to illustrate how TREERAY solves the RTE for ionizing radiation.

NEW FEATURES IN TREE SOLVER. The simplified flowchart in Fig. 1 shows the connection between tree solver and TREERAY.[2]

In comparison to Paper I, two modifications have been made. First, the three main steps of the tree-solver (tree-build, communication and tree-walk) can be called several times (outer iteration loop in Fig. 1) to calculate correctly absorption in regions irradiated by more than one source. Second, the tree walk is now called on a cell-by-cell basis (instead of block-by-block), because the temporary data structures storing quantities needed for solving the RTE are too big to be stored in memory for all the grid cells of a block.

ALGORITHM FLOW. The algorithm proceeds as follows. First, the tree is built to store quantities provided by the TREERAY modules, and appropriate parts of the tree are communicated to other processors. Next, the tree is traversed for each cell (hereafter a *target cell*) and these quantities are mapped on to a system of rays originating at the target cell and covering the whole computational domain. Then, the RTE is solved along each ray and the radiative fluxes arriving at each target cell are calculated and converted into a radiation energy density. Finally, the whole process (starting from the tree-build) is repeated until the radiation energy density converges everywhere, i.e. until it does not change by more than a user-defined fraction between successive iterations. Below, we describe individual steps in detail.

2.1 Tree build

QUANTITIES ADDED TO THE TREE. The general TREERAY module does not add any quantities to the tree; the tree-solver itself stores four numbers on each node, its mass and centre of mass. However, the TREERAY submodules typically need to store two or three numbers for each energy band on each tree node (node index n). In the case of the TREERAY/ONTHESPOT submodule, we are concerned with Extreme Ultraviolet radiation (EUV); i.e. radiation that can ionize hydrogen. The module stores the rate of emission of EUV photons within the node ($\varepsilon_{\mathrm{EUV},n}$), the rate of recombination of hydrogen into excited levels ($\alpha_{\mathrm{EUV},n}$), and the mean EUV radiation energy density from the previous iteration step ($e_{\mathrm{EUV},n}$):

$$\varepsilon_{\mathrm{EUV},n} = \max\left(\sum_c \dot{n}_{\mathrm{EUV},c} - \alpha_{\mathrm{B}} n_{\mathrm{H},c}^2 \mathrm{d}V_c, 0\right), \tag{1}$$

$$\alpha_{\mathrm{EUV},n} = \max\left(\sum_c \alpha_{\mathrm{B}} n_{\mathrm{H},c}^2 \mathrm{d}V_c - \dot{n}_{\mathrm{EUV},c}, 0\right), \tag{2}$$

$$e_{\mathrm{EUV},n} = \sum_c e_{\mathrm{EUV},c} \mathrm{d}V_c. \tag{3}$$

Here, $\dot{n}_{\mathrm{EUV},c}$ is the rate at which EUV photons are emitted from hot stars in cell c, $n_{\mathrm{H},c}$ is the number density of hydrogen nuclei, α_{B} is the recombination coefficient into excited states only, $e_{\mathrm{EUV},c}$ is the radiation energy density calculated for cell c in the previous iteration, and $\mathrm{d}V_c$ is the cell volume. Note that the photons emitted and absorbed within cell c are subtracted from each other in equations (1) and (2), as this is more accurate than doing it later, during the RTE integration along rays, when both quantities have been modified by the approximations inherent in mapping the tree nodes on to the rays.

MAPPING SOURCES ON TO THE GRID. In order to obtain $\dot{n}_{\mathrm{EUV},c}$, each source of radiation, i, characterized by the rate at which it emits EUV photons, $\dot{N}_{\mathrm{EUV},i}$ and its radius, r_i, is mapped on to the grid before the tree-build. During the mapping, $\dot{N}_{\mathrm{EUV},i}$ is divided between the grid cells which the source intersects, in proportion to the intersecting volumes.

COMMUNICATION. After the tree is built, it is communicated in the same way as described in Paper I (Section 2.1). In summary, the code first distributes information about all the block positions and sizes to all the processors. Then, each processor runs a tree walk with target points at the closest point to the local node of all the blocks of a given remote sub-domain, and determines which local nodes will be opened when the tree walk is executed at the processor calculating that sub-domain. In this way, the local processor determines lists of the nodes needed at all the remote processors, and sends the nodes there. This ensures that all the information about the nodes to be opened during the tree walk is present at each processor, and at the same time it minimizes the amount of data communicated.

2.2 Structure of rays

RAY-ARRAYS. Before the tree is walked for each target cell, three arrays (cdMaps, rays, and raysEb) are created in order to store the results of the tree-walk. The first one, cdMaps (standing for column density maps), is two dimensional and has size $N_{\mathrm{PIX}} \times l_{\mathrm{q}}$, where N_{PIX} is the number of directions and l_{q} is the number of quantities. The directions are defined with the HEALPIX algorithm (Górski et al. 2005), which tessellates a unit sphere into pixels of equal solid angle, each with a unit vector, $\hat{\boldsymbol{n}}_k$, which points from the sphere centre (target point) to the centre of the pixel. The value N_{PIX} is specified at the outset by setting the HEALPIX level N_{side}, with possible values $N_{\mathrm{side}} = 1, 2, 4, 8, \ldots$, corresponding to $N_{\mathrm{PIX}} = 12N_{\mathrm{side}}^2 = 12, 48, 192, 768, \ldots$. The cdMaps array is used by the TREERAY/OPTICALDEPTH module and has already been described in Paper I.[3]

RAYS. The second array, rays, adds another dimension representing radial distance from the target point. We define N_{r} *evaluation points* on each ray leading from the target point in the direction of vector $\hat{\boldsymbol{n}}_k$. Each ray is associated with a volume given by the area of the HEALPIX pixel extended along the radial direction. Thus we create a system of rays pointing from each target point in N_{PIX} directions, and covering the whole computational domain. The rays array has size $N_{\mathrm{r}} \times N_{\mathrm{PIX}} \times m_{\mathrm{q}}$, where m_{q} is the number of quantities mapped on to the rays.

RAYSEB. The last array, raysEb, adds another dimension, which is necessary in cases involving multiple energy bands. Its size is $(N_r \times N_{\mathrm{eb}} \times N_{\mathrm{PIX}} \times n_{\mathrm{q}}$, where N_{eb} is the total number of energy bands for all sub-modules and n_{q} is the maximum number of quantities per energy band (by default 3). Typically, each sub-module requires one or more energy bands to treat radiation at different wavelengths (or radiation involved in different physical processes), and each energy band uses its own emission coefficient, absorption coefficient and radiation energy density, which are stored in the raysEb array. For simplicity, we refer to all three arrays as ray-arrays, considering cdMaps to be a special ray-array with only one data point in each radial direction.

EVALUATION POINTS ON RAYS. Each ray intercepts N_r evaluation points, where the quantities mapped on to the ray (e.g. emission and absorption coefficients) are set. Evaluation points are located so that the distances between successive points (hereafter ray *segment* lengths), are proportional to the distance from the target cell. This is because the size of tree nodes that need to be opened also increases – often linearly – with distance from the target point. Hence, starting

---

[2]The full scheme of the tree solver and its modules down to the level of individual subroutines can be found at http://galaxy.asu.cas.cz/pages/treeray.

[3]Note that in Paper I this array had an additional dimension going through all the cells within a block, because the tree-walk was run on a block-by-block, rather than a cell-by-cell, basis.

with the first evaluation point at the target point ($r_0 \equiv 0$), the radial coordinate of the ith evaluation point is

$$r_i = \frac{\Delta x \; i^2}{2 \, \eta_R^2}. \tag{4}$$

Here, Δx is the size of the smallest cell in the simulation and η_R is a user-defined parameter that controls the resolution in the radial direction. By default, $\eta_R = 2$, which ensures that (if the BH criterion for node acceptance is used, see Section 2.4) the segment lengths correspond, approximately, to half the size of the nodes with which the target cell interacts during the tree walk, i.e.

$$|r_{i+1} - r_i| \sim \frac{\theta_{\text{lim}} \times d}{2}, \tag{5}$$

where d is the distance between the target cell and the interacting node. The maximum ray length, L_{ray}, is set automatically to the length of the three-dimensional diagonal of the computational domain, and hence the number of evaluation points along each ray is

$$N_R = \eta_R \times \texttt{floor}\left(\sqrt{\frac{2 L_{\text{ray}}}{\Delta x}}\right) + 1, \tag{6}$$

where `floor(x)` is the largest integer less than or equal to x. Each module M includes a user-defined parameter, $L_{\text{ray},M}$, by which the user can specify the maximum distance from the target cell to which the calculation (mapping and/or RTE solution) should be carried out.

2.3 Tree-walk

MAPPING NODES ON TO RAYS. When the tree is walked, the quantities stored on the tree nodes are mapped on to the `ray-arrays` in a two-step process. First, each mapped quantity, e.g. the node mass, is divided amongst different rays in proportion to the volume of the intersection between the node and the ray (see Fig. 2). This is implemented as described in Paper I, using a pre-calculated table of intersection volumes between a ray and a node with given angular coordinates (θ, ϕ) and angular size ψ. Second, the part of the node belonging to the ray is divided between evaluation points, r_i, using a kernel function $W(x)$ with $x \equiv (r_i - d)/h_n$ where d is the distance of the node mass centre from the target point, and h_n is the node linear size. This mapping also uses a precalculated table in which the weights of the evaluation points are recorded as a function of r_i, d, and h_n.

KERNELS. Three kernels are presently available, as discussed in detail in Appendix A. The first is a Gaussian kernel, $W_g(x)$ (equation A2), truncated at $x = \sqrt{3}/2$. The second, $W_p(x)$ (equation A3), has the form of a piece-wise polynomial of third order and has been obtained by fitting the mean intersection between a uniform cubic node and a randomly oriented line. TREERAY uses $W_p(x)$ by default for mapping the node masses and volumes. The third, $W_f(x)$, is tailored to the requirements of radiative transfer, and is used with the ONTHESPOT module; the considerations informing the prescription for the kernel coefficients are described in detail in Appendix A. Due to its nature, $W_f(x)$ is only used for mapping quantities related to radiative transfer (e.g. the radiation energy); all other quantities (e.g. mass) are mapped using $W_p(x)$.

2.4 Multipole acceptance criteria (MACs)

MACS IN THE TREE SOLVER. During the tree walk for a given target cell, a node is accepted if it satisfies the *MAC*. The simplest and most commonly used MAC is the Barnes–Hut (BH) MAC (Barnes & Hut
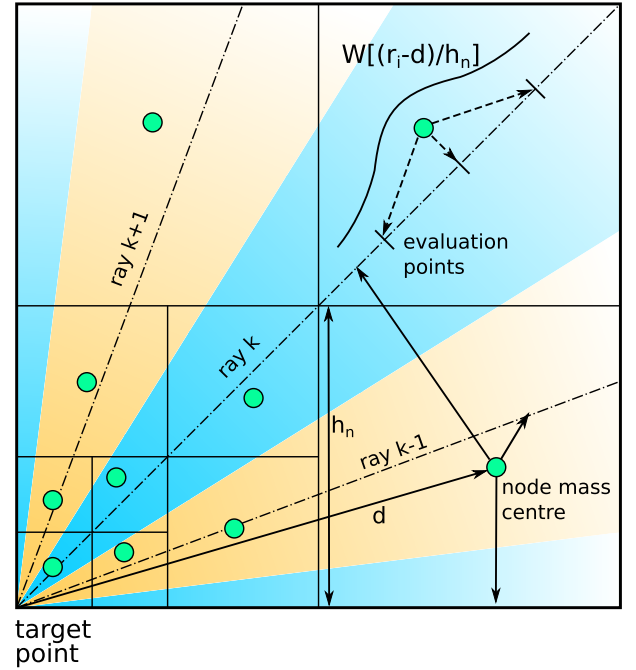

Figure 2. Mapping tree nodes on to rays during the tree walk. The tree nodes that need to be opened during the tree walk for a target point in the bottom left-hand corner are denoted by solid line squares. The system of rays which is cast from the target point is represented by the blue and yellow areas. The green circles are node mass centres. The quantities that are stored in each of the tree nodes are distributed to all rays which a node intersects (solid arrows), weighted by the relative intersection volume. For a given ray, the quantities are distributed to the so-called evaluation points along the ray using weights given by the kernel function $W[(r_i - d)/h_n]$ (dashed arrows).

1986). With the BH MAC, a node of size h_n, at distance d from the target cell, is accepted if

$$h_n/d < \theta_{\text{lim}}, \tag{7}$$

where θ_{lim} is a user-specified limiting opening angle. In Paper I, we describe several data-dependent MACs that make the GRAVITY module calculation more efficient. Below we define two new MACs, IF MAC (standing for Ionization Front) and Src MAC (standing for Sources), designed for the TREERAY/ONTHESPOT module, and also applicable to other TREERAY submodules. When these MACs are combined with the GRAVITY module MACs to ensure accurate gravity calculation, smaller nodes are opened in regions of dense, highly structured gas, and this leads to an increase in the computational cost of the tree walk. However, this does not impact the accuracy or performance of the TREERAY RTE solution, since the small nodes are smeared out on to the HEALPIX rays with the angular resolution given by N_{PIX}.

BH MAC. When the BH MAC is used together with TREERAY, a sensible choice is to set $\theta_{\text{lim}} \simeq (4\pi/N_{\text{PIX}})^{1/2}$. This ensures that the angular size of accepted tree nodes is approximately the same as the angle assigned to each HEALPIX ray (see analysis of this criterion for the OPTICALDEPTH module in Paper I, Section 3.3.1).

NEW MACS. Both the IF MAC and the Src MAC are controlled by limiting opening angles, respectively θ_{IF} and θ_{Src}. If the IF MAC is adopted, the code records for each node, n, both the total mass, m_n, and the mass of ionized gas, m_{ion} (defined as a threshold on the ionized hydrogen abundance or on the temperature). Then, during the tree walk, nodes for which $\delta_{\text{IF}} m_n < m_{\text{ion}} < (1 - \delta_{\text{IF}}) m_n$ (where

$\delta_{\rm IF} = 10^{-8}$) are assumed to include an ionization front. Such nodes are accepted only if

$$h_n/d < \theta_{\rm IF}. \qquad (8)$$

Similarly, if the Src MAC is adopted, nodes that include sources of radiation, i.e. $\varepsilon_{\rm n} > 0$, are accepted only if

$$h_n/d < \theta_{\rm Src}. \qquad (9)$$

2.5 Radiation transport equation

GENERAL RTE. After the tree walk is completed, the `ray-arrays` contain all the quantities needed to solve the Radiation RTE using the reverse ray-tracing method (Altay & Theuns 2013). For frequency band ν, the RTE takes the form[4]

$$\frac{{\rm d}I_\nu}{{\rm d}r} = -\varepsilon_\nu + \alpha_\nu I_\nu, \qquad (10)$$

where ε_ν and α_ν are the emission and absorption coefficients. The RTE must be solved along each of the $N_{\rm PIX}$ rays cast from a target point. On each ray, at each evaluation point i (at distance r_i from the target point along the ray) radiation transport is regulated by (i) the local emission coefficient, $\varepsilon_{\nu, i}$; (ii) the local absorption coefficient, $\alpha_{\nu, i}$; and (iii) the radiation energy density from the previous iteration step, $e_{\nu, i}$. These quantities have all been mapped on to the evaluation points during the tree walk. $e_{\nu, i}$ is only required if the emission or absorption coefficients depend on it. If the emission or absorption coefficients *do not* depend on $e_{\nu, i}$, it is straightforward to integrate equation (10) along each of the $N_{\rm PIX}$ rays, and thereby obtain the (direction-dependent) radiation intensity $I_{\nu, 0, k}$ ($k = 1$ to $N_{\rm PIX}$) at the target point. These $I_{\nu, 0, k}$ can then be summed to obtain a new estimate of the radiation energy density at the target point, and this can in turn be used to calculate radiative ionization and/or heating rates. However, if the emission or absorption coefficients *do* depend on $e_{\nu, i}$, more elaborate formulations of the problem may be appropriate, and this is the case for the TREERAY/ONTHESPOT module.

RTE IN ONTHESPOT. In the TREERAY/ONTHESPOT module treating the EUV radiation, we use different quantities from those in equation (10) to characterise the radiation field. $\varepsilon_{\rm EUV,i}$ gives the rate of emission of EUV photons from sources (i.e. hot stars) associated with evaluation point i. $\alpha_{\rm EUV,i}$ gives the rate of recombination of hydrogen into excited levels, per unit volume, at evaluation point i; in the ONTHESPOT approximation, such recombinations are exactly balanced by photoionizations,[5] so $\alpha_{\rm EUV,i}$ is also the rate at which EUV photos are destroyed in unit volume at evaluation point i. $\Phi_{\rm EUV,i,j}$ gives the rate at which EUV photons emitted into unit solid angle by the sources associated with evaluation point j reach the spherical surface through evaluation point i (see Fig. 3).

INTEGRATION PROCEDURE. The integration procedure first calculates

$$\Phi_{{\rm EUV},i=j,j} = \frac{\varepsilon_{{\rm EUV},j}}{4\pi}, \qquad \Phi_{{\rm EUV},i \neq j,j} = 0, \qquad (11)$$

at all the evaluation points that have sources, and sets it to zero everywhere else. Then, for all evaluation points, j, that have sources,

starting from $N_r - 1$, it cycles over the evaluation points from $i = j - 1$ to $i = 0$ and calculates

$$\Phi_{{\rm EUV},i,j} = \Phi_{{\rm EUV},i+1,j} - \mathcal{R}_{i+1,j}, \qquad (12)$$

truncating it to zero if it becomes negative. $\mathcal{R}_{i+1,j}$ is the rate at which photons from evaluation point j are lost due to recombinations in the ray segment between r_{i+1} and r_i, and is given by

$$\mathcal{R}_{i,j} = \frac{1}{2}(\alpha_{{\rm EUV},i} + \alpha_{{\rm EUV},i+1})\,{\rm d}V_{i,j} \times \zeta_{{\rm d},i+1} \times \zeta_{{\rm s},i+1,j} \times \zeta_{{\rm g},i,j}, \qquad (13)$$

where

$$\mathrm{d}V_{i,j} = [(r_j - r_i)^3 - (r_j - r_{i+1})^3]/3 \qquad (14)$$

is the volume of the cone of radiation from source j in the segment between r_i and r_{i+1} (see Fig. 3). The absorption rate is further corrected by three factors $\zeta_{{\rm s},i+1,j}$, $\zeta_{{\rm d},i+1}$, and $\zeta_{{\rm g},i,j}$, which are defined below.

