CRASH2: coloured packets and other updates

A. Maselli,1⋆ B. Ciardi1 and A. Kanekar1,2

1Max-Planck-Institut für Astrophysik, Karl-Schwarzschild-Strasse 1, 85748 Garching, Germany
2Indian Institute of Technology, Mumbai, India

Accepted 2008 October 30. Received 2008 October 28; in original form 2008 October 8

ABSTRACT

In this paper, we report on the improvements implemented in the cosmological radiative transfer code CRASH. In particular, we present a new multi-frequency algorithm for spectra sampling which makes use of coloured photon packets: we discuss the need for the multi-frequency approach, describe its implementation and present the improved CRASH performance in reproducing the effects of ionizing radiation with an arbitrary spectrum. We further discuss minor changes in the code implementation which allow for more efficient performance and an increased precision.

Key words: radiative transfer – methods: numerical – intergalactic medium – cosmology: theory.

1 INTRODUCTION

The proper and accurate treatment of radiative transfer (RT) in cosmological numerical simulations is a longstanding problem, which has received great and ever-increasing attention in recent decades. Despite this, the progress made in numerical RT has been somewhat slower with respect to other fields of numerical cosmology and astrophysics, like large-scale structure simulations or magnetohydrodynamics schemes, which in recent years underwent an impressive series of successes and refinements. The reason for the lag above is mainly due to the extreme complexity underlying the process of RT, which in a cosmological setting couples also with gas dynamics, cosmic expansion, structure formation and chemistry.

The seven-dimensional cosmological RT equation does not allow an exact solution, either analytic or numerical, and the need to deal with arbitrary geometries and with a huge dynamical range in the optical depth distribution prevents the adoption of simplifying assumptions to reduce the dimensionality of the equation and the complexity of the problem. Several methods have been proposed in the literature, relying on a variety of assumptions and approximations, and often optimizing the treatment of RT for different problems, namely different astrophysical configurations. For a detailed review of the various methods developed in the literature, we refer the reader to Iliev et al. (2006, hereafter I06), where a systematic comparison of 11 cosmological RT codes through a set of tests is described. Since the publication of the paper, more RT codes designed for cosmological applications have appeared in the literature (Altay, Croft & Pelupessy 2008; Finlator, Ōzel & Davé 2008; Pawlik & Schaye 2008).

The strategy adopted by our group to face the problem of cosmological RT is based on a ray-tracing Monte Carlo (MC) scheme which applies to three-dimensional Cartesian grids and which is based on the particle nature of the radiation field. This approach offers several advantages. The radiation is described in terms of photons which, grouped for computational convenience into monochromatic photon packets, travel through the simulation volume along rays. This allows us to solve the transfer in one dimension: (i) by following monochromatic packets of photons along rays one gets rid of the explicit dependence on direction and frequency, and (ii) by casting rays into the assigned grid, one can furthermore remove the dependence on the position. Instead of solving directly for the intensity of the radiation field, \(I_{\nu}(r, \Theta, \iota)\), one just needs then to model the interaction of the photons with the gas inside a cell. This approach allows us to enormously simplify the treatment of RT, and to leave the geometry of the problem completely arbitrary. On the other hand, as a drawback, ray-tracing techniques are usually computationally expensive and tend to suffer from low angular resolution. These problems are partially overcome by adopting a statistical description of the radiation field, by MC sampling directions and by spectral energy distributions (SED). Differently from short and long characteristic methods, in which the rays are cast along fixed directions, in MC ray tracing the rays are shot in random directions, allowing for a more efficient implementation of multiple sources and a diffuse component of the ionizing radiation, for example recombination radiation or the ultraviolet background (UVB), as well as anisotropic angular emissivities. These advantages come with the introduction of numerical noise, intrinsic to all MC schemes, which can be nevertheless kept arbitrarily low by increasing the number of rays shot. In practice, one always needs to look for the best compromise with computational efficiency.