CALCULATION OF FLUXES. The contribution from the sources associated with evaluation point j to the flux of EUV photons through evaluation point i is then given by

$$F_{{\rm EUV},i,j} = \Phi_{{\rm EUV},i,j}/(r_j - r_i)^2, \qquad (15)$$

and the total flux of EUV photons at r_i from all sources j on the ray at $r_j > r_i$ is

$$F_{{\rm EUV},i} = \sum_{j=i+1}^{N_R} F_{{\rm EUV},i,j}. \qquad (16)$$

CORRECTION FACTORS IN EQUATION (13). The first correction factor in equation (13), $\zeta_{{\rm s},i+1,j}$, accounts for recombinations that absorb photons coming along the same computed ray but from sources other than j. It is simply the ratio of the contribution to the flux from source j, $F_{i+1,j}$, to the total flux along the ray, F_{i+1},

$$\zeta_{{\rm s},i+1,j} = \frac{F_{{\rm EUV},i+1,j}}{F_{{\rm EUV},i+1}}. \qquad (17)$$

The second correction factor in equation (13), $\zeta_{{\rm d},i+1}$, follows a similar logic. It accounts for recombinations that absorb photons passing through the ray segment in other directions than along the computed ray, and it is defined as the ratio between $F_{{\rm EUV},i+1,j}$ and the trace of the radiation pressure tensor $\boldsymbol{P}_{{\rm EUV},i+1}$ (expressed in photons per unit area and time). The latter is obtained from the radiation energy density as ${\rm tr}(\boldsymbol{P}_{{\rm EUV},i+1}) = e_{{\rm EUV},i+1}c/h\nu_{\rm EUV}$, where $h\nu_{\rm EUV}$ is the mean energy of an EUV photon and c is the speed of light. The correction factor $\zeta_{{\rm s},i+1,j}$ is further truncated to lie between 0 and 2 to deliver a moderate convergence rate during the tree solver iterations, yielding

$$\zeta_{{\rm d},i+1} = \min\left(2, \frac{F_{{\rm EUV},i+1}h\nu_{\rm EUV}}{e_{{\rm EUV},i+1}c}\right). \qquad (18)$$

The third correction factor in equation (13), $\zeta_{{\rm g},i,j}$, corrects for the geometry of the ray segments. The segment length $r_{i+1} - r_i$, controlled by parameter η_R, can be substantially smaller than the tangential extent of the ray $\sim r_i \times N_{\rm side}$. In this situation, the flux from sources close to r_{i+1} would be too high, leading to an overrepresentation of these sources in equation (13). Therefore, we add a correction of the form

$$\zeta_{{\rm g},i,j} = \min\left(1, \frac{N_{\rm side}(r_j - r_i)}{2r_j}\right). \qquad (19)$$

Note that the minimum in equation (19) ensures that the correction is only applied to segments with $r_j - r_i < 2r_j/N_{\rm side}$.

---

[4]The signs on the right-hand side are reversed, as compared with the standard form, because the radiation propagates in the negative direction with respect to coordinate r originating at the target point.

[5]Although the on-the-spot approximation does not strictly require ionization/recombination equilibrium for recombinations into excited states (Case B), it is only under extreme circumstances that the on-the-spot approximation is valid and ionization/recombination equilibrium is not.
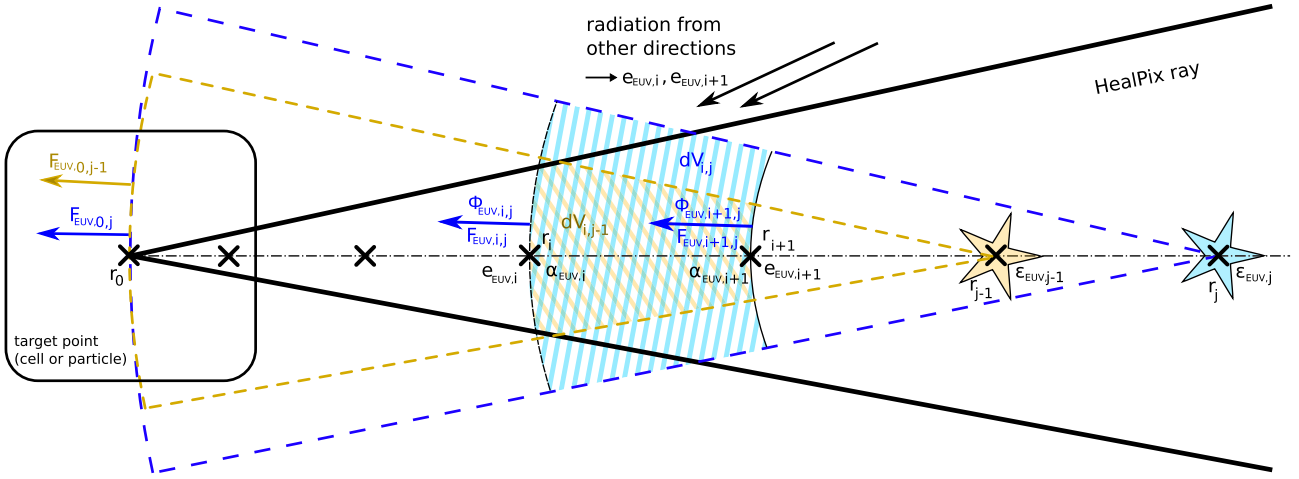
Figure 3. Schematic view of TREERAY, depicting the solution of the RTE along a single ray (bounded by thick black lines) towards a given target cell (on the left-hand side) after all the required quantities have been mapped on to the evaluation points (crosses). Each evaluation point is characterized by its distance from the target point, r_i, its absorption coefficient, α_i, the emission coefficient of all sources mapped on to it, ε_i, and the local radiation energy density from the previous time-step, e_i. See Section 2.3 and Fig. 2 for details of the mapping. For the furthest two evaluation points, $j-1$ and j (marked by stars, with emission coefficients, ε_{j-1} and ε_j), the yellow (blue) shaded area represents the volume $dV_{i,j-1}$ ($dV_{i,j}$) of the cone of radiation from source $j-1$ (j) in the segment between evaluation points i and $i+1$ (equation 14). The procedure described in Section 2.5 calculates the photon rates $\Phi_{i,j}$ and fluxes $F_{i,j}$ at each evaluation point. Fluxes $F_{0,j}$ at the target point are summed to give the radiation energy density there.

SUM OVER DIRECTIONS. Finally, the fluxes $F_{\mathrm{EUV},0,k}$ incident on the target point (i.e. $i = 0$) from the directions of all the HEALPIX rays ($k = 1$ to N_{PIX}) are summed to give a new estimate of the radiation energy density $e_{\mathrm{EUV},0}^{\mathrm{new}}$. It is assumed that the target point is a small sphere with radius r_{tp}.[6] Using the approximation of single direction radiation, the radiation energy density is

$$e_{\mathrm{EUV},0}^{\mathrm{new}} = \sum_{k=1}^{N_{\mathrm{PIX}}} \frac{F_{\mathrm{EUV},0,k} h\nu_{\mathrm{EUV}}}{c}. \tag{20}$$

2.6 Iterations and error control

NEED FOR ITERATIONS. If the absorption or emission coefficients depend on the ambient radiation field – as in the TREERAY/ONTHESPOT sub-module, where photons coming from different directions compete to be absorbed balancing recombinations – the code must iterate to find an acceptable estimate of the EUV radiation energy density, e_{EUV}. Modules and sub-modules that do not need iterations are executed only once, during the first iteration step, to save computing time. The first iteration starts with the radiation field e_{EUV} from the previous hydrodynamic time-step. This significantly reduces the number of iterations needed, because changes in the distribution of gas and sources between time-steps are typically small. In most cases, fewer than ten iterations are needed.

ERROR CONTROL. Iteration is terminated once the fractional change in the radiation field between successive iterations, $\delta_{e_{\mathrm{EUV}}}$, falls below a user-defined limit ϵ_{lim} (default value is 10^{-2}). Currently, two ways of determining the fractional change in the radiation field are implemented. The first considers the total radiation energy,

$$\delta_{e_{\mathrm{EUV}},\mathrm{tot}} = \frac{2\left(\sum e_{\mathrm{EUV},c}^{\mathrm{new}} dV_c - \sum e_{\mathrm{EUV},c} dV_c\right)}{\sum e_{\mathrm{EUV},c}^{\mathrm{new}} dV_c + \sum e_{\mathrm{EUV},c} dV_c}, \tag{21}$$

[6] r_{tp} is half of the grid cell size (if the target point is a grid cell) or the sink particle accretion radius (if the target point is a sink particle).

where subscript c denotes the quantity in a grid cell with volume dV_c, and the sums are taken over all grid cells in the computational domain. This criterion is similar to the one adopted by Dale, Ercolano & Bonnell (2012), who checked for the fractional change in the total mass of ionized gas; it works well when there are a few sources with comparable luminosity. The second criterion considers the change in the radiation field on a cell-by-cell basis,

$$\delta_{e_{\mathrm{EUV}},\mathrm{cell}} = \max_c \left(\frac{e_{\mathrm{EUV},c}^{\mathrm{new}} - e_{\mathrm{EUV},c}}{e_{\mathrm{EUV},\mathrm{norm},c}}\right), \tag{22}$$

where the maximum is taken over all grid cells c in the computational domain, and the normalizing energy is

$$e_{\mathrm{EUV},\mathrm{norm},c} = \max(e_{\mathrm{EUV},c}, e_{\mathrm{EUV},\mathrm{med}}). \tag{23}$$

Here, $e_{\mathrm{EUV},\mathrm{med}}$ is the median of all the $e_{\mathrm{EUV},c}$ values over the whole computational domain. The median is used to avoid zero or nearly zero normalizations, because e_{EUV} can have arbitrarily small values. The second criterion, using $\delta_{e_{\mathrm{EUV}},\mathrm{cell}}$, is safer and is set as the default.

INTERACTION WITH GAS. The calculated radiation energy density is used to modify the properties of the target point. If the target point is a grid cell, the code can for instance update the temperature or the chemical composition (including the ionization degree). If $e_{\mathrm{EUV},0}$ is non-zero, the TREERAY/ONTHESPOT module implements two possible treatments: (i) the temperature and ionization degree are given user-defined values (e.g. 10^4 K and 1.0, respectively), or (ii) $e_{\mathrm{EUV},0}$ is passed to the CHEMISTRY module (see Haid et al. 2018, for details, and Section 4 for a test case including chemistry).

2.7 Sources of radiation

SOURCE PROPERTIES. For some TREERAY sub-modules, the emission coefficient $\varepsilon_{\nu,c}$ in cell c is derived directly from the quantities that describe the gas in the cell. Other sub-modules, including TREERAY/ONTHESPOT, use radiation sources that are independent of the grid (e.g. sink particles representing stars or star clusters). Such sources are characterized by their position, luminosity (e.g. in the case of the TREERAY/ONTHESPOT module, number of EUV

photons emitted per second), and radius. These properties can either be read from a file at the start of a simulation and stay constant for the whole run, or they can be obtained from sink particles. In the latter case, the sinks move and their luminosities (and in principle also their radii) can vary with time. The FEEDBACKSINKS module can be used to calculate the luminosities of the stars and stellar clusters that sink particles represent, as functions of their age, mass and mass accretion history (see Gatto et al. 2017; Peters et al. 2017, for details).

MAPPING ON TO THE GRID. At each call of the tree solver, before the tree is built, all sources are mapped on to the grid, assuming that the sources are uniform spheres. In this way, the emission coefficient in each grid cell, which geometrically intersects with the volume of the source is set to a fraction of the source luminosity proportional to the intersection volume. In order to keep the mapping process fast, it is performed in two stages: creating a list of sources intersecting with each block, and then calculating the intersection of each cell with each source on the list.

2.8 Boundary conditions

BOUNDARY CONDITION TYPES. Three types of boundary condition are available for the tree solver; they are independent of the boundary conditions for the hydrodynamic solver. The first two types (*isolated* and *periodic*) have already been described in Paper I, the third one (*corner of symmetry*) is newly implemented and described in more detail below. *Isolated* boundary conditions assume that all source quantities (e.g. for gravity or emission of radiation) are zero outside the computational domain. *Periodic* boundary conditions invoke periodic copies of each tree node during the tree walk, and use only the closest copy to the target point. The GRAVITY module uses the Ewald method to take into account all the periodic copies, but calculations with the TREERAY sub-modules are presently limited to the nearest periodic copy.

CORNER OF SYMMETRY. *Corner of Symmetry* boundary condition allow the user to simulate one-eighth of the problem under investigation, if the problem has the appropriate symmetry. We refer to the truly simulated part of the computational domain as the *original*, and we specify one corner of it as the *corner*. During the mapping of the tree on to the rays we reconstruct the entire problem by spawning seven ghosts of the *original*. The first three ghosts are copies of the *original* mirrored in the faces adjacent to the *corner*. The remaining four ghosts are point reflections of the first four ghosts through the *corner*. One can show that if the MACs discussed in Section 2.4 hold for the *original*, they also hold for all seven ghosts. One useful application of this boundary condition is to problems with spherical symmetry such an the Spitzer test (see Section 3.1, model (p)), giving us as an effective resolution of level n while actually only computing at level $n - 1$.

2.9 Tree solver time-step

ABU IN TREERAY. In Paper I we introduced the *adaptive block update* (ABU) method, which improves the code performance by only updating blocks where the quantities calculated by the tree solver change significantly; we also illustrated the benefits when ABU is used with the GRAVITY or TREERAY/OPTICALDEPTH modules. However, ABU does not deliver such significant benefits with the TREERAY/ONTHESPOT module, or any other module where the absorption or emission coefficients depend on the local radiation energy density. This is because calculating the radiation energy density couples together large regions of space, thereby requiring the code to update many blocks.

TREE SOLVER TIME-STEP. However, in many applications, the time-scale for the physical processes which are calculated by the tree solver is much longer than the hydrodynamic time-step. For example, in calculations with a multiphase ISM, the dense cold/warm gas is the main contributor to the gravitational potential as it contains most of the mass in the system and absorbs most of the radiation, and this gas typically moves with velocities smaller than $10 \, \mathrm{km \, s^{-1}}$ (e.g. Girichidis et al. 2016). On the other hand, the hydrodynamic time-step calculated from the Courant condition can be very small, because it is defined by the hot gas (with sound speeds $\gtrsim 300 \, \mathrm{km \, s^{-1}}$) or stellar winds (with velocities $\gtrsim 1000 \, \mathrm{km \, s^{-1}}$), but the contribution of this hot or fast gas to the calculation of gravity and radiation is typically negligible. Since the tree solver is used to calculate the gas self-gravity and the EUV or longer wavelength radiation field, it may not need to be called at every hydrodynamic time-step; the decision whether this approximation is reasonable depends on the nature of the phenomena being simulated, and is left to the user. The frequency of calls to the tree solver is regulated by setting the time-step, Δt_{ts}, which can be longer than the hydrodynamic time-step. A convenient way to set Δt_{ts} is to define a parameter

$$v_{\mathrm{tsdt}} = \Delta x / \Delta t_{\mathrm{ts}}, \qquad (24)$$

where Δx is the size of the smallest grid cell. v_{tsdt} then sets the maximum velocity with which the gas and/or sources relevant to the tree solver can move. By default, we set $v_{\mathrm{tsdt}} = \infty$ and hence $\Delta t_{\mathrm{ts}} = 0$, in which case the tree solver is called at each hydrodynamic time-step. Examples of how setting v_{tsdt} to a finite value impacts the code performance and accuracy are given in Section 4.

2.10 Load balancing

WORKLOAD PROBLEM. The computational cost of the tree solver is almost always dominated by the tree walk, and the most expensive operation is mapping the tree nodes on to the system of rays. If no radiation passes through a node, this operation is omitted, and as a result, the tree walk can be significantly cheaper in regions where no radiation is present. However, this may lead to a non-optimal distribution of the work load among different processors if each processor computes the same number of blocks (which is the default in FLASH).

LOAD BALANCING SCHEME. Therefore, we implement a simple load balancing scheme, based on measuring the wall clock time needed to execute a tree walk for all grid cells in a block, $t_{\mathrm{wl, blk}}$. After each tree solver call, we collate the $t_{\mathrm{wl, blk}}$ measurements and calculate their median value, $t_{\mathrm{wl, med}}$. Then we increase the workload weights, w_{blk}, of blocks with $t_{\mathrm{wl, blk}} > t_{\mathrm{wl, med}}$ by the factor $t_{\mathrm{wl, blk}}/t_{\mathrm{wl, med}}$. These workload weights are then used by FLASH in the next redistribution of blocks amongst the processors; each processor is assigned a number of blocks such that the sum of their weights is approximately the same for all processors. Note that this scheme improves the code performance only if the computational time is dominated by the tree solver. Therefore, we use this scheme, in combination with the tree solver time-step, only for time-steps when the tree solver is called. All other time-steps are calculated with a default flat block distribution, i.e. the same number of blocks on each processor.

3 ACCURACY AND PERFORMANCE TESTS

In this section, we describe four tests of the TREERAY algorithm illustrating its strengths and weaknesses, using idealized configurations of gas and sources. A more complex test, involving physical processes commonly included in astrophysical simulations of star

formation and its feedback, is presented in Section 4. All tests in Sections 3 and 4 have been run on the IT4I/Salomon supercomputer cluster[7] consisting of 1008 compute nodes, each equipped with two 12-core Intel Xeon E5-2680v3 @ 2.5 GHz processors and 2GB of RAM per core. The nodes are interconnected with the InfiniBand FDR56 network using the 7D Enhanced hypercube topology.

3.1 Spitzer test

SPITZER BUBBLE. The Spitzer bubble is one of the simplest models of the interaction of ionizing radiation with gas. In this model, the UV radiation from a young, massive star ionizes and heats the surrounding gas, creating an H II region, i.e. an overpressured, expanding bubble of photoionized gas bounded by a sharp ionization front (e.g. Spitzer 1978; Whitworth 1979; Deharveng et al. 2010). For a typical O-star, the radiative energy input can be very large, $\gtrsim 10^4 \, L_\odot$. However, only a small fraction of this radiative energy is converted into kinetic energy ($\lesssim 0.1$ per cent; Walch et al. 2012). The expanding ionization front drives a shock into the surrounding neutral gas, sweeping it up into a dense cold shell, and this may trigger the formation of a second generation of stars (e.g. Elmegreen & Lada 1977; Walch et al. 2013). H II regions are able to disrupt low-mass molecular couds long before the massive stars explode as supernovae (Whitworth 1979; Dale et al. 2012; Walch et al. 2012; Haid et al. 2019)

SPITZER BUBBLE EXPANSION. The Spitzer test describes the spherically symmetric expansion of an H II region into a uniform ambient medium with density ρ_o, which supposedly resembles a cold molecular cloud. It is used as a standard test for radiative transfer schemes (e.g. Mellema et al. 2006; Krumholz et al. 2007; Iliev et al. 2009; Mackey & Lim 2010a, 2011; Raga, Cantó & Rodríguez 2012; Rosdahl et al. 2013; Bisbas et al. 2015; Raskutti, Ostriker & Skinner 2017). The analytic solution for the time evolution of the D-type ionization front is given by (Spitzer 1978)

$$R_{\text{IF,anl}}(t) = R_S \left(1 + \frac{7}{4} \frac{c_i t}{R_S} \right)^{4/7}, \qquad (25)$$

where

$$R_S = \left(\frac{3 \dot{N}_{\text{LyC}} m_p^2}{4\pi \alpha_B X^2 \rho_o^2} \right)^{1/3} \qquad (26)$$

is the Strømgren radius (Strömgren 1939); c_i is the sound speed in the ionized gas inside the bubble; $\dot{N}_{\text{LyC}}$ is the rate at which the central star emits hydrogen-ionizing photons (i.e. photons with energy $E_\nu \geq 13.6 \, \text{eV}$); m_p is the proton mass, X is the mass fraction of hydrogen; and $\alpha_B = 2.7 \times 10^{-13} \, \text{cm}^3 \, \text{s}^{-1}$ is the Case-B recombination coefficient for an isothermal H II region with temperature $T_i = 10^4 \, \text{K}$. We assume that the gas is composed of hydrogen, with mass fraction $X = 0.70$, and helium, with mass fraction $Y = 0.30$; ionization of helium is ignored.