All the above ingredients have been successfully implemented in our code CRASH which is to date one of the main references among RT numerical schemes used in cosmology. The first version of the code (Ciardi et al. 2001, hereafter C01; in the following we will refer to this version as CRASH0) integrated the principles of ray tracing...
and MC to follow the evolution of hydrogen ionization for multiple sources under the assumption of a fixed temperature for the ionized gas. In its actual reference version (Maselli, Ferrara & Ciardi 2003, hereafter MFC03; Maselli & Ferrara 2005; in the following we will refer to this version as CRASH1), the code has been further developed by including a variety of additional physical ingredients (like helium chemistry, temperature evolution, background radiation), and a more sophisticated algorithm for processes like photon absorption, re-emission, as well as for the chemistry solver. All these new ingredients make the code extremely versatile in terms of problems that can be studied.

One of the main advantages of the CRASH implementation consists of the straightforward way in which the matter/radiation interactions are modelled and calculated. This allows us to easily implement new physics in the code without the need for major changes in its structure. Illustrative examples are given by MCLYR (Verhamme, Schaerer & Maselli 2006) and CRASH2 (Pierleoni, Maselli & Ciardi 2008). Adopting the CRASH algorithms for ray tracing, spectra sampling, photon propagation and optical depth sampling (as described in C01), we have implemented the physics of line resonant scattering in MCLYR, a tool mainly dedicated to the study and modelling of Lyα emitters (LAEs). Moving forward in this direction, we extended CRASH and implemented CRASH2, the single RT code to date which can self-consistently account for the simultaneous and coupled transfer of line and ionizing continuum radiation. This makes CRASH2 a unique tool for studying the implications of high-redshift LAE surveys on cosmic reionization and the 21-cm line from neutral hydrogen in the primordial Universe. More recently, we have started including molecular hydrogen in CRASH to study the inhomogeneous H2 photo-dissociation occurring at the dawn of the Universe or in star-forming regions. Once again, this requires very little manipulation on the main core of the code and can be done by complementing the existing numerical scheme with algorithms which model the new physics to be implemented.

As the code is continuously being updated and optimized, in this paper we describe some major changes that have been introduced in the implementation described in MFC03, together with some test results clearly showing the improvements achieved in performance and accuracy.

The rest of the paper is organized as follows. We first give a brief review on the basics of the CRASH implementation in Section 2. In Section 3, we describe the updates done in the implementation and the new algorithms introduced, presenting the results from test runs designed to show the improvements obtained. Finally, we summarize our work in Section 4.

2 CRASH: A BRIEF DESCRIPTION

In this section, we give a very brief and general description of the CRASH code. A more detailed and complete report is given in MFC03 and also in Maselli & Ferrara (2005), where a refined and improved version of the implementation of the background radiation is described. We refer the interested reader to these works for more details on the implementation.

As discussed in the Introduction, CRASH is a RT numerical scheme which applies to three-dimensional Cartesian grids and which is based on MC ray-tracing techniques that are used to sample the probability distribution functions (PDFs) of several quantities involved in the calculation, for spectrum of the sources, emission direction, optical depth. The algorithm follows the propagation of the ionizing radiation through an arbitrary H/He static density field and, at the same time, computes (i) the evolution of the thermal and ionization state of the gas and (ii) the distortion of the ionizing radiation field due to matter–radiation interactions, which produce the typical RT effects of filtering, shadowing and self-shielding. The radiation field is described in terms of photons which, for computational convenience, are grouped into photon packets and shot through the box. Both multiple point sources, located arbitrarily in the box, and diffuse radiation fields (e.g. the UVB or the radiation produced by H/He recombinations) are treated.

The energy emitted by each ionizing source is discretized into photon packets emitted at regularly spaced time intervals. More specifically, the total energy radiated by a single source of luminosity $L_s$, during the total simulation time $t_{\text{sim}}$, is $E_s = \int_0^{t_{\text{sim}}} L_s(t_s)dt_s$. For each source, $E_s$ is distributed in $N_p$ photon packets, emitted at the source location. In this way, the total number of photon packets emitted by each source, $N_p$, is the main control parameter which sets both the time and the spatial resolution of a given run. The emission direction of each photon packet is assigned by MC sampling the angular PDF characteristic of the source.