SPITZER TEST SETUP. In Bisbas et al. (2015) and Haid et al. (2018), we have already demonstrated that TREERAY/ONTHESPOT, as implemented in the FLASH code, delivers high accuracy in the Spitzer test. Here, we explore how the code behaviour depends on the various control parameters. We set up a cubic computational domain of size $30 \times 30 \times 30 \, \text{pc}^3$, which is filled with cold, dense molecular gas, having uniform temperature, $T_n = 10 \, \text{K}$, and uniform density, $\rho_0 = 7.63 \times 10^{-22} \, \text{g cm}^{-3}$; the hydrogen is assumed to be

molecular, and the helium atomic, so the mean molecular weight is 2.35, and the total number-density of gas particles $n_0 \approx 195 \, \text{cm}^{-3}$. At $t = 0$, a radiation source at the centre starts emitting ionizing photons at rate $\dot{N}_{\text{LyC}} = 10^{49} \, \text{s}^{-1}$. In the ionized gas the temperature is set to $T_i = 10^4 \, \text{K}$ and the mean molecular weight to $\mu_i = 0.678$, corresponding to ionized hydrogen and atomic helium. All the gas has adiabatic index, $\gamma = 5/3$. If, during the evolution, the temperature in the neutral gas exceeds 300 K, it is instantaneously cooled to 300 K; consequently the shell of shock-compressed neutral gas immediately ahead of the expanding IF is effectively isothermal at $T_s = 300 \, \text{K}$ (strictly speaking, the adiabatic index should be slightly below $\gamma = 5/3$ at this temperature, because the rotational degrees of freedom of H_2 are starting to be excited, but we ignore this detail). This temperature limit is chosen in order to resolve the thickness of the shell and prevent numerical instabilities in the hydrodynamic (PPM – piecewise parabolic method) solver that would otherwise occur. The sound speeds in the molecular cloud, the shell and the H II region, are $c_n = 0.187 \, \text{km s}^{-1}$, $c_s = 1.30 \, \text{km s}^{-1}$, and $c_i = 14.3 \, \text{km s}^{-1}$, respectively. These parameters result in a Strømgren radius of 1.434 pc.

VARIED PARAMETERS. We present results for 16 models, denoted (a)–(p) (see Table 1) with different combinations of the following six parameters: the radiation-field error control limit ($\delta_{e_{\text{EUV}},\text{cell}}$ or $\delta_{e_{\text{EUV}},\text{tot}}$, see Section 2.6), the grid resolution (given by the refinement level l_r), the angular resolution (given by the limiting opening angle θ_{lim}, the number of HEALPIX rays N_{PIX}, and the limiting opening angles θ_{IF} and θ_{Src} for the IF MAC and Src MAC, respectively), the radiation-energy error limit (ϵ_{lim}), the radial resolution of rays (η_R), and the source radius (R_{src}). In model (l), the single radiation source is replaced by 100 sources with luminosities $\dot{N}_{\text{LyC}} = 10^{47} \, \text{s}^{-1}$ distributed in a sphere of radius eight grid cells, instead of a single source of that radius. This is done to demonstrate that TREERAY can faithfully handle a large number of radiation sources. The last model (p), is calculated with the *corner of symmetry* boundary condition, i.e. only one octant of the total domain is computed (see Section 2.8). For each model, we evaluate the error in the ionization front position, $e_{\text{IF}} \equiv (R_{\text{IF, num}} - R_{\text{IF, anl}})/R_{\text{IF, anl}}$, at the end of the run (1.5 Myr), the processor time for a single iteration step, t_{iter}, and the processor times in the tree solver, t_{tr}, and in the hydrodynamic solver, t_{hydro}, for the whole run.

FIDUCIAL RUN. Fig. 4 shows the evolution of model (a), the fiducial run, displaying the radial profiles of the gas density, pressure and temperature, averaged over all directions, at three times throughout the evolution. By the end of the evolution ($t = 1.5 \, \text{Myr}$), the pressure in the shell approaches the value within the H II region indicating that the shell thickness is approximately resolved. Fig. 5 shows the position of the ionization front, $R_{\text{IF, num}}$, as a function of time, and compares it with the prediction of equation (25). The fractional error in the ionization front position, $e_{\text{IF}} \equiv |R_{\text{IF, num}} - R_{\text{IF, anl}}|/R_{\text{IF, anl}}$, at $t = 1.5 \, \text{Myr}$, is shown in Table 1. For the fiducial run, this error is ~ 2.5 per cent. Fig. 6 shows the radial profile of the radiation energy density, e_{EUV}, and compares it with the analytically obtained radiation energy density, $e_{\text{EUV,anl}}$, denoted by the black lines. The latter is determined by combining equations (25) and (26), and replacing ρ_o with the density in the H II region, ρ_i, which is assumed to be uniform between $r = 0$ and R_{IF}:

$$e_{\text{EUV,anl}}(r, t) = \frac{E_\nu \dot{N}_{\text{LyC}}}{4\pi r^2 c} \times \left[1 - \frac{r^3}{R_S^3} \left(1 + \frac{7}{4} \frac{c_i t}{R_S} \right)^{-12/7} \right]. \qquad (27)$$

ERROR CONTROL. Figs 7 and 5 show that most of the other models (all but models (e), (i), and (m)) also agree very well with the analytic solution: the error in the ionization front position is better than 5 per cent (corresponding to ~ 1.7 grid cells at the standard

Table 1. Accuracy and performance of the Spitzer test.

Model	l_r	$\theta_{\lim}$	$\theta_{IF}, \theta_{Src}$	N_{PIX}	$\delta_{e_{EUV}}$	$\epsilon_{\lim}$	η_R	R_{src}	e_{IF}	t_{iter}	t_{tr}	t_{hydro}
(a) fiducial	5	0.5	∞	48	cell	10^{-2}	2	1	0.025	0.24	158	3.4
(b) $\delta_{e_{EUV}}$ = tot	5	0.5	∞	48	tot	10^{-2}	2	1	0.024	0.25	78	3.4
(c) $l_r = 4$	4	0.5	∞	48	cell	10^{-2}	2	1	0.011	0.03	9.2	0.3
(d) $l_r = 6$	6	0.5	∞	48	cell	10^{-2}	2	1	0.040	2.5	3100	44
(e) $N_{PIX} = 12$	5	1.0	∞	12	cell	10^{-2}	2	1	0.173	0.05	65	5.6
(f) $N_{PIX} = 192$	5	0.25	∞	192	cell	10^{-2}	2	1	0.046	1.5	940	3.4
(g) $\epsilon_{\lim} = 10^{-1}$	5	0.5	∞	48	cell	10^{-1}	2	1	0.024	0.24	125	3.4
(h) $\epsilon_{\lim} = 10^{-3}$	5	0.5	∞	48	cell	10^{-3}	2	1	0.025	0.24	250	3.4
(i) $\eta_R = 1$	5	0.5	∞	48	cell	10^{-2}	1	1	0.203	0.21	180	3.5
(j) $\eta_R = 8$	5	0.5	∞	48	cell	10^{-2}	8	1	0.033	0.39	260	3.4
(k) $R_{src} = 4$ gc	5	0.5	∞	48	cell	10^{-2}	2	4	0.026	0.24	150	3.4
(l) 100 sources	5	0.5	∞	48	cell	10^{-2}	2	8*	0.023	0.24	240	4.5
(m) $\theta_{IF} = 0.5$, $N_{PIX} = 12$	5	1.0	0.5	12	cell	10^{-2}	2	1	0.034	0.11	100	4.1
(n) $\theta_{IF} = 0.5$, $N_{PIX} = 48$	5	1.0	0.5	48	cell	10^{-2}	2	1	0.029	0.12	90	3.7
(o) $\theta_{IF} = 0.25$, $N_{PIX} = 192$	5	1.0	0.25	192	cell	10^{-2}	2	1	0.048	0.34	240	3.7
(p) COS	4**	0.5	∞	48	cell	10^{-2}	2	1	0.065	0.12	80	0.8

Notes. Column 1 gives the model name. The following columns list:

(i) l_r: the refinement level defining the grid resolution($4 \to 64^3$, $5 \to 128^3$, $6 \to 256^3$)

(ii) $\theta_{\lim}$: limiting opening angle for BH MAC

(iii) $\theta_{IF}, \theta_{Src}$: limiting opening angles for IF MAC and Src MAC, respectively (see Section 2.4)

(iv) N_{PIX}: number of rays (defining the angular resolution)

(v) $\delta_{e_{EUV}}$: error control method (either $\delta_{e_{EUV},cell}$ given by equation (22), or $\delta_{e_{EUV},tot}$ given by equation (21); see Section 2.6)

(vi) $\epsilon_{\lim}$: maximum allowed relative error

(vii) η_R: resolution in the radial direction; η_R is inversely proportional to the distance between evaluation points on rays (see equation 4)

(viii) R_{src}: size of the source (in grid cells)

(ix) e_{IF}: relative error in the ionization front position at $t = 1.5$ Myr

(x) t_{iter}: processor time for a single iteration step (in core hours)

(xi) t_{tr}; processor time spent by the tree solver in the whole run (in core hours)

(xii) t_{hydro}; processor time spent by the hydrodynamic solver in the whole run (in core hours).

*in model (k), 100 sources of radius 1 grid cell were distributed randomly in a sphere with radius 8 grid cells.

**model (p) uses *corner of symmetry*, i.e. only one octet is calculated, the grid cell size is the same as in model (a)
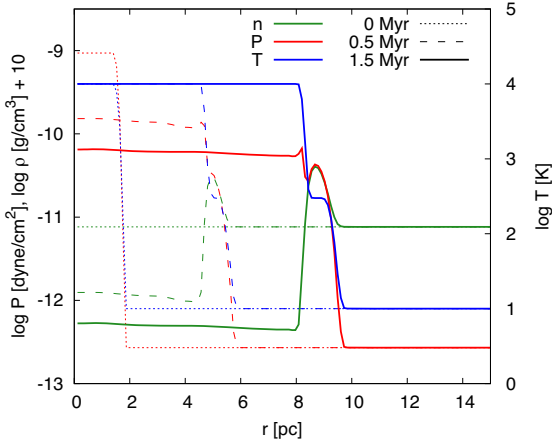

Figure 4. Spitzer test: the evolution of the fiducial run (model a). The green, red and blue lines show the mean radial variation of, respectively, the gas particle density, the pressure and the temperature. Different line types distinguish different stages in the evolution: initial conditions (dotted), 0.5 Myr (dashed) and 1.5 Myr (solid).

resolution), and the discrepancy in the radiation energy density is negligible. Models (b), (g), and (h) evolve almost identically to model (a), indicating that the choice of error control method has almost no impact, and that the calculation is accurate even with the fractional error limit $\epsilon_{\lim} = 0.1$. All three models use the same processor time per iteration, t_{iter}, as the fiducial run, which is to be expected since the error control condition does not affect the calculation within an iteration. The total time spent in the tree solver, t_{tr}, is approximately two times lower for model (b), as $\delta_{e_{EUV},tot}$ requires fewer iterations. Conversely, model (h) has t_{tr} approximately ~60 per cent higher than model (a), because the lower error limit leads to a larger number of iterations. In the case of model (g), with increased $\epsilon_{\lim} = 0.1$, t_{tr} drops by ~20 per cent relative to model (a), reaching approximately two iterations per hydrodynamic time-step.

GRID RESOLUTION. A comparison of model (a) with models (c) and (d) shows that a lower (higher) resolution leads to a smoother and more blurred (denser and better resolved) shell. However, even the low-resolution model (c), where $R_{IF, num}$ is only resolved with 17 grid cells at $t = 1.5$ Myr, and with ~3 grid cells at $t = 0$, results in an H II region with the correct radius and shape. The processor time per iteration, t_{iter}, scales with the number of grid cells to the power ~1.1, between models (c), (a), and (d) (i.e. slightly superlinearly). Similarly, the total time in the tree solver, t_{tr}, scales with the number of grid cells to the power ~1.4 between the same models (again, slightly above the theoretical (4/3)-power, where the extra (1/3) derives from the shorter time-steps required by the Courant condition at higher resolution).

ANGULAR RESOLUTION. Models (e) and (f) test the dependence on the angular resolution by varying N_{PIX} and $\theta_{\lim}$, setting them so that the typical tree node angular size is similar to the angular size of the rays, as suggested in Paper I. Model (f) evolves in a similar way to model (a), but t_{iter} and t_{tr} are ~6 times larger. This shows that the fiducial run is well converged with regard
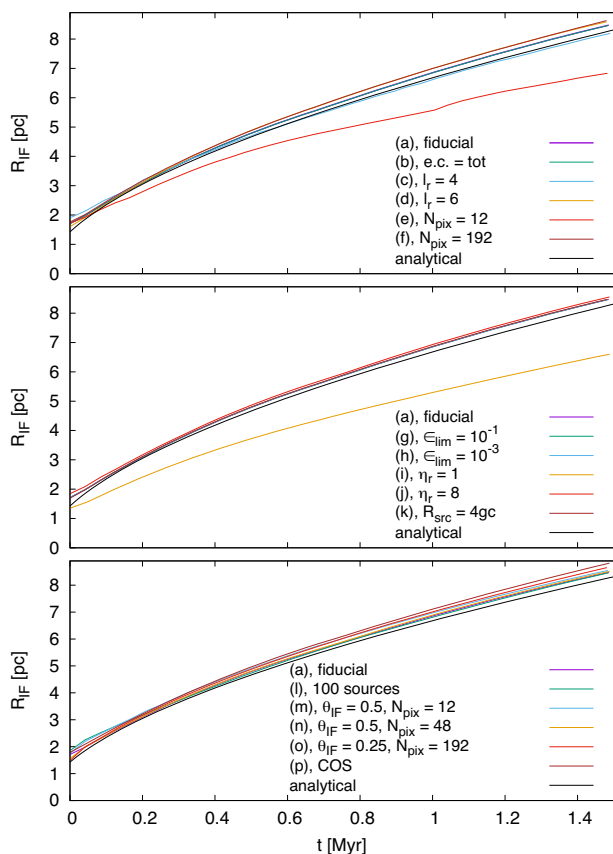
Figure 5. Spitzer test: the evolution of the ionization front radius. The top panel shows models (a)–(f); the middle panel shows models (a) and (g)–(k); and the bottom panel shows models (a) and (l)–(p). The black line shown on all three panels is the analytic solution given by equation (25).

Figure 6. Spitzer test: the mean radial profile of the radiation energy density at 1.5 Myr. The top panel shows models (a)–(f); the middle panel shows models (a) and (g)–(k); and the bottom panel shows models (a) and (l)–(p). The black line on all three panels is the radiation energy density in the analytical solution (equation 27).

to angular resolution, and that the scaling is in agreement with Paper I where we found that $t_{\rm tr}$ scales with $\theta_{\rm lim}$ somewhere between $\sim \theta_{\rm lim}^{-2}$ and $\sim \theta_{\rm lim}^{-3}$. On the other hand, model (e), with very low-angular resolution, $N_{\rm PIX} = 12$, exhibits significant departures from the analytical solution. The volume of the HII region reaches only ~ 53 per cent of the correct value, and the radiation energy is distributed non-spherically, with lower values along the Cartesian diagonals. Consequently, the shell expands more slowly along the diagonals than along the axes. Along the axes, the faster expansion is also accelerated by numerical instabilities in the inadequately resolved shell. The main reason for the depressed radiation-energy density along the diagonals is that the very large $\theta_{\rm lim}$ allows the acceptance of very large tree nodes by the BH MAC. Such large nodes include the source in one corner and a part of the shell in the opposite one, leading to a poor estimate of the rate of absorption of photons within the node, and hence poor estimates of the radiation fluxes and energies.

NEW MACS. The new criteria for accepting nodes in the tree walk, IF MAC and Src MAC, also control the angular resolution, setting it higher in regions where increased resolution is needed. The new MACs are tested in models (m), (n) and (o). Model (m) with $\theta_{\rm IF} = \theta_{\rm Src} = 0.5$ and $N_{\rm PIX} = 12$ demonstrates that it is not worth increasing the angular resolution in the tree walk without also increasing the number of rays. Even though model (m) does not suffer from the problem of accepting too large tree nodes (unlike model (e)), the radiation field shows significant departures from spherical symmetry, and its computational cost is similar to the much more accurate model (n).
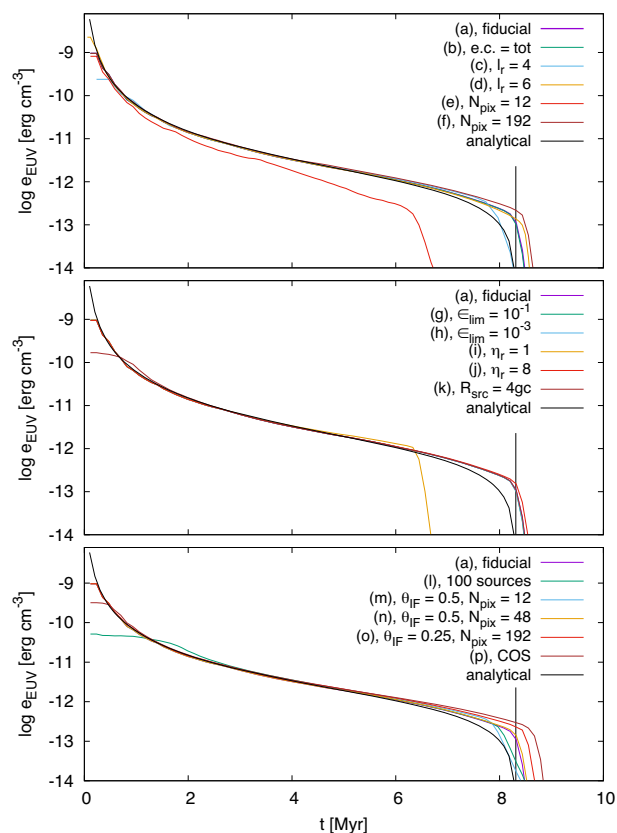
Model (n) with $\theta_{\rm IF} = \theta_{\rm Src} = 0.5$ and $N_{\rm PIX} = 48$ evolves in a similar way to the fiducial run (a), and its computational cost is approximately 40 per cent lower. Model (o), with even higher angular resolution, $\theta_{\rm IF} = \theta_{\rm Src} = 0.25$ and $N_{\rm PIX} = 192$, shows improvement in the spherical symmetry of the radiation field, like model (f), but it is calculated in approximately one quarter of the time. This demonstrates the benefits of the new MACs, particularly at higher angular resolution.

RADIAL RESOLUTION. Models (i) and (j) explore the code behaviour at different radial resolutions, controlled by the parameter η_R. Model (j), with four times more evaluation points on the rays, gives almost identical results to model (a), indicating that the fiducial run with $\eta_R = 2$ is well converged. The much higher density of evaluation points results in an ~ 40 per cent increase in the computational time. On the other hand, model (i), with $\eta_R = 1$, takes almost the same amount of time as model (a), and the error in the position of the ionization front increases to 7 per cent.

SOURCE SIZE. Models (k) and (l) show that the HII region still evolves correctly if the source of ionizing radiation is distributed throughout a larger volume. Model (l) exhibits moderate departures from spherical symmetry; these are attributable to significant departures from spherical symmetry in the ionizing flux from 100 discrete sources distributed *randomly* in a sphere of radius 1.9 pc. The time per iteration, $t_{\rm iter}$, is essentially the same as in model (a), demonstrating a unique property of our algorithm, namely that the computational cost
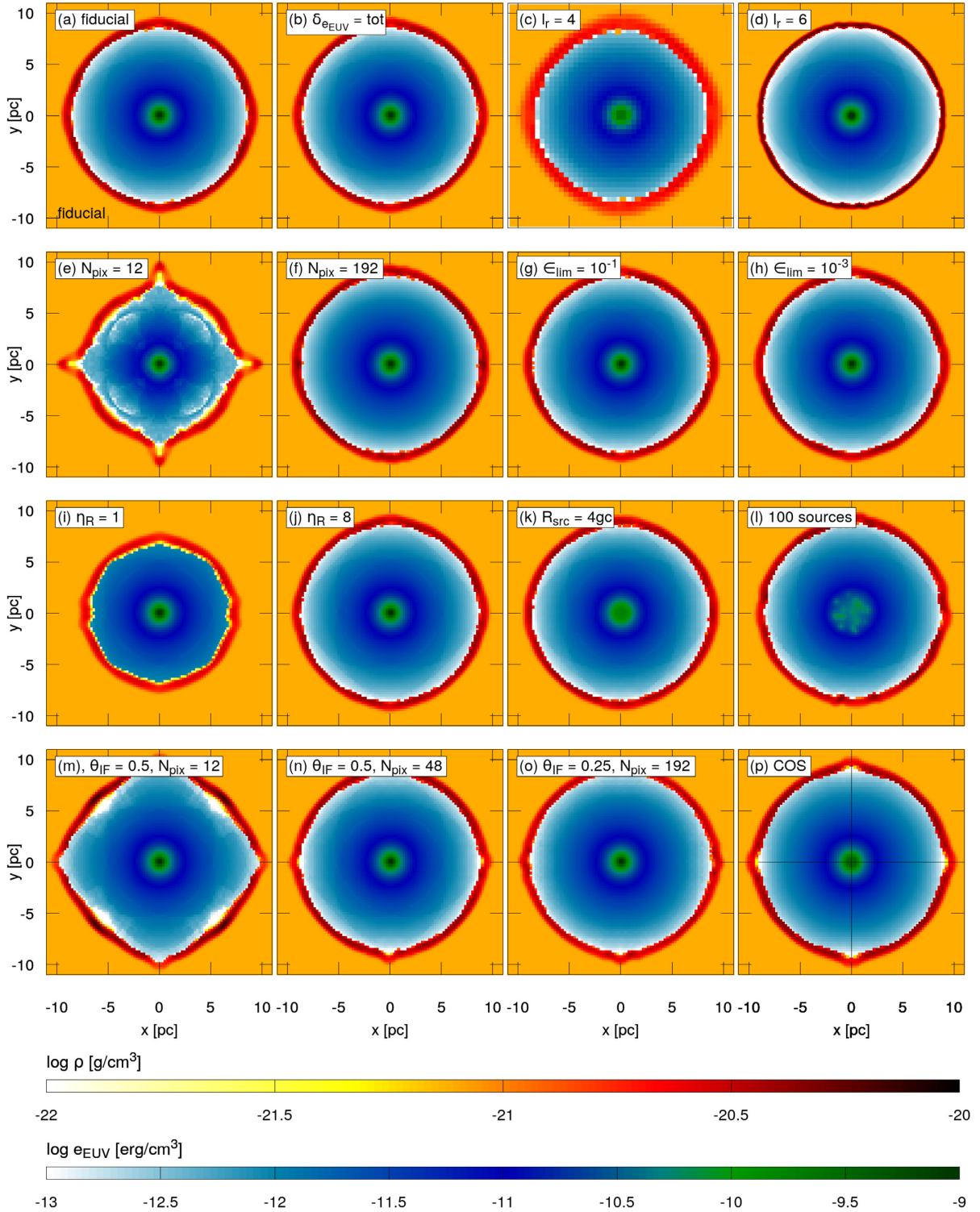
Figure 7. Spitzer test: the distribution of the radiation energy (blue–green) and the gas density (yellow–red), in the plane $z = 0$, at time 1.5 Myr, for models (a)–(p), as noted in the top left-hand corner of each panel; see Table 1 for the model parameters. The logarithm of the radiation energy is shown in the region with non-zero ionization degree, the logarithm of the gas density is shown in the remaining parts (i.e. for the neutral gas only).

does not depend on the number of radiation sources (see Section 5.2 for a more detailed discussion).

CORNER-OF-SYMMETRY. Finally, model (p) tests the special 'Corner of Symmetry' boundary conditions, which allow the user to simulate one-eighth of a spherically symmetric problem (see Section 2.8). Model (p) runs at approximately twice the speed of model (a), and produces broadly similar results. However, the shell exhibits numerical artefacts along the axes resulting from the directionally split hydrodynamic solver. As a result, the error on the radius of the ionization front is greater, $e_{IF} \sim 6.5$ per cent.
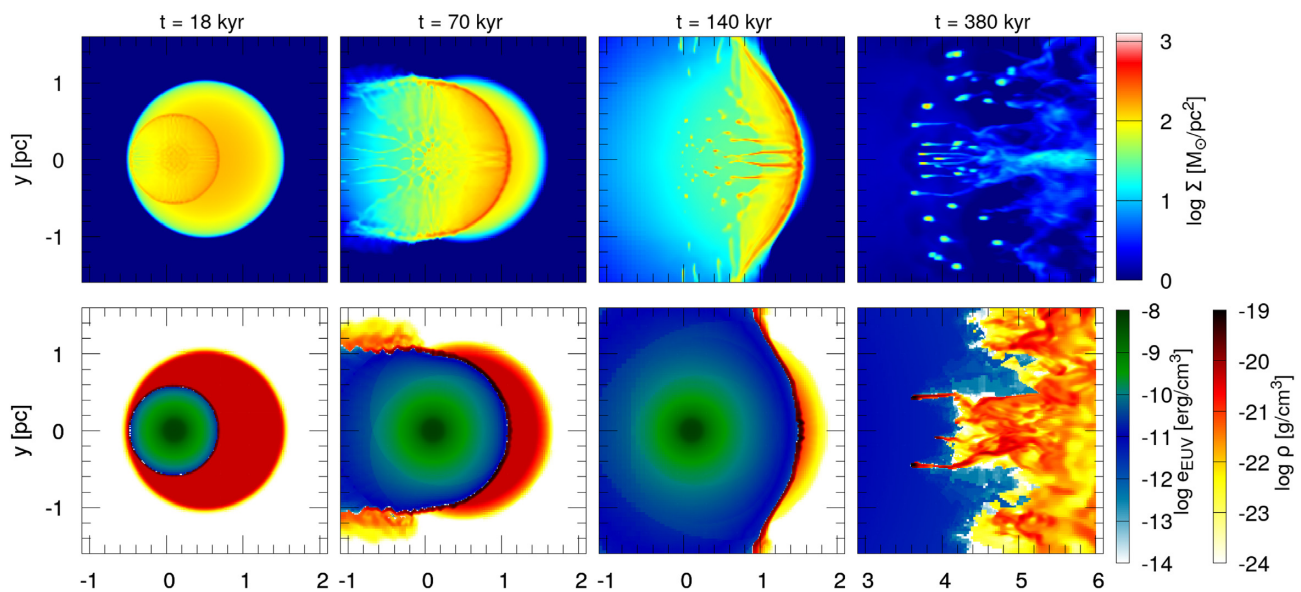
Figure 8. Blister-type H II region test, at times $t = 18$, 70, 140, and 380 kyr. The top panels show the logarithm of the gas column density. The bottom panels show the logarithm of the radiation energy in the region with non-zero ionization degree, and the logarithm of the gas density in the remaining parts (i.e. for the neutral gas only). Note that the computational domain has dimensions $8 \times 4 \times 4 \, \text{pc}^3$, and is much larger than the region shown in these maps.

3.2 Blister-type H II region

BLISTER H II REGION. In order to test the algorithm in a situation that is not spherically symmetric, we model a spherical cloud with an ionizing star located inside it, but not at its centre. This model was first discussed by Whitworth (1979) and Tenorio-Tagle (1979) who suggested that, as soon as the ionization front reaches the edge of the cloud on one side, the HII region bursts out of the cloud and the remainder of the cloud on the other side is accelerated by the rocket effect (Kahn 1954). This scenario was later studied numerically by Bisbas et al. (2009) (hereafter B09), Gendelev & Krumholz (2012) and others. Here, we set up the cloud and the radiation source with the same physical parameters as in B09, and compare our results with theirs.

INITIAL CONDITIONS. The radiation source is at the centre of the coordinate system and emits ionizing photons at rate $\dot{N}_{\text{LyC}} = 10^{49} \, \text{s}^{-1}$. The spherical cloud has mass $M_0 = 300 \, \text{M}_\odot$, radius $R_0 = 1 \, \text{pc}$ and uniform density $\rho_0 = 4.85 \times 10^{-21} \, \text{g cm}^{-3}$; its centre is at $(0.4, 0, 0) \, \text{pc}$. The cloud is embedded in a rarefied ambient gas with density $\rho_{\text{amb}} = 10^{-24} \, \text{g cm}^{-3}$, and the computational domain has dimensions $8 \times 4 \times 4 \, \text{pc}^3$. The neutral gas has temperature $T_n = 100 \, \text{K}$, but otherwise both neutral and ionized (i.e. irradiated) phases have the same properties (molecular weights, adiabatic index, hydrogen mass fraction) as in Section 3.1. All the gas that is not irradiated is immediately returned to the temperature T_n, at each time-step. Self-gravity is switched off. The simulation is run for 0.5 Myr, corresponding to 1574 time-steps, and the computational cost is $\sim 10\,000$ core hours.

NUMERICAL PARAMETERS. The model is calculated on an AMR grid with minimum and maximum refinement levels of 4 and 6, respectively (i.e. the highest resolution corresponds to $512 \times 256 \times 256$). A simple density-based criterion is used to refine/derefine blocks wherever the maximum density exceeds $10^{-21} \, \text{g cm}^{-3}$, or drops below $5 \times 10^{-22} \, \text{g cm}^{-3}$. The hydrodynamic boundary conditions are set to outflow. The parameters controlling the TREERAY accuracy are set as follows: the tree solver uses both the IF MAC and the Src MAC with $\theta_{\text{IF}} = 0.25$, $\theta_{\text{src}} = 0.25$, and $\theta_{\text{lim}} = 1.0$; the

number of rays is $N_{\text{PIX}} = 192$; the radial resolution is $\eta_R = 2$; and the maximum allowed relative error is $\epsilon_{\text{lim}} = 0.01$.

BLISTER EVOLUTION. The evolution of the blister-type H II region is shown in Fig. 8. The ionized region expands spherically until $t = 18 \, \text{kyr}$, when it reaches the edge of the cloud on the left-hand side. After that time, the ionized gas flows out of the cloud on this side, while opening a growing cavity within it. On the right-hand side, the H II region continues to expand into the remainder of the cloud, opening a growing cavity. The originally spherical shell of swept-up gas is at first converted into an hemispherical shell (at $t \sim 70 \, \text{kyr}$), and later into an almost flat layer (after $t \sim 140 \, \text{kyr}$). When all the cloud material has been swept up in a given direction from the source, accretion on to the shell stops and the shell starts to accelerate and become Rayleigh–Taylor unstable. As a result, the layer breaks into a large number of cloudlets. These cloudlets were called 'cometary knots' in B09, due to their almost spherical core and a tail created by ablation by the ionized gas streaming around them and away from the source.

COMPARISON WITH B09. A qualitative comparison with the SPH simulation of the same model in B09 (see their Figs 13 and 14) shows an almost perfect agreement. The location and the shape of the ionization front and other large-scale features are indistinguishable. The formation of cometary knots is also reproduced remarkably well, given that the angular resolution of the TREERAY simulation ($\theta_{\text{IF}} = 0.25$) is more than an order of magnitude coarser than in B09, where the angular separation between neighbouring rays is set by the local SPH-particle smoothing lengths, typically $\theta \lesssim (0.1 \, \text{pc})/(4 \, \text{pc}) = 0.025$. This demonstrates the ability of the reverse ray-tracing method to deliver high spatial resolution of the radiation field at the ionization front, even with relatively large angles between neighbouring rays. However, there are some differences in the small-scale structures. First, the TREERAY simulation generates larger perturbations of the shell along the grid axes, and the higher noise there seeds the Rayleigh–Taylor instability through the odd-even decoupling mechanism identified by Quirk (1994). Second, by $t = 380 \, \text{kyr}$, the number the cometary knots is substantially lower than in B09, because many of them have evaporated, due to their

lower density, which in turn is caused by the lower spatial resolution of the grid code.

3.2.1 Rabbit hole test

PENETRATION DEPTH PROBLEM. If the ionization front has a complex structure, as in the previous test, it is questionable whether the radiation field computed by a code with limited angular resolution can properly follow the ionization front geometry. A particularly difficult configuration for TREERAY is a radiation source shining into a deep, narrow hole. Such holes form frequently in astrophysical applications, for instance when a swept-up shell becomes unstable and breaks apart (e.g. as in Walch et al. 2013). In order to quantify the accuracy of the code in this situation, we implement a test called the *rabbit hole*, in which we measure the penetration of the radiation field as a function of the width of the hole.

RABBIT HOLE SETUP. To mimic a hole within a dense shell, we set up an elongated box containing two media: the walls of the hole are formed of cold dense gas with sound speed $c_c = 0.25 \, \mathrm{km \, s^{-1}}$ and $\rho_c = 10^{-18} \, \mathrm{g \, cm^{-3}}$; the inside of the hole is filled with warm rarefied gas with $c_w = 10 \, \mathrm{km \, s^{-1}}$ and $\rho_w = 10^{-24} \, \mathrm{g \, cm^{-3}}$. The two media are not in pressure equilibrium, but, since we only calculate the first time-step, this is irrelevant. However, the extreme density contrast renders this test particularly difficult, because nodes that are far away from a given target point are large. Hence, the denser the cold medium is, the more mass will be accumulated in these large and distant nodes, and this can lead to an artificial blocking of the radiation.

TEST PARAMETERS. The computational domain is $6 \times 2 \times 2 \, \mathrm{pc}^3$. The hole has a square cross-section with side l_w and stretches from 0 to 6 pc. The ionizing source is placed at the entrance to the hole, $(x, y, z) = (0, 0, 0)$, and emits ionizing photons at rate $N_{\mathrm{LyC}} = 10^{49} \, \mathrm{s^{-1}}$. The test is performed with the hole pointing in the *x*-direction and then in the *z*-direction (as illustrated in Fig. 9, top panels), and with uniform resolution ($384 \times 128 \times 128 \simeq 6.3 \times 10^6$ cubic cells with side length $\sim 0.16 \, \mathrm{pc}$).

PENETRATION DEPTH MEASUREMENT. The Strömgren radius for a star with $N_{\mathrm{LyC}} = 10^{49} \, \mathrm{s^{-1}}$, in a uniform medium with $\rho_w = 10^{-24} \, \mathrm{g \, cm^{-3}}$, is 72 pc (see equation 26), i.e. the radiation should shine right through the rabbit hole. However, for narrow holes (small l_w), TREERAY is unable to obtain the correct solution without reducing the angular resolution to intolerably low values. This is shown in Fig. 9 (bottom panel), where we plot the maximum depth, l_d, up to which we measure a non-zero radiation energy density as a function of l_w, for two different settings of the control parameters.

EXPECTED PERFORMANCE. Theoretically, we expect l_d to fall in the range $l_w/(2\theta_{\mathrm{lim}}) \lesssim l_d \lesssim l_w/\theta_{\mathrm{lim}}$. The lower limit corresponds to the situation where the HEALPIX cone widens symmetrically from the target cell towards the source, and touches the walls on both sides of the hole at distance $l_w/(2\theta_{\mathrm{lim}})$. The upper limit corresponds to the situation where one border of the cone is parallel to the wall of the hole and the opposite border touches the wall at $l_w/\theta_{\mathrm{lim}}$.

ACTUAL PERFORMANCE. To test this behaviour, we use two settings. In the first setting, we adopt a constant opening angle of $\theta_{\mathrm{lim}} = 0.25$ corresponding to $N_{\mathrm{PIX}} = 192$. For this Oct-Tree resolution, we expect $l_d = l_w/\theta_{\mathrm{lim}} = 4l_w$, which is slightly better than the performance actually achieved. Moreover, simply using more rays without implementing the source and ionization-front MACs does not improve this result. In the second setting, we adopt a larger $\theta_{\mathrm{lim}} = 1.0$ but also implement the physical MACs, with $\theta_{\mathrm{IF}} = \theta_{\mathrm{src}} = 0.125$, corresponding to HEALPIX level 8, i.e. $N_{\mathrm{PIX}} = 768$ rays. Despite the large value of θ_{lim}, the results with the additional physical MACs outperform the more expensive simulations with $\theta_{\mathrm{lim}} = 0.25$ (see the
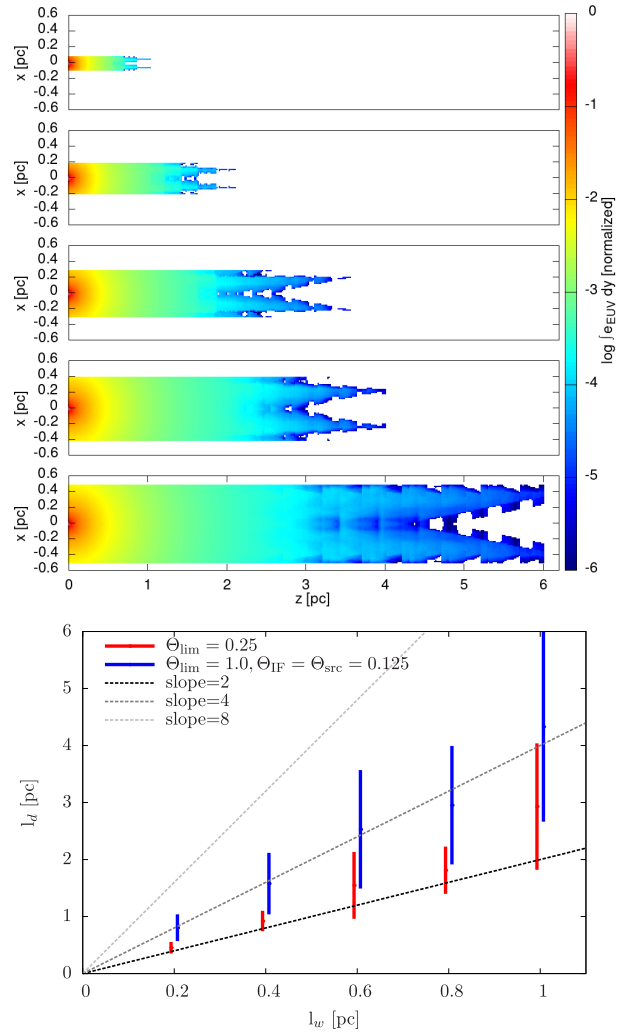
Figure 9. Top five panels: The radiation energy density integrated along the line of sight (*y*-direction) with $\theta_{\mathrm{lim}} = 1.0$, $\theta_{\mathrm{IF}} = \theta_{\mathrm{src}} = 0.125$ and – from the top to bottom panel – $l_w = 0.2, 0.4, 0.6, 0.8$, and 1.0 pc. Radiation only reaches the end of the computational domain ($z = 6$ pc) for $l_w = 1$ pc. Bottom panel: The range of depths up to which the radiation propagates as a function of the width of the rabbit hole. The 'error-bars' indicate the minimum and maximum depth, as found by different simulations with the hole pointing in the *x*- or *z*-directions. The dotted lines show $l_d = ml_w$ with slopes $m = 2, 4$, and 8; $m = 4$ corresponds to the theoretical expectation, $l_d = l_w/\theta_{\mathrm{lim}}$ with $\theta_{\mathrm{lim}} = 0.25$, and $m = 8$ corresponds to the theoretical expectation, $l_d = l_w/\theta_{\mathrm{IF}}$ with $\theta_{\mathrm{IF}} = \theta_{\mathrm{src}} = 0.125$. The results with physical MACs are significantly better than those obtained with $\theta_{\mathrm{lim}} = 0.25$, but a slope of $m = 8$ cannot be achieved for the high density contrast simulated here.

bottom panel of Fig. 9), and in the run with $l_w = 1$ pc the radiation field reaches the end of the computational domain. However, the theoretical scaling of $l_d = 8l_w$ is not achieved, due to the high density contrast (a factor of 10^6) between the warm gas within the hole and the surrounding cold gas.

3.3 Radiation driven implosion (RDI)

In this test, we study a compact, dense, neutral cloud illuminated by ionizing radiation from a single direction. The astronomical motivation is the cometary globules, commonly observed in Galactic HII regions, with bright rims on the side irradiated by a nearby hot star star (or stars) and tails pointing in the opposite direction

(see e.g. Lefloch & Lazareff 1995; Deharveng et al. 2010; Getman et al. 2012; Schneider et al. 2012). It has been suggested that, in such a configuration, star formation can be triggered by the RDI mechanism (Bertoldi 1989) and this mechanism has been extensively studied analytically (e.g. Mellema et al. 1998; Miao et al. 2009; Mackey & Lim 2010b) and numerically using SPH codes (e.g. Kessel-Deynet & Burkert 2003b; Gritschneder et al. 2010; Bisbas et al. 2011; Dale et al. 2012, and references therein), grid-based hydrodynamic codes (e.g. Lefloch & Lazareff 1994; Mellema et al. 1998) and magnetohydrodynamic codes (Mackey & Lim 2010b).

RDI AND TREERAY. It is usually assumed that the size of the cloud is small, compared with the distance to the radiation source. Technically, this can be arranged, either by using a plane–parallel radiation field (as in the Lefloch & Lazareff setup), or by setting a large distance between the source and the cloud (as in Bisbas et al. 2009, hereafter B09). We choose the latter option here, even though TREERAY can easily be modified to treat a plane–parallel radiation field, and we postpone description of this feature to a future paper. This test is relatively hard for algorithms with the limited angular resolution, due to the small angular size of the cloud, as seen from the source. This is why it was chosen by B09, to demonstrate the ability of their algorithm to split rays adaptively, so that the ray separation is everywhere similar to the resolution of the hydrodynamic solver. TREERAY achieves a similar resolution at the irradiated border of the cloud, by using reverse ray-tracing, which ensures a small separation between neighbouring rays at the point of the flux calculation.

RDI SETUP. We use a similar setup to B09, which was chosen to resemble as closely as possible the setup defined by Lefloch & Lazareff (1994). A spherical cloud with mass $M = 20\,M_\odot$, radius $R = 0.5$ pc, and uniform density $\rho_0 = 2.6 \times 10^{-21}$ g cm^{-3}, is illuminated by a source at distance $D = 3.5$ pc, emitting ionizing photons at rate $N_{\mathrm{LyC}} = 3.2 \times 10^{48}$ s^{-1}. The neutral gas (i.e. the cloud and the gas in its shadow) has temperature $T_{\mathrm{n}} = 100$ K and is composed of pure atomic hydrogen, i.e. $\mu_n = 1$. The ionized gas has temperature $T_i = 10^4$ K and $\mu_i = 0.5$. The computational domain is $2 \times 2 \times 6$ pc^3. Initially, the source is located at $(x, y, z) = (1, 1, 0)$ pc, and the cloud centre at $(1, 1, 3.6)$ pc. The whole computational domain (apart from the cloud) is filled with a rarefied gas having density $\rho_{\mathrm{amb}} = 10^{-24}$ g cm^{-3}. We calculate six models denoted (a)–(f), for which we vary the angular resolution (parameters θ_{lim} and N_{PIX}), the grid resolution (refinement level l_r) and the MAC criterion (IF MAC on or off; see Table 2). Models (a)–(e) use a uniform grid, i.e. the minimum and maximum refinement levels are the same, while model (f) uses AMR with refinement levels 4 to 6, so the coarsest grid is $64^2 \times 192$ and the finest one is $256^2 \times 768$. The AMR criterion refines a block if the maximum density within it exceeds 10^{-21} g cm^{-3}, and de-refines it if the maximum density drops below 5×10^{-22} g cm^{-3}, thereby ensuring that only the cloud and its immediate surroundings are calculated on the highest resolution. A typical grid structure can be seen in the bottom right-hand panel of Fig. 10. We evaluate the morphology of the cloud and its shadow, and compute the mass of the neutral gas as a function of time, comparing these quantities with B09 and Lefloch & Lazareff (1994). The error in the neutral gas mass is $e_M = (M_{\mathrm{n}} - M_{\mathrm{n, B09}})/M_{\mathrm{n, B09}}$, where M_{n} is the neutral gas mass at $t = 0.5$ Myr and $M_{\mathrm{n, B09}}$ is the reference value from B09.

RDI EVOLUTION. We select model (f) as the fiducial model, since – along with model (e) – it gives the best accuracy at reasonable computational cost. Fig. 11 shows the logarithms of column density (top) and radiation energy density (bottom) for model (f) at a sequence of times. The column density can be compared directly with fig. 15 in B09, and we see good agreement between the two codes. Qualitatively, the cloud evolves in the same way as in previous

Table 2. Accuracy and performance of the RDI test.

Model	l_r	θ_{lim}	N_{PIX}	θ_{IF}	e_M	t_{iter}	t_{tr}
(a) $N_{\mathrm{PIX}} = 12$	5	1	12	∞	0.24	0.49	840
(b) $N_{\mathrm{PIX}} = 48$	5	0.5	48	∞	0.22	1.1	1900
(c) $N_{\mathrm{PIX}} = 192$	5	0.25	192	∞	0.12	6.1	10500
(d) $\theta_{\mathrm{IF}} = 0.25$	5	1	192	0.25	0.11	2.6	4800
(e) $l_r = 6$	6	1	192	0.25	0.05	26	45000
(f) Fiducial	4,6	1	192	0.25	0.05	1.7	6300

Notes. Column 1 gives the model name. The following columns list:

(i) l_r: the refinement level defining grid resolution '5' → $128^2 \times 384$; '6' → $256^2 \times 792$; '4,6' → AMR with minimum and maximum refinement levels 4 and 6, respectively.)
(ii) θ_{lim}: the limiting opening angle for the BH MAC
(iii) N_{PIX}: the number of rays (defining the angular resolution)
(iv) θ_{IF}: the limiting opening angle for the IF MAC
(v) e_M: the relative error in the neutral gas mass at $t = 0.5$ Myr
(vi) t_{iter}: the processor time for a single iteration step (in core-hours)
(vii) t_{tr}; the processor time in the tree solver for the whole simulation (in core-hours)

studies. Initially, the radiation ionizes the outer layers of the cloud in the direction of the source and a shock starts to propagate into the remaining neutral gas, compressing it from the sides. At ~ 130 kyr, a dense core is formed on the cloud axis near the ionization front and it re-expands due to its internal thermal pressure, while at the same time being ablated by radiation on the side facing the source. Eventually, a cometary tail is formed at ~ 200 kyr. The bottom panels show the radiation energy density and the shadow behind the cloud. Even though a certain amount of radiation diffuses artificially into the shadow region (due to smoothing the edges of the cloud into larger tree nodes), the overall shape of the shadow looks reasonably good. The mass of neutral gas (see Fig. 12) follows almost exactly the curve from B09 up to $t \simeq 0.25$ Myr, and then becomes slightly higher, leading to a discrepancy ~ 5 per cent at $t \simeq 0.5$ Myr. This discrepancy is due to insufficient spatial resolution of the dense core, which starts to be ablated by the ionizing radiation after 200 kyr. The rate at which gas is ionized is very sensitive to the density of the neutral gas close to the ionization front, and the SPH code used in B09 uses many more resolution elements to describe the core density profile; in our simulation, it is only a few grid cells in diameter.

RDI COMPARISON. Fig. 10 shows the logarithms of the column density (top panel) and the radiation energy density (bottom panel), for models (a)–(f), at $t = 180$ kyr. Models (a)–(c) explore the effect of changing the angular resolution. As expected, a relatively high angular resolution is needed to compute this configuration faithfully. In Model (a), with $\theta_{\mathrm{lim}} = 1$, $N_{\mathrm{PIX}} = 12$, the radiation energy is incorrect by tens of percent, which is mainly due to the very large sizes of tree nodes. In model (b) with $\theta_{\mathrm{lim}} = 0.5$, $N_{\mathrm{PIX}} = 48$, the radiation field is approximately correct in the ionized regions, but the shadow is too wide, and the irradiated side of the cloud is too flat. In both models (a) and (b), the mass of neutral gas is higher than in B09 by ~ 20 per cent. In model (c), with $\theta_{\mathrm{lim}} = 0.25$, $N_{\mathrm{PIX}} = 192$, the shape of the cloud and its shadow closely match the results of B09, and the error in the neutral gas mass is about 10 per cent. The computational cost increases by a factor between 5 and 6 for each reduction of θ_{lim} by a factor of 2. Model (d) behaves almost exactly like model (c), but computationally it is almost five times cheaper, demonstrating how effective the IF MAC can be. Model (e) refines the spatial resolution by a factor of 2 in each direction (as compared with models (a)–(d)), and this reduces the error to $e_{\mathrm{M}} \sim 5$ per cent but increases the computational cost by a factor of ~ 48 relative to
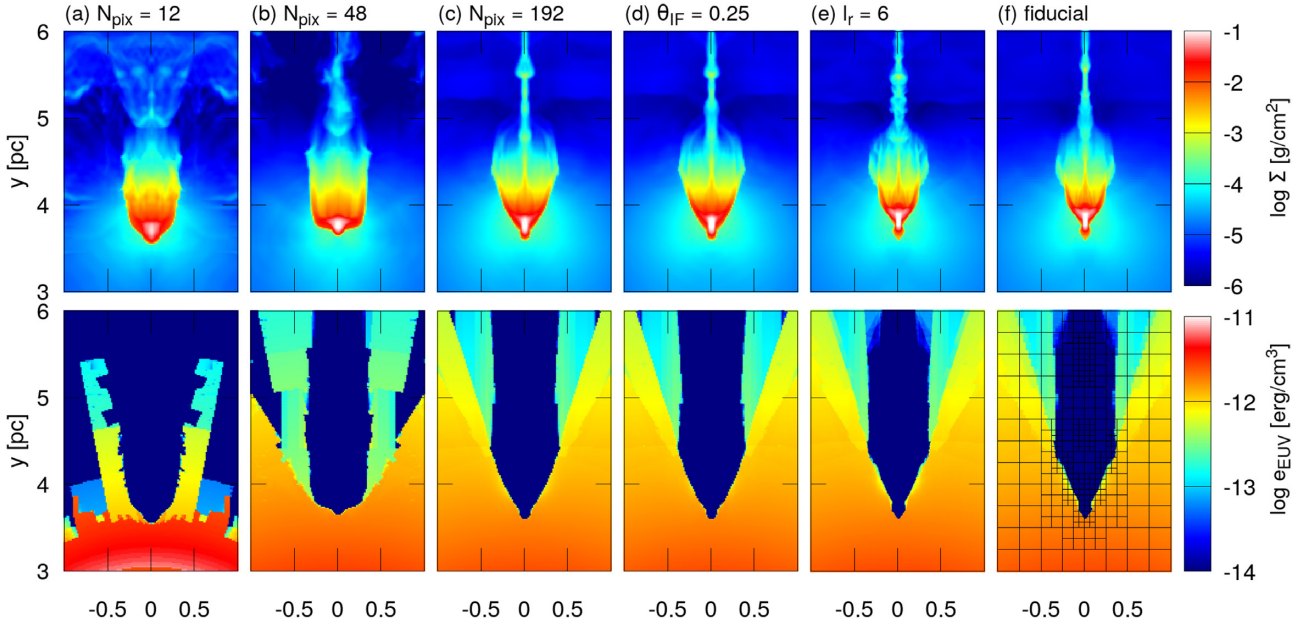
Figure 10. RDI test, comparison of models (a)–(f). The top panels show the logarithm of the gas column density. The bottom panels show the logarithm of the radiation energy density on the mid-plane. All models are plotted at time 180 kyr.
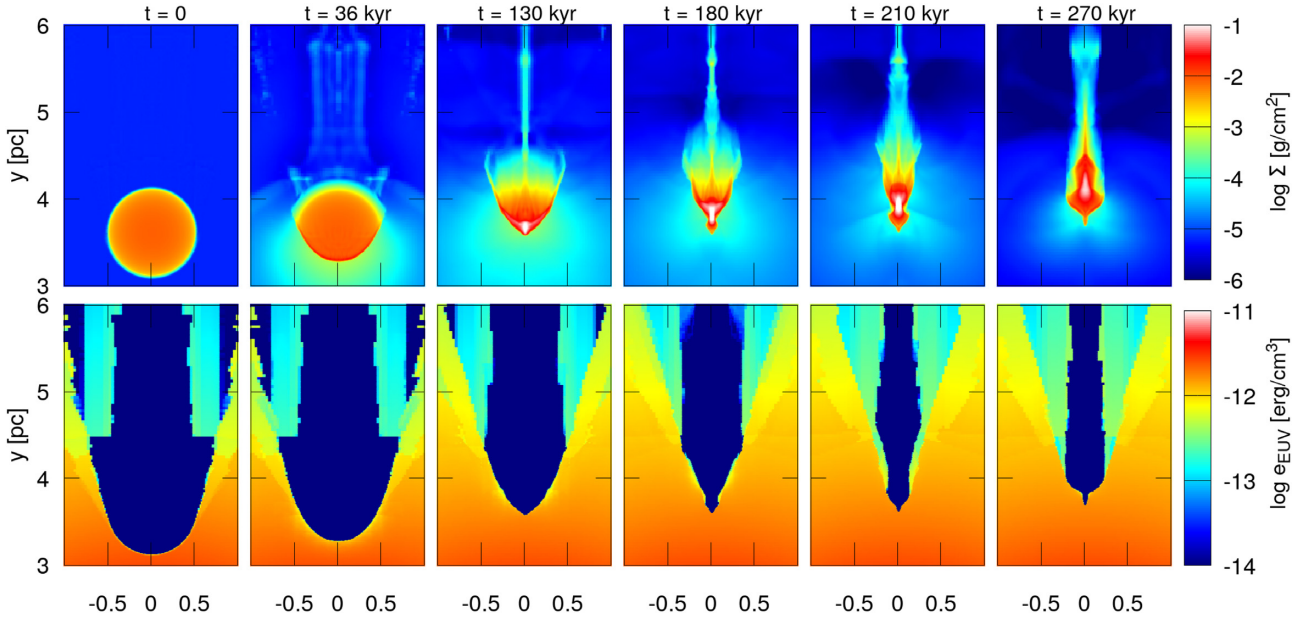

Figure 11. RDI, evolution of the fiducial model (f). The top panels show the logarithm of the gas column density at times 0, 36, 130, 180, 210, and 270 kyr. The bottom panels show the logarithm of the radiation energy density on the mid-plane at the same times.

model (d). Part of this (a factor of ∼16) is due to the higher number of grid cells and the shorter time-step. The remainder (a factor of ∼3) is partly due to the higher number of tree nodes that need to be opened, and partly due to the larger number of evaluation points on each ray.

3.4 Cloud irradiated by two sources

TWO-SOURCE MOTIVATION. This test assesses the fidelity of the code when treating a cloud that is irradiated by two identical sources, from different directions. Similarly to the previous tests (Sections 3.2.1 and 3.3), it is sensitive to the angular resolution, represented by $\theta_{\lim}$

and $N_{\rm PIX}$, and to the choice of MAC. Additionally, it evaluates the iteration and error control procedures (see Section 2.6), since their failure would corrupt the symmetry of the radiation field with respect to the plane perpendicular to the line connecting the two sources.

TWO-SOURCE SETUP. The computational domain is $-2 \le x \le 4$ pc, $-2 \le y \le 4$ pc, and $-3 \le z \le 3$ pc. The two sources are located at $(x, y, z) = (-2, 0, 0)$ pc and $(0, -2, 0)$ pc, and each emits ionizing photons at rate $N_{\rm LyC} = 3.2 \times 10^{48}$ s^{-1}. The cloud has mass $M = 20\,{\rm M_\odot}$, radius $R = 0.5$ pc, density 2.6×10^{-21} g cm^{-3}, and is located at the centre of coordinates. Outside the cloud, the computational domain is filled with rarefied gas with density 10^{-24} g cm^{-3}. The other parameters
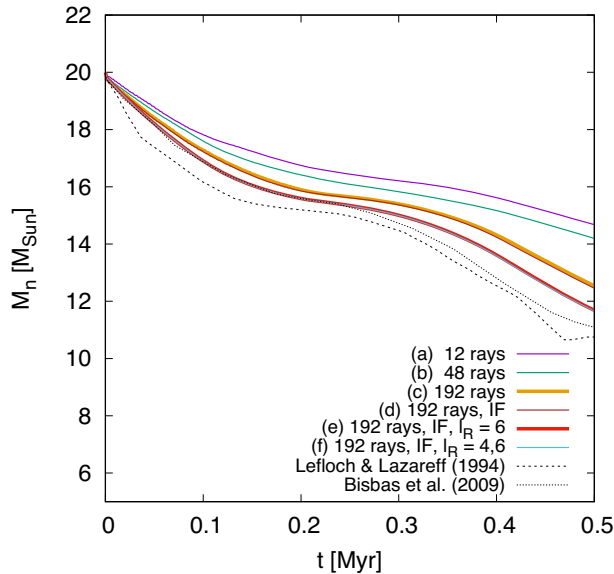
Figure 12. RDI: evolution of the neutral gas mass for models (a)–(f), compared with B09 and Lefloch & Lazareff (1994).

Table 3. Accuracy and performance of the two-source test.

model	l_r	θ_{lim}	N_{PIX}	$\theta_{IF}, \theta_{Src}$	t_{iter}
(a) $N_{PIX} = 12$	5	1	12	∞	0.095
(b) $N_{PIX} = 48$	5	0.5	48	∞	0.3
(c) $N_{PIX} = 192$	5	0.25	192	∞	1.7
(d) $\theta_{IF} = 0.25$	5	1	192	0.25	0.65
(e) $l_r = 6$	6	1	192	0.25	6.8
(f) AMR	4,6	1	192	0.25	0.2
(g) $\theta_{IF} = 0.125$	5	1	768	0.125	2.1

Notes. Column 1 gives the model name. The following columns list:

(i) l_r: the refinement level defining the grid resolution ('5' → 128^3; '6' → 256^3; '4,6' → AMR with minimum and maximum refinement levels 4 and 6, respectively.)
(ii) θ_{lim}: the limiting opening angle for the BH MAC
(iii) N_{PIX}: the number of rays (defining the angular resolution)
(iv) $\theta_{IF}, \theta_{Src}$: the limiting opening angles for the IF MAC and the Src MAC, respectively
(v) t_{iter}: the processor time for a single iteration step (in core-hours)

are given in Table 3 for all seven calculated models, (a)–(g). Models (a)–(f) have the same parameters as the corresponding models for the RDI test; model (g) has a very high angular resolution given by $\theta_{IF} = \theta_{Src} = 0.125$ and $N_{PIX} = 768$ (as used in the Rabbit hole test, Section 3.2.1). All the models were run for a single time-step to let the TREERAY iteration process converge; the time-evolution was not explored.

TWO-SOURCE RESULTS. We only evaluate this test qualitatively, by comparing the computed radiation field with the analytic solution. Fig. 13 shows the radiation energy density on the $z = 0$ plane for all models, and for the analytic solution, assuming a completely opaque cloud. We see that none of the models exhibit significant deviation from symmetry about the line $x = y$. Model (a), with the lowest angular resolution, exhibits the largest deviations from the analytic solution. The shadow behind the cloud has the wrong shape, some radiation leaks into the shadowed region, and the radiation energy drops to zero at distances $D \gtrsim 3$ pc from the cloud. The last effect is caused by the large tree opening angle $\theta_{lim} = 1$: at large distance from

a target cell, the two sources and parts of the cloud are merged into a single node with high absorption coefficient. The incorrect shadow geometry is the result of both a low number of rays and a high θ_{lim}. In model (b), the drop to zero disappears, but distortion of the shadow geometry is still significant. The shadow geometry is approximately correct in model (c) with $\theta_{lim} = 0.25$ and 192 rays. Nevertheless, even this relatively high angular resolution is not sufficient to reproduce the radiation energy in the top right-hand corner, which should be relatively high, because this region is irradiated by both sources. Model (d) shows that almost the same radiation field as in model (c) can be obtained at lower computational costs by invoking the physical MACs with $\theta_{IF} = \theta_{Src} = 0.25$. Models (e) and (f) show that the grid resolution and the AMR have a negligible effect on the radiation field. Finally, model (g) with very high angular resolution (shown in the top right-hand corner) better reproduces the radiation field (though not perfectly) showing that the method converges to the correct solution with increasing angular resolution.

4 STAR FORMATION AND FEEDBACK WITH TREERAY

STAR FORMATION AND FEEDBACK MOTIVATION. The purpose of the final test is to demonstrate the combination of TREERAY with other physical modules that are often used in simulations of star formation with feedback, and to evaluate the code performance under realistic conditions. We derive this test setup from model CNM 60 (i.e. 60 $M_\odot$ star located in the cold neutral medium) of Haid et al. (2018), who explore the relative impact of radiation and stellar winds in different environments. In the CNM 60 model, a source of radiation and stellar wind representing a 60-$M_\odot$ star is placed in a dense cold neutral medium, resulting in the formation of an expanding H II region with a stellar wind bubble at its centre. A challenging aspect of this test is the combination of relatively complex physics (related to radiation and cold gas chemistry) with the high velocity of stellar winds, $\gtrsim 1000$ km s^{-1} (obliging the hydrodynamic solver to take very short time-steps, due to the Courant–Friedrichs–Lewy criterion). In order to demonstrate the ability of TREERAY to deal efficiently with a large number of sources, we split the single source used by Haid et al. (2018) into 100 smaller sources, which together emit ionizing photons at the same net rate, and together deliver the same total wind power (with the same wind velocity, so the mass-loss rate from each individual star is simply divided by 100). The 100 sources are distributed randomly in a sphere of radius 8 pc, within a cloud of radius 10 pc, to represent a toy-model star cluster.

PHYSICAL PROCESSES. The physical processes included in this test and the corresponding FLASH modules are the following. In addition to the standard FLASH PPM hydrodynamic solver, we use the tree solver to calculate the gas self-gravity, and the TREERAY/OPTICALDEPTH module to include the ambient interstellar radiation field (see Paper I for both modules). The ionizing radiation is treated using the TREERAY/ONTHESPOT module described here, and its coupling to the chemistry module is given in Haid et al. (2018). The CHEMISTRY module implements a network with seven active species (H$_2$, H, H$^+$, CO, C$^+$, O, and e^-; see Walch et al. 2015; Glover et al. 2010; Nelson & Langer 1997 for details). The sources are modelled as FLASH sink particles (Federrath et al. 2010), and move under the influence of the gravitational field, but accretion on to sinks is switched off. Stellar winds are treated with the procedure described by Gatto et al. (2017) and implemented in the FEEDBACKSINKS module.

PHYSICAL PARAMETERS. Apart from the number of sources, the parameters are similar to Haid et al. (2018). The computational
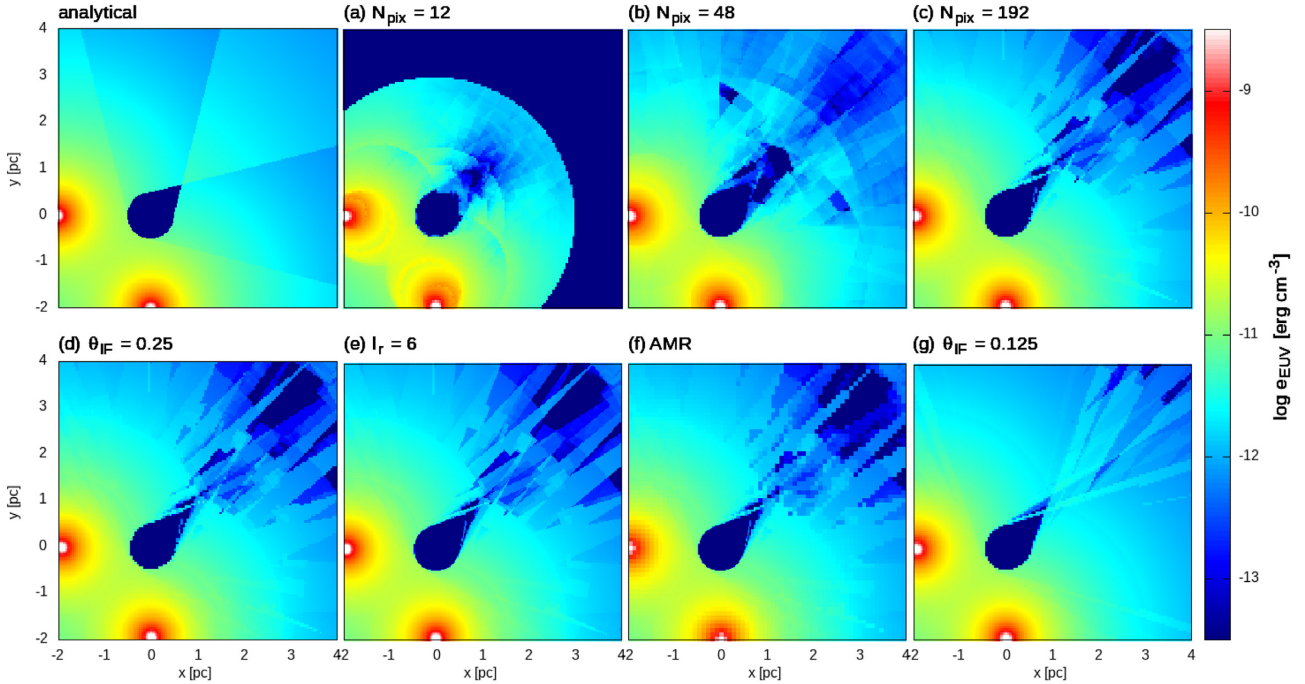
Figure 13. A spherical cloud irradiated by two sources. The top left-hand panel shows the radiation energy density on the $z=0$ plane, calculated analytically. The remaining panels show the same quantity for models (a)–(g).

domain is a $30 \times 30 \times 30 \, \mathrm{pc}^3$ cube and the grid is uniform with refinement level $l_r = 6$ (corresponding to 256^3 grid cells). The hydrodynamic boundary conditions are set to 'diode', and the gravitational boundary conditions are 'isolated'. The interstellar radiation field, from which the heating is calculated by the TREERAY/OPTICALDEPTH module, has strength $G_0 = 1.7$ (Habing 1968; Draine 1978). At the centre of the computational domain is a cold neutral cloud with mass $1.3 \times 10^4 \, \mathrm{M}_\odot$, radius 10 pc, density $2.1 \times 10^{-22} \, \mathrm{g \, cm}^{-3}$, and temperature 20 K. The remainder of the computational domain is filled with a rarefied ambient medium having density $10^{-24} \, \mathrm{g \, cm}^{-3}$. The sources are positioned randomly in a sphere of radius 8 pc, centred on the centre of the cloud. Each of the 100 sources has ionizing output $N_{\mathrm{LyC}} = 2.4 \times 10^{48} \, \mathrm{s}^{-1}$, surface temperature $T = 45\,000 \, \mathrm{K}$, wind mass-loss rate $3 \times 10^{-8} \, \mathrm{M}_\odot \, \mathrm{yr}^{-1}$, and wind velocity $2700 \, \mathrm{km \, s}^{-1}$.

TECHNICAL PARAMETERS. We evaluate this test by comparing runs computed with different TREERAY parameters. The model is most interesting, and computationally most demanding, when (i) a significant fraction of the cloud is ionized, and (ii) there is hot shocked stellar-wind gas. This happens after several hundred kyr, and therefore we only compare the runs during the time period from 500 to 600 kyr. The first 500 kyr are only calculated once, with model (a). The parameters of the fiducial model (a) are selected as a compromise between accuracy and performance. Model (a) uses moderate angular resolution with $N_{\mathrm{PIX}} = 48$ and TREERAY-specific MACs with $\theta_{\mathrm{IF}} = 0.5$ and $\theta_{\mathrm{Src}} = 0.5$. This allows us to adopt a large general opening angle $\theta_{\mathrm{lim}} = 1.0$. Since the presence of very hot gas results in a very short hydrodynamic time-step, we apply the tree solver time-step (see Section 2.9) and set the velocity limit to $v_{\mathrm{tsdt}} = 50 \, \mathrm{km \, s}^{-1}$. Model (b) does not use the tree solver time-step ($v_{\mathrm{tsdt}} = \infty$) and the tree solver is called at each hydrodynamic time-step; hence, (b) is by far the most expensive model. Model (c) differs from the fiducial model by invoking higher angular resolution, with $N_{\mathrm{PIX}} = 192$ and $\theta_{\mathrm{IF}} = \theta_{\mathrm{Src}} = 0.25$. Models (d) and (e) explore the behaviour when

the low velocity limit for the tree solver time-step is reduced to $v_{\mathrm{tsdt}} = 25 \, \mathrm{km \, s}^{-1}$; model (e) also adopts the less demanding error control criterion involving the total radiation energy. Finally, model (f) involves only a single source of radiation and stellar wind, located at the centre of the cloud, with a total photon emissivity and mass-loss rate equal to the sum of the 100 sources of the fiducial model; this model is intended to reproduce model CNM 60 from Haid et al. (2018). The parameters of all models are summarized in Table 4.

MODEL EVOLUTION. Fig. 14 illustrates the first Myr of evolution for the fiducial model (a). At early times, H II regions appear around the sources and start to expand at $\sim 6 \, \mathrm{km \, s}^{-1}$, in agreement with the Spitzer solution (equations 25 and 26). The H II regions around neighbouring sources eventually merge. At $t \simeq 70 \, \mathrm{kyr}$, some hot gas, resulting from shocked stellar winds, appears and quickly expands due to its high pressure; consequently, the H II regions are squeezed between the hot wind bubbles on the inside and the surrounding shells of swept-up cold neutral gas on the outside. The bubbles and H II regions continue to expand and merge, and at the same time the cloud slowly collapses due to its self-gravity. Eventually, at $t \simeq 250 \, \mathrm{kyr}$, some bubbles reach the cloud edge and break out, in the process known as a *champagne flow* (Tenorio-Tagle 1979). Thereafter, the warm photoionized H II, and the shocked hot wind-gas, start to flow out of the cloud, with the shocked hot wind–gas also being accelerated by the buoyancy force. The remainder of the cloud decays into a network of filaments, which expands slowly outwards, accelerated by the pressure of the warm and hot gas, and by the rocket effect (Oort & Spitzer 1955). Some structures formed in the later stages of evolution resemble the *elephant trunks* that are frequently observed in star-forming regions (see e.g. Hillenbrand et al. 1993; McLeod et al. 2015).

COMPARISON OF MODELS. Fig. 15 compares the evolution of models (a)–(e) between 500 and 600 kyr. The thin lines show the total radiation energy in the computational domain (E_{rad}); its maximum fractional difference relative to model (a), $e_{E_{\mathrm{rad}}}$, is given in Table 4.

Table 4. Star formation and feedback test.

model	N_{src}	l_r	θ_{lim}	N_{PIX}	θ_{IF}	v_{tsdt}	$\delta_{e_{EUV}}$	n_{steps}	n_{tr}	e_{Erad}	$e_{M_{H+}}$	t_{hydro}	t_{chem}	t_{part}	t_{tr}	t_{tot}
(a) Fiducial	100	6	1.0	48	0.5.	50	cell	4253	309	–	–	1430	1000	980	950	4600
(b) $v_{tsdt} = \infty$	100	6	1.0	48	0.5.	∞	cell	4227	36278	0.005	0.011	2030	1520	1180	89000	94500
(c) $N_{PIX} = 192$	100	6	1.0	192	0.25	50	cell	4997	358	0.12	0.085	1700	1160	1140	4500	8800
(d) $v_{tsdt} = 25$	100	6	1.0	48	0.5.	25	cell	4253	180	0.008	0.016	1460	990	960	630	4300
(e) $\delta_{e_{EUV}} = $ tot	100	6	1.0	48	0.5.	25	tot	4288	22	0.01	0.032	1440	1010	970	240	3900
(f) 1 source	1	6	1.0	48	0.5.	50	cell	2690	156	–	–	890	670	370	2000	4060

Notes. Column 1 gives the model name. The following columns list:

(i) N_{src}: the number of sources

(ii) l_r: the refinement level defining grid resolution ('5' $\rightarrow$ 128^3; '6' $\rightarrow$ 256^3; '4,6' $\rightarrow$ AMR with minimum and maximum refinement levels 4 and 6, respectively.)

(iii) θ_{lim}: the limiting opening angle for the BH MAC

(iv) N_{PIX}: the number of rays (defining the angular resolution)

(v) θ_{IF}, : the limiting opening angle for the IF MAC, parameter θ_{Src} of the Src MAC is set to the same value

(vi) v_{tsdt}: the velocity limit in km s^{-1} for the adaptive tree solver time-step

(vii) $\delta_{e_{EUV}}$: error control method (either $\delta_{e_{EUV},cell}$ given by equation (22), or $\delta_{e_{EUV},tot}$ by equation (21); see Section 2.6)

(viii) n_{steps}: the number of hydrodynamic time-steps for the whole run

(ix) n_{tr}: number of tree solver iterations for the whole run

(x) e_{Erad}: the maximum fractional difference in the total radiation energy, relative to model (a)

(xi) $e_{M_{H+}}$: maximum relative difference of the total mass of the ionized gas with respect to model (a)

(xii) t_{hydro}: the processor time spent in the hydro module, for the whole run (in core-hours)

(xiii) t_{chem}: the processor time spent in the chemistry module, for the whole run (in core-hours)

(xiv) t_{part}: the processor time in the particles module, for the whole run (in core-hours)

(xv) t_{tr}: the processor time spent in the tree solver, for the whole run (in core-hours)

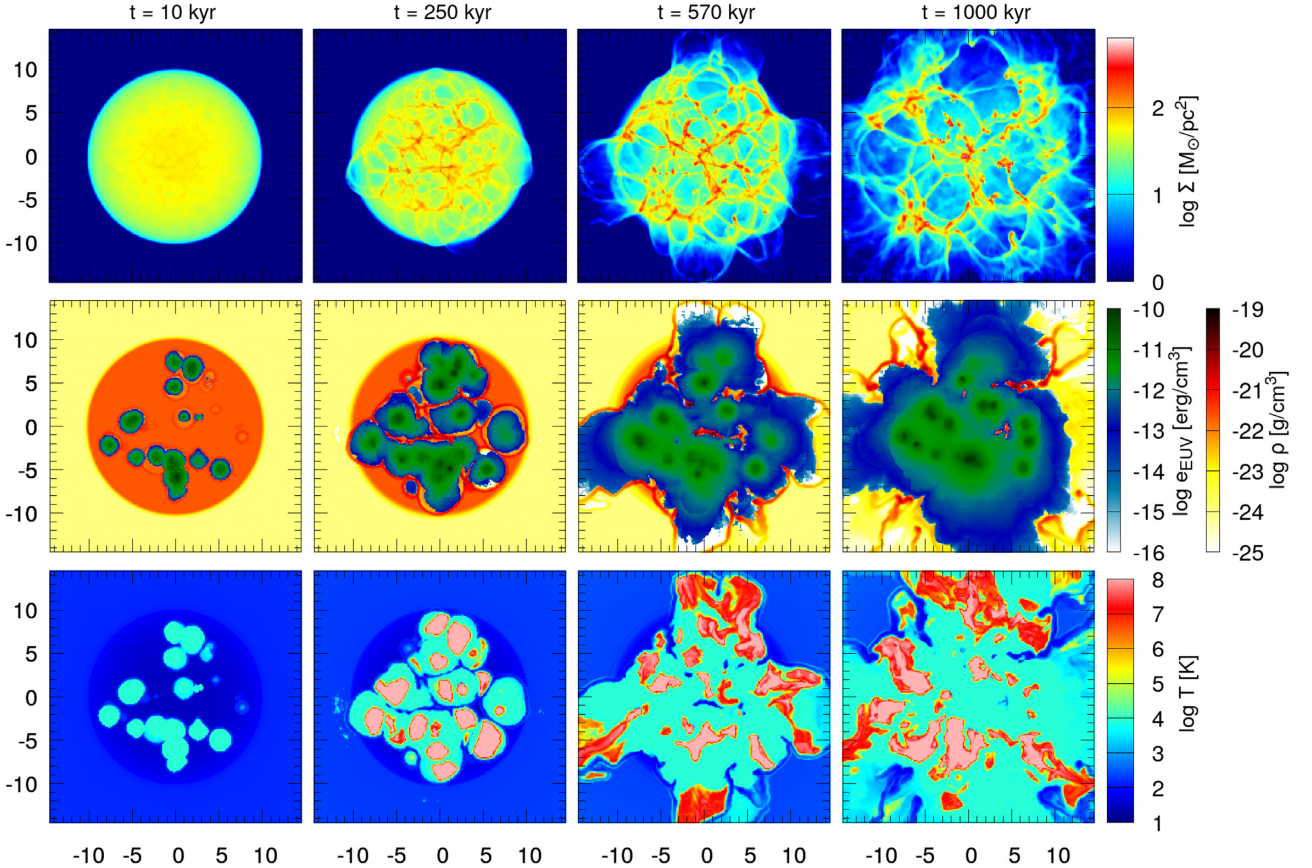(xvi) t_{tot}: the processor time in all modules, for the whole run (in core-hours)

Figure 14. Star formation and feedback test: the fiducial model (a) at times $t = 10$, 250, 570, and 1000 kyr (from the left to right-hand panel). The top row shows the logarithm of the column density. The middle row shows the logarithm of the radiation energy, in the region with non-zero ionization degree, and the logarithm of the gas density, in the remaining parts (i.e. for the neutral gas only). The bottom row shows the logarithm of the gas temperature. Panels in the middle and bottom rows show the quantities on the $z = 0$ plane.
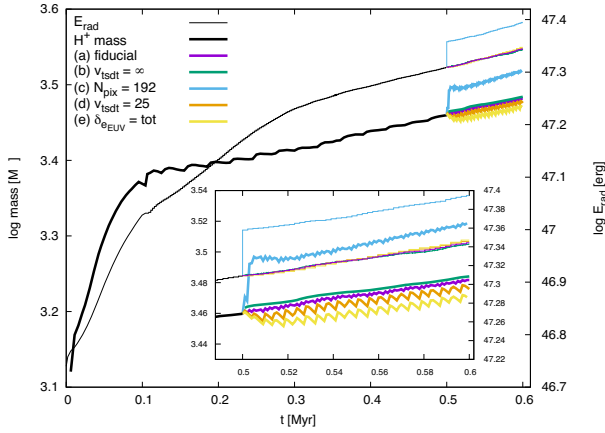
Figure 15. Star formation and feedback: the evolution of the total radiation energy, E_{rad}, (thin lines, right-hand ordinate), and the mass of ionized gas, M_{H^+}, (thick lines, left-hand ordinate), in the whole computational domain, for models (a)–(e). The main figure shows the first 500 kyr of evolution for model (a) only (black lines). The inset shows the evolution between 500 and 600 kyr for all models. Note that, even at the resolution of the inset, the differences in E_{rad} between models (a), (b), (d) and (e) cannot be resolved.

Models (a), (b), (d), and (e) have almost the same E_{rad} with relative differences of order 1 per cent or smaller. For model (c), E_{rad} is higher by ~10 per cent, because its higher angular resolution allows the radiation to follow better the curved surfaces of irregular shells, resulting in slightly larger H II regions. The thick lines in Fig. 15 show the total mass of ionized gas, M_{H^+}, and its fractional difference relative to model (a), $e_{M_{H^+}}$, is again given in Table 4. For model (c) M_{H^+} is higher than for model (a) by ~8.5 per cent, again because of higher resolution. Models (a), (b), (d), and (e) show M_{H^+} differing by of order 1 per cent, with higher M_{H^+} for models with a shorter time between calls to the tree solver (i.e. higher v_{tsdt}). This is because during time-steps when the tree solver is not called, the shells continue to expand and a small fraction of the ionized gas gets into regions that are not irradiated, where it recombines and so M_{H^+} drops unphysically (see periods of decrease in the saw-tooth pattern of models (d) and (e)). We conclude that the angular resolution (i.e. parameters N_{PIX}, θ_{IF}, and θ_{Src}) has an impact on the accuracy of calculations of this type. In contrast, the tree solver time-step parameter, v_{tsdt}, seems to have little impact, provided $v_{tsdt} \gtrsim 50$ km s^{-1}. Indeed, even models (d) and (e) with $v_{tsdt} = 25$ km s^{-1} give satisfactory results, and would be suitable for quick tests scanning the parameter space.

PERFORMANCE. Table 4 shows the total CPU times spent in the four computationally most demanding modules: the hydrodynamic solver (t_{hydro}), the CHEMISTRY module (t_{chem}), the particle module (t_{part}), and the tree solver including TREERAY (t_{tr}). In model (a), the times taken by these four modules are comparable, with $t_{tr} \sim 2t_{hydro}/3$. The small t_{tr} is largely due to setting a finite v_{tsdt}; the tree solver is called only 309 times, while the hydrodynamic solver is called 4253 times (columns n_{tr} and n_{step} in Table 4, respectively). The benefits of setting a finite v_{tsdt} are further illustrated by model (b), where v_{tsdt} is not set to a finite value, and therefore defaults to ∞; the results are indistinguishable from model (a) but the tree solver takes ~90 times more time than in model (a). Model (c), with finer angular resolution, is approximately two times slower than model (a), and the tree solver takes approximately three times longer than the hydrodynamic solver. In models (d) and (e), the tree solver is called less often than in model (a), and t_{tr} is proportionally smaller; model
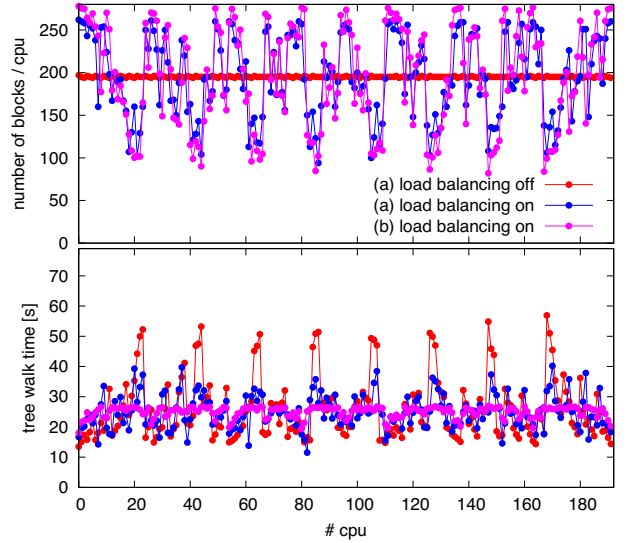

Figure 16. The effect of load balancing in the star formation and feedback test. *Top panel:* the number of blocks on each processor for model (a) with load balancing off (red, flat distribution), model (a) with load balancing on (blue), and model (b) with load balancing on (magenta). Bottom panel: the duration of the tree walk on each processor for the same three cases. The measurements shown have been made at time $t = 500$ kyr, but are representative of the majority of the time.

(e) uses the total radiation energy as the error control, and TREERAY does not need to iterate at all. Model (f), with a single source of radiation and wind, needs almost two times fewer hydrodynamic time-steps, due to the lower maximum temperature of the hot shocked gas. This also results in a smaller number of tree solver calls than in model (a). However, each tree solver iteration takes more time, because the ionization front has larger surface area, and consequently a higher number of tree nodes must be opened.

LOAD BALANCING. Models (a)–(f) have been calculated with load balancing switched on (see Section 2.10). In order to evaluate the impact of load balancing on the tree solver performance, we calculate a few time-steps of model (a), starting at 500 kyr, with load balancing off. The top panel of Fig. 16 compares the number of blocks per processor for model (a) with load balancing on and off, and for model (b) with load balancing on. The bottom panel shows, for the same three cases, the time spent in a single tree walk on each processor. It can be seen that models with load balancing on have smaller variations in the tree walk time, and that the variation is much smaller for model (b) where the tree solver is called at every time-step. Since processors that finish the tree walk earlier have to wait for the slowest processor, we estimate that in model (a) the load balancing decreases the tree walk time from ~60 to ~40 s, saving approximately 30 per cent of the tree solver time.

5 SCALING TESTS

5.1 Weak and strong scaling

HARDWARE FOR SCALING TESTS. We carry out weak and strong scaling tests for the TREERAY algorithm based on the Spitzer test (see Section 3.1). The tests are run on the HPC system COBRA, hosted by the Rechenzentrum Garching at the Max-Planck Computing and Data Facility. COBRA has Intel Xeon 'Skylake' processors. In total, there are 3188 compute nodes with 40 cores @ 2.4 GHz each and
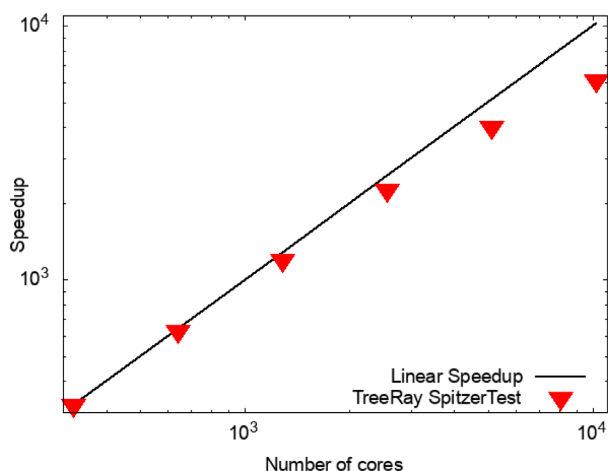
Figure 17. Strong scaling on up to 10240 cores, showing almost linear speedup for the Spitzer test.
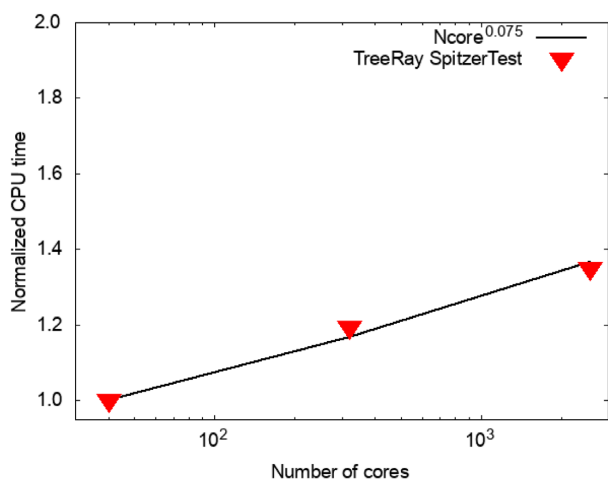

Figure 18. Log-linear plot showing the results of a weak scaling test on 40 to 20480 cores. The scaling within one node is not ideal, but for more than 40 cores, the scaling is close to ideal. A power-law fit gives a very weak dependence on the number of cores $\propto N_{\mathrm{core}}^{0.075}$.

a memory of more than 2.4 GByte per core. The available memory per core is thus at least 2.2 GByte.

STRONG SCALING. Fig. 17 shows the results of the strong scaling test. We plot the time in seconds measured for the tree build, communication, walk, and radiative transfer calculation as measured for 10 time-steps during the Spitzer test. The spatial resolution is set to 512^3 cells, and therefore the average number of blocks per core changes from 819.2 on 320 cores, to 25.6 on 10240 cores. The scaling is very good, showing an almost ideal, linear speedup with the number of cores.

WEAK SCALING. Fig. 18 shows the result of the weak scaling test. The simulations are chosen such that there is the same average number of blocks per core of 102.4. In order to achieve this, we change the resolution of the Spitzer test from 128^3 cells on 40 cores to 512^3 cells on 2560 cores. The processor time in the tree solver only depends very weakly on the number of cores, N_{core}, and can be approximated by a power law, $\propto N_{\mathrm{core}}^{0.075}$. This is close to an ideal weak scaling for which the same amount of wall-clock time should be used for all simulations.
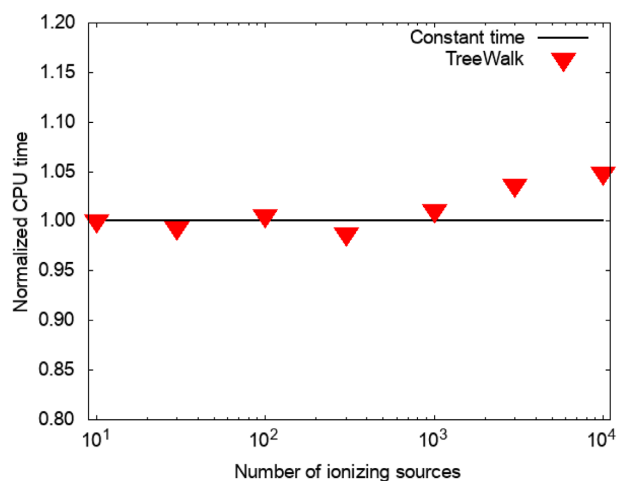
Figure 19. Scaling with the number of sources. The setup consists of a uniform density medium that contains from $N_{\mathrm{src}} = 10$ up to 10^4 randomly distributed ionizing sources. The log-linear plot shows the integration time normalized to the time it takes to calculate 10 time-steps for only 10 sources, i.e. 10 Spitzer bubbles (left-most point). As N_{src} is increased by a factor of 1000, the integration time only increases by ∼5 per cent. This means that the integration time is almost independent of the number of sources.

5.2 Scaling with the number of sources

NUMBER-OF-SOURCES SCALING TEST SETUP. To test the extent to which the performance of the code degrades as the number of ionizing sources is increased, we set up a $30 \times 30 \times 30\,\mathrm{pc}^3$ computational domain, containing atomic hydrogen with uniform density $\rho = 7.63 \times 10^{-22}\,\mathrm{g\,cm}^{-3}$ and uniform temperature $T = 10\,\mathrm{K}$. N_{src} sources are placed randomly in the computational domain, and each source emits ionizing photons at rate $N_{\mathrm{LyC}} = 5 \times 10^{48}\,\mathrm{s}^{-1}$ into an injection region with radius $r_{\mathrm{inj}} = 0.32$ pc (which corresponds to about 1.4 cells). N_{src} is set to 10, 32, 100, 316, 1000, 3162, and 10 000. All setups are evolved for 10 time-steps, on 32 cores, in order to measure the exact integration time.

NUMBER-OF-SOURCES SCALING RESULTS. Fig. 19 shows the resulting simulation times as a function of N_{src}, normalized to the simulation time for $N_{\mathrm{src}} = 10$. For $N_{\mathrm{src}} \lesssim 10^3$ there is essentially no difference in the simulation times. For $N_{\mathrm{src}} > 10^3$, the simulation time increases slightly and becomes ∼5 per cent longer for $N_{\mathrm{src}} = 10^4$ than for $N_{\mathrm{src}} = 10$. We conclude that the simulation time is very nearly independent of the number of sources, as expected from the algorithm design. Therefore, it is an excellent basis for implementing radiation transport schemes where every grid cell represents a source of radiation, e.g. emission from hot gas or dust.

6 SUMMARY

In this paper, we describe TREERAY, a new, fast algorithm for treating radiation transport in gaseous media. It is based on the combination of reverse ray tracing (e.g. Altay & Theuns 2013) and a tree-based (Barnes & Hut 1986) accelerated integration scheme. In general, the incident flux of radiation is computed for every grid cell, but it can also be computed for any other target point in the computational domain, for example the position of a sink particle. From every target point, reverse ray tracing is executed in N_{PIX} directions (hence the angular resolution is user defined), and the directions are interpreted as cones with equal solid angle, based on the HEALPIX scheme (Górski et al. 2005). Due to the equal solid-angle pixelation, every direction's

contribution carries equal weight. In the limit of infinite angular resolution, TREERAY converges to the long characteristics method, which is very accurate but usually prohibitively expensive for time-dependent astrophysical simulations.

TREERAY treats all the gas in the computational domain, and can capture the shadows of even quite small and dense objects, with the limitation that structures at large distances from a target point are smoothed out over a solid angle $\sim 4\pi/N_{PIX}$. The smoothing is controlled by the HEALPIX resolution (user-specified N_{PIX}) and the limiting opening angles set for the MAC (user-specified θ_{lim}, θ_{IF}, θ_{Src}). The number of evaluation points at which the radiative transfer equation is integrated, along a given ray, is of secondary importance.

A key strength of TREERAY is that its computational cost is essentially independent of the number of radiation sources. This enables TREERAY to treat big star clusters with many radiation sources, and extended sources like radiatively cooling shock fronts or cool dust clouds, without incurring an unacceptable computational overhead.

Furthermore, TREERAY scales extremely well with the number of processors, which is due to the communication and local tree-walk strategy of the scheme (see also Paper I). We demonstrate *both* an almost ideal weak scaling up to $\sim 2.5 \times 10^3$ cores, *and* an almost ideal strong scaling on up to $\sim 10^4$ cores (which is usually even harder to achieve).

TREERAY can easily be extended to include additional radiative transfer sub-modules. Additional sub-modules that are already in preparation include the transfer of non-ionizing radiation including the emission from dust and the associated radiation pressure (Klepitko et al., in preparation); the multiwavelength transfer of X-rays originating from point sources such as high-mass X-ray binaries (Gaches et al., in preparation., based on the diffuse X-ray radiative transfer scheme with the TREERAY/OPTICALDEPTH module developed by Mackey et al. 2019); and the multiwavelength transfer of Far Ulraviolet (FUV) and EUV radiation, including the dissociation of molecules (Walch et al., in preparation). We plan in future to include the option to have directionally dependent absorption coefficients, similar to those introduced in (Grond et al. 2019).

DATA AVAILABILITY

The TREERAY module source code and the simulation data underlying this paper will be shared on reasonable request to the corresponding author. We plan to include TREERAY into a future version of the FLASH code.

REFERENCES

Agertz O., Kravtsov A. V., Leitner S. N., Gnedin N. Y., 2013, ApJ, 770, 25 (Paper I)
Altay G., Theuns T., 2013, MNRAS, 434, 748
Baczynski C., Glover S. C. O., Klessen R. S., 2015, MNRAS, 454, 380
Barnes J., Hut P., 1986, Nature, 324, 446
Bertoldi F., 1989, ApJ, 346, 735
Bisbas T. G., Wünsch R., Whitworth A. P., Hubber D. A., 2009, A&A, 497, 649 (B09)
Bisbas T. G., Wünsch R., Whitworth A. P., Hubber D. A., Walch S., 2011, ApJ, 736, 142
Bisbas T. G., Haworth T. J., Williams R. J. R., Mackey J., Tremblin P., Raga A. C., Arthur S. J., and 12 co-authors, 2015, MNRAS, 453, 1324
Clark P. C., Glover S. C. O., Klessen R. S., 2012, MNRAS, 420, 745
Dale J. E., Ercolano B., Bonnell I. A., 2012, MNRAS, 424, 377
Deharveng L. et al., 2010, A&A, 523, A6+
Draine B. T., 1978, ApJS, 36, 595
Elmegreen B. G., Lada C. J., 1977, ApJ, 214, 725
Federrath C., Banerjee R., Clark P. C., Klessen R. S., 2010, ApJ, 713, 269
Fryxell B. et al., 2000, ApJS, 131, 273
Gatto A. et al., 2017, MNRAS, 466, 1903
Gendelev L., Krumholz M. R., 2012, ApJ, 745, 158
Getman K. V., Feigelson E. D., Sicilia-Aguilar A., Broos P. S., Kuhn M. A., Garmire G. P., 2012, MNRAS, 426, 2917
Girichidis P. et al., 2016, MNRAS, 456, 3432
Glover S. C. O., Federrath C., Mac Low M.-M., Klessen R. S., 2010, MNRAS, 404, 2
Górski K. M., Hivon E., Banday A. J., Wandelt B. D., Hansen F. K., Reinecke M., Bartelmann M., 2005, ApJ, 622, 759
Gritschneder M., Naab T., Burkert A., Walch S., Heitsch F., Wetzstein M., 2009, MNRAS, 393, 21
Gritschneder M., Burkert A., Naab T., Walch S., 2010, ApJ, 723, 971
Grond J. J., Woods R. M., Wadsley J. W., Couchman H. M. P., 2019, MNRAS, 485, 3681
Habing H. J., 1968, Bull. Astron. Inst. Neth., 19, 421
Haid S., Walch S., Seifried D., Wünsch R., Dinnbier F., Naab T., 2018, MNRAS, 478, 4799
Haid S., Walch S., Seifried D., Wünsch R., Dinnbier F., Naab T., 2019, MNRAS, 482, 4062
Hillenbrand L. A., Massey P., Strom S. E., Merrill K. M., 1993, AJ, 106, 1906
Iliev I. T. et al., 2009, MNRAS, 400, 1283
Iliev I. T., et al., 2006, MNRAS, 371, 1057
Kahn F. D., 1954, Bull. Astron. Inst. Neth., 12, 187
Kannan R., Vogelsberger M., Marinacci F., McKinnon R., Pakmor R., Springel V., 2019, MNRAS, 485, 117
Kessel-Deynet O., Burkert A., 2003a, MNRAS, 338, 545
Kessel-Deynet O., Burkert A., 2003b, MNRAS, 338, 545
Kim C.-G., Ostriker E. C., 2017, ApJ, 846, 133
Kim J.-G., Kim W.-T., Ostriker E. C., Skinner M. A., 2017, ApJ, 851, 93
Klassen M., Kuiper R., Pudritz R. E., Peters T., Banerjee R., Buntemeyer L., 2014, ApJ, 797, 4
Krumholz M. R., Klein R. I., McKee C. F., 2007, ApJ, 656, 959
Kuiper R., Klahr H., Dullemond C., Kley W., Henning T., 2010, A&A, 511, A81
Lefloch B., Lazareff B., 1994, A&A, 289, 559
Lefloch B., Lazareff B., 1995, A&A, 301, 522

Mackey J., Lim A. J., 2010a, MNRAS, 403, 714
Mackey J., Lim A. J., 2010b, MNRAS, 403, 714
Mackey J., Lim A. J., 2011, MNRAS, 412, 2079
Mackey J., Walch S., Seifried D., Glover S. C. O., Wünsch R., Aharonian F., 2019, MNRAS, 486, 1094
McLeod A. F., Dale J. E., Ginsburg A., Ercolano B., Gritschneder M., Ramsay S., Testi L., 2015, MNRAS, 450, 1057
Mellema G., Raga A. C., Canto J., Lundqvist P., Balick B., Steffen W., Noriega-Crespo A., 1998, A&A, 331, 335
Mellema G., Arthur S. J., Henney W. J., Iliev I. T., Shapiro P. R., 2006, ApJ, 647, 397
Miao J., White G. J., Thompson M. A., Nelson R. P., 2009, ApJ, 692, 382
Nelson R. P., Langer W. D., 1997, ApJ, 482, 796
Oort J. H., Spitzer L., Jr., 1955, ApJ, 121, 6
Osterbrock D. E., 1988, Publ. Astron. Soc. Pac., 100, 412
Paardekooper J. P., Kruip C. J. H., Icke V., 2010, A&A, 515, A79
Peters T. et al., 2017, MNRAS, 466, 3293
Petkova M., Maio U., 2012, MNRAS, 422, 3067
Quirk J. J., 1994, Int. J. Numer. Methods Fluids, 18, 555
Raga A. C., Cantó J., Rodríguez L. F., 2012, MNRAS, 419, L39

Raskutti S., Ostriker E. C., Skinner M. A., 2017, ApJ, 850, 112
Rijkhorst E.-J., Plewa T., Dubey A., Mellema G., 2006, A&A, 452, 907
Rosdahl J., Blaizot J., Aubert D., Stranex T., Teyssier R., 2013, MNRAS, 436, 2188
Rosen A. L., Krumholz M. R., Oishi J. S., Lee A. T., Klein R. I., 2017, J. Comput. Phys, 330, 924
Schneider N. et al., 2012, A&A, 542, L18
Skinner M. A., Ostriker E. C., 2013, ApJS, 206, 21
Spitzer L., 1978, Physical processes in the interstellar medium. A Wiley-Interscience Publication, New York, NY, p. 333
Strömgren B., 1939, ApJ, 89, 526
Tenorio-Tagle G., 1979, A&A, 71, 59
Walch S. et al., 2015, MNRAS, 454, 238
Walch S., Whitworth A. P., Bisbas T. G., Wünsch R., Hubber D. A., 2013, MNRAS, 435, 917
Walch S. K., Whitworth A. P., Bisbas T., Wünsch R., Hubber D., 2012, MNRAS, 427, 625
Whitworth A., 1979, MNRAS, 186, 59
Wise J. H., Abel T., 2011, MNRAS, 414, 3458
Wünsch R., Walch S., Dinnbier F., Whitworth A., 2018, MNRAS, 475, 3393

APPENDIX A: KERNELS FOR RADIAL MAPPING OF NODES ON TO RAYS

In general, any kernel depends on three tree parameters: the distance from the node mass centre to the target point, d, the node linear size, h_n, and the distance from the ith evaluation point on the ray to the target point, r_i. It is convenient to combine them into a single parameter

$$x \equiv (r_i - d)/h_n. \tag{A1}$$

Kernel evaluation is a performance-critical operation because it is carried out during the tree walk for each target point, each tree node and each intersecting ray. Therefore, it is useful to tabulate the kernels. This is easy to do, because the parameters r_i and h_n can have only a small number of discrete values. The possible values of r_i are given by equation (4); the possible values of h_n are given by the AMR octal tree and their number is the binary logarithm of the ratio of the computational domain size to the size of the smallest grid cell. Parameter d can have an arbitrary value between 0 and the 3D diagonal of the computational domain, and we sample it similarly to r_i but with ten times more points. The kernels are normalised so that, for each tabulated combination of d and h_n, the sum of the values at all r_i is unity.

The left-hand panel of Fig. A1 shows the three implemented kernels, W_g (Gaussian), W_p (piece-wise polynomial), and W_f (flux conserving), as functions of r_i for two combinations of d and h_n. The right-hand panel of this figure displays the error in the radiation energy density when these kernels are used to calculate a model similar to the two-source model described in Section 3.4. However, here the dense cloud is missing, i.e. the whole computational domain is filled with rarefied gas ($\rho = 10^{-24}\,\mathrm{g\,cm^{-3}}$), and the two sources are located closer to each other [at $(x, y, z) = (-0.4, 0, 0)$ pc and $(0, -0.4, 0)$ pc]. The latter modification was made to increase the fraction of target cells for which the two sources are located in the same tree node. Otherwise, all the parameters are the same as for model (b) described in Table 3. The right-hand panel of Fig. A1 shows the discrepancy between the numerically obtained radiation energy density, $e_{\mathrm{EUV,num}}$, and the corresponding analytical value, $e_{\mathrm{EUV,anl}}$, normalized by $e_{\mathrm{EUV,anl}}$ at the point $(x, y, z) = (-0.2, -0.2, 0)$ pc, along the line of the symmetry between the sources ($y = x, z = 0$).
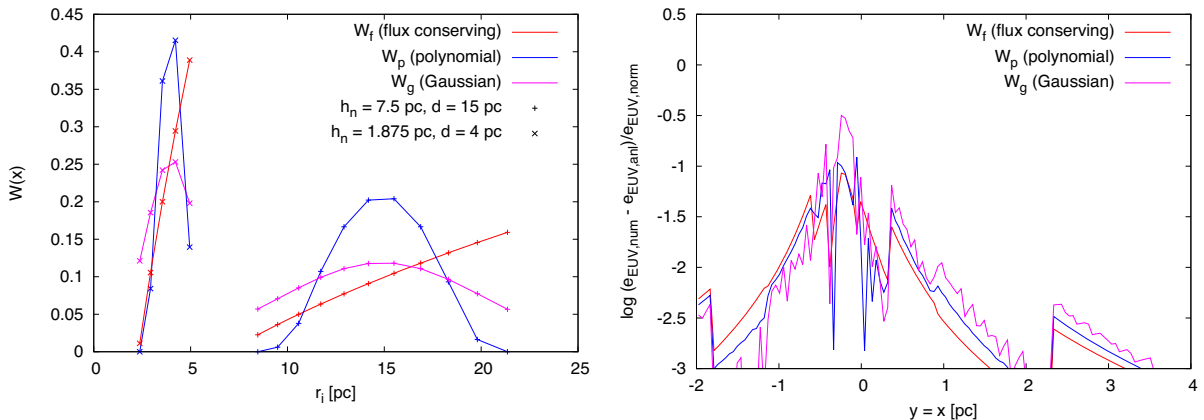

Figure A1. Left-hand panel: the three implemented kernels W_g (magenta), W_p (blue), and W_f (red) plotted as functions of r_i for two combinations of d and h_n: $d = 4$ pc, $h_n = 1.875$ pc ($\times$ symbols), and $d = 15$ pc, $h_n = 7.5$ pc ($+$ symbols). The positions of the evaluation points r_i are calculated from equation (4) with $\eta_R = 2$ and $\Delta x = 0.23$ pc corresponding to model (a) of the Spitzer test (Section 3.1). Right-hand panel: the error in the radiation energy density along the line $y = x, z = 0$ in the test with two radiation sources as calculated using the three kernels W_f, W_p, and W_g. The meaning of colour is the same as in the left-hand panel.
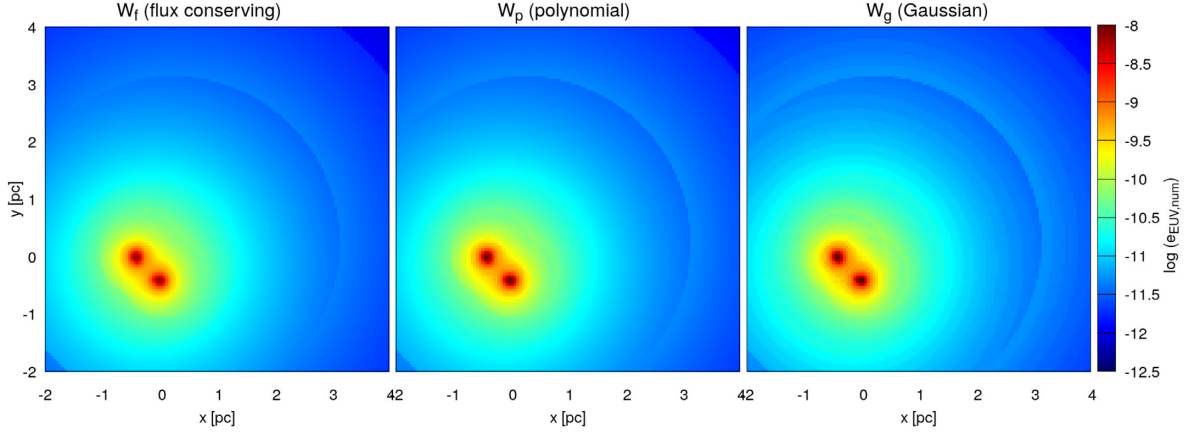
Figure A2. The radiation energy density $e_{\text{EUV,num}}$ calculated using the three kernels W_{f} (left-hand panel), W_{p} (middle panel) and W_{g} (right-hand panel).

The corresponding radiation energy density on the $z = 0$ plane for all three models is displayed in Fig. A2. We can see that the smallest error and the least pronounced numerical artifacts (visible as discontinuities in the radiation field) are obtained with the flux conserving kernel W_{f}. Conversely, the Gaussian kernel W_{g} yields the worst results. Detailed descriptions of the kernels are given below.

A1 Gaussian kernel

The Gaussian kernel is

$$W_{\text{g}}(x) = \begin{cases} \exp(-x^2), & \text{for } x < \sqrt{3}/2; \\ 0, & \text{for } x \geq \sqrt{3}/2. \end{cases} \tag{A2}$$

A2 Piece-wise polynomial kernel

Kernel W_p is derived under the assumption that the emission and/or absorption coefficients – or generally any field that the kernel represents – are distributed uniformly within a cubic tree node. The tree node intersects with randomly oriented, randomly positioned rays (i.e. they do not have to pass through the node center) and the kernel represents the average cross-section taken over all ray positions and orientations. With this motivation, the kernel is obtained by summing up a large number of top-hat functions with extents given by the intersection of a unit cube and a randomly oriented line. The sum is then normalized and fitted with a third-order piece-wise polynomial consisting of two parts $W_{\text{p},1}$ between $x = 0$ and $1/2$ and $W_{\text{p},2}$ between $x = 1/2$ and $\sqrt{3}/2$. The central value is $W_{\text{p},1}(0) \equiv M$ and the kernel drops to zero at $\sqrt{3}/2$. We require that the kernel be continuous, and hence $W_{\text{p},1}(1/2) = W_{\text{p},2}(1/2) \equiv A$. Furthermore, we require that its derivative is zero at the center and at the edge, i.e. $W'_{\text{p},1}(0) = 0$ and $W'_{\text{p},2}(\sqrt{3}/2) = 0$. The resulting kernel has the form

$$W_p(x) = \begin{cases} a_1 x^3 - (a_1/2 + 4(M-A))x^2 + M, & \text{for } x < 1/2; \\ a_2 x^3 - (a_2(1+2\sqrt{3})/2 - 2A(\sqrt{3}+2))x^2 - \\ (9a_2/4 - a_2(\sqrt{3}+6) + 2A(2\sqrt{3}+3))x + 3\sqrt{3}a_2/4 - 3a_2(1+2\sqrt{3})/8 + 3A(\sqrt{3}+2), & \text{for } 0.5 < x < \sqrt{3}/2; \\ 0, & \text{for } x > \sqrt{3}/2 \end{cases} \tag{A3}$$

with

$$a_1 = 1.21, \quad a_2 = -6.35, \quad A = 4.08, \quad \text{and } M = 1.184. \tag{A4}$$

A3 Flux conserving kernel

To illustrate a limitation of the preceding kernels, we consider the situation in which a point source is located in a single node, as seen from a target cell, and the gas is so rarefied that the absorption is negligible. During the tree walk, individual nodes are mapped on to rays, and the location of the sources of radiation within the node is approximated by the node mass centre. The smoothing kernels have large wings spanning from $|r_i - d| - \sqrt{3}h_n/2$ to $|r_i - d| + \sqrt{3}h_n/2$, which together with the typical condition for node acceptance $d \geq \theta_{\text{lim}}h_n$, and with the typical value of the limiting angle $\theta_{\text{lim}} = 0.5$, means that the source is spread out so that it contributes to evaluation points significantly closer to (and further from) the target cell than the original point source. When the emission coefficient is mapped on to a ray using a wide kernel, the source is often spread over many ray evaluation points and each of them becomes a source of radiation. It is then very unlikely that the flux at the target cell due to these multiple "sources" at the ray evaluation points is the same as the flux due to the original single source at distance d. Moreover, as the RTE is integrated along the ray from the source, the flux at the evaluation point r_i is the sum of contributions from the flux coming through the evaluation point r_{i+1}, and from the possible source term at the evaluation point r_i. This is clearly not equal to the situation where the flux is coming from a single-point source.
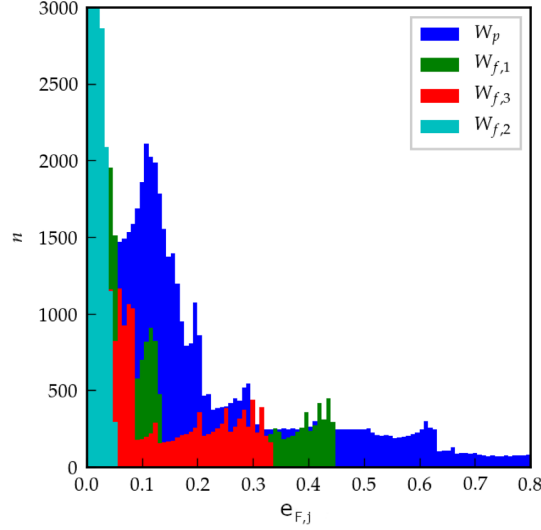
Figure A3. Histograms of relative errors in the radiation flux, $e_{F,j}$ (see equation A6) at evaluation points r_j for the three variants of the flux-conserving kernel and the polynomial kernel (see equation A3).

To overcome this limitation, we consider a kernel which conserves the flux at the target point by construction. In the following, we neglect the ionization rates for the sake of simplicity, so the radiation transport reduces to the situation in the absence of the absorption. Consider a kernel which is so narrow that a single source is only mapped on to two evaluation points along a ray. Hence, the emission coefficient, ε, of a source at distance d must be divided between just two evaluation points, r_i and r_{i+1}, such that the flux F_0 at $r = 0$ is as close as possible to the correct value $\varepsilon/(4\pi d^2)$. We seek a numerical solution, since a general analytic solution for wide kernels with many evaluation points probably does not exist. The above condition for the emission coefficients ε_i and ε_{i+1}, at r_i and r_{i+1}, respectively, gives

$$\frac{\varepsilon}{d^2} = \frac{\varepsilon_i}{r_i^2} + \frac{\varepsilon_{i+1}}{r_{i+1}^2}. \tag{A5}$$

A second condition is needed to find the particular values of ε_i and ε_{i+1}. It can be formulated by minimizing the relative error of the flux, $e_{F,j}$, at the remaining evaluation points r_j,

$$e_{F,j} = \frac{(r_j - d)^2}{\varepsilon} \left| \frac{\varepsilon_i}{r_{j,i}^2} + \frac{\varepsilon_{i+1}}{r_{j,i+1}^2} - \frac{\varepsilon}{(r_j - d)^2} \right|, \tag{A6}$$

where $r_{j,i} = r_j - r_i$ and $r_{j,i+1} = r_j - r_{i+1}$. We explore three possible variants of the flux-conserving kernel.

The first variant, $W_{f,1}$, is constructed using the second condition of the form $\varepsilon_{i+1}/\varepsilon_i = (d - r_i)/(r_{i+1} - d)$, where the evaluation points satisfy the constraint $r_i \leq d \leq r_{i+1}$.

The second variant, $W_{f,2}$, is based on a second condition of the form $\varepsilon_i/(r_i - r_{i-1})^2 + \varepsilon_{i+1}/(r_{i+1} - r_{i-1})^2 = \varepsilon/(d - r_{i-1})^2$, which means that the flux is exact at the evaluation point r_{i-1}. Since the flux is then exact at two points along the ray (at $r = 0$ and $r = r_{i-1}$), it should be close to the correct value everywhere else on the ray between 0 and r_{i-1}.

The third variant, $W_{f,3}$, uses a simple second condition of the form $\varepsilon_i + \varepsilon_{i+1} = \varepsilon$.

If the kernel spans of larger number of evaluation points, N, the two conditions defining the generalized kernel $W_{f,3}$ are

$$\frac{\varepsilon}{d^2} = \sum_{i=j}^{j+N-1} \frac{\varepsilon_i}{r_i^2}, \tag{A7a}$$

$$\varepsilon = \sum_{i=j}^{j+N-1} \varepsilon_i. \tag{A7b}$$

A further $N - 2$ conditions are then required, and these are generated by setting a linear relationship between the emission coefficient ε_j and its index j, i.e. $\varepsilon_j = \varepsilon \times (a + b(j - i))$ for $i \leq j \leq i + N - 1$, where i is the lowest index for which the kernel yields a non-zero value. With this prescription, equations (A7a) and (A7b) give

$$b = \left(\frac{1}{d^2} - \frac{A_1}{N} \right) \left[A_2 - \frac{(N-1)A_1}{2} \right]^{-1}, \tag{A8a}$$

$$a = \frac{1}{N} - \frac{b(N-1)}{2}, \tag{A8b}$$

with

$$A_1 = \sum_{i=j}^{j+N-1} \frac{1}{r_i^2}, \tag{A9a}$$

$$A_2 = \sum_{i=j+1}^{j+N-1} \frac{i-1}{r_i^2}.$$ (A9b)

Note that this kernel is non-local in the sense that the positions of all the evaluation points over which the kernel is non-zero have to be known to obtain the kernel coefficients at a certain point. This is straightforward in TREERAY, because the positions of all the evaluation points on all the rays are known a priori (see equation 4).

Fig. A3 compares the relative error in the radiation flux, $e_{F,j}$, calculated using the three variants of the flux-conserving kernel, $W_{f,1}$ (green), $W_{f,2}$ (cyan), and $W_{f,3}$ (red). For reference, the relative error obtained using the polynomial kernel, W_p, is also shown (blue). Each histogram was obtained by placing a source at 10^4 different positions on a ray with 10 evaluation points placed according to equation (4). $W_{f,2}$ leads to the most accurate results, but $W_{f,3}$ also leads to acceptable results and is much simpler. Therefore, we use $W_{f,3}$ as the default in TREERAY.

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