The propagation of the packet through the given density field is then followed and the impact of radiation–matter interaction on the gas properties is computed on the fly. Each time a photon packet pierces a cell $k$, the cell optical depth to ionizing continuum radiation, $\tau_x^k$, is estimated summing up the contribution of the different absorbers (H1, He1, He2). As the probability for a single photon to be absorbed in such a cell is

$$P(\tau_x^k) = 1 - e^{-\tau_x^k},$$

the number of photons absorbed in the cell $k$ is the fraction $P(\tau_x^k)$ of packet content when it enters the cell. The total number of photons deposited in the cell is then distributed among the various absorbers according to the corresponding contributions given to the total cell optical depth. From these quantities, the discrete increments of the ionization fractions and temperature are calculated:

$$\Delta x_{\text{H1}}, \Delta x_{\text{He1}}, \Delta T_{\text{He2}}, \Delta T_{\text{He1}}$$

where $\Delta T$ is the positive increment in temperature due to photo-heating and $(\Delta T)_n$ is the negative one associated with the increased number of free particles. The trajectory of the photon packet is followed until its photon content is extinguished or, if periodic boundary conditions are not assumed, until it exits the simulation volume.

The time evolution of the gas physical properties (ionization fractions and temperature) is computed solving in each cell the appropriate set of discretized differential equations each time the cell is crossed by a packet. In addition to the impact of photo-ionization and photo-heating that, as discussed above, are included in the form of discrete contributions, the implemented chemistry network explicitly includes the other relevant radiative processes: recombination and collisional ionization for the ionization fractions and various cooling processes for the temperature (bremstrahlung, Compton cooling/heating, collisional ionization cooling, collisional excitation cooling and recombination cooling). All these radiative processes are instead implemented in the code as continuous, namely by integrating their associated rate coefficients.

We again refer the reader to the documentation mentioned above for further details on the integration technique and on the technical implementation devised to treat multiple sources.

3 WHAT IS NEW IN CRASH

3.1 Coloured packet

In this section, we describe the new implementation (in the following we will refer to it as CRASH2) developed to deal with ionizing
sources with arbitrary spectra. In CRASH1, each point and diffuse ionizing source is assigned with an arbitrary spectrum which is reproduced by MC sampling the given SED and by emitting a large number of monochromatic photon packets. This implementation is highly versatile, allowing us to reproduce arbitrary spectral shapes in a continuum range of frequency, but has two main drawbacks that could affect the precision and efficiency of the code. First, the nature of the MC sampling procedure induces a poor sampling of the spectral region with lower intensity, which in most applications of interest is the hard energy tail of the spectrum. This limitation could be crucial in studies involving, for example helium (He II) photo-ionization, hard radiation photo-heating (X-rays), ionization front (IF) profiles of hard sources like quasars, just to give some examples among many. Secondly, the monochromatic packet approach aggravates the well-known shortcoming of ray-tracing techniques, which oversample the regions close to the sources at the expense of undersampling the farthest ones. This introduces unphysical features in the RT results, such as spikes and uneven IFs (e.g. see fig. 12 in I06).

To overcome these problems, we have introduced a new algorithm in CRASH which treats the polychromatic nature of the ionizing radiation field by shooting ‘coloured’ packets. Each packet is now represented by a vector of \( N\nu \) bins, whose total energy \( \Delta E \) (determined as in CRASH1) is distributed among a number of photons which populates the \( N\nu \) frequency bins to reproduce the SED characteristic of the ionizing source. The number and spacing of bins can be arbitrarily chosen together with the frequency range covered. Typically, the spacing is chosen to be logarithmic and more refined at the photo-ionization thresholds, that is \( \nu_{\text{HI}}, \nu_{\text{HeI}} \), and \( \nu_{\text{HeII}} \). Although we now have polychromatic photon packets travelling through the box, their interaction with the gas has not been modified significantly with respect to CRASH1.

While propagating through the gas the coloured packet sees different optical depths, one for each frequency bin:

\[
\tau_c(\nu_j) = \sum_{i=\text{HI,HeI,HeII}} \tau_i(\nu_j) = \sum_{i=\text{HI,HeI,HeII}} \delta l \times \sigma_i(\nu_j) \times n_i, \quad (2)
\]

where the indexes \( i \) and \( j \) span over the species included and the frequencies sampled, respectively. The other symbols in equation (2) have the conventional meaning, \( \sigma_i \) and \( n_i \) being the photo-ionization cross-section and the number density of the \( i \)th species, and \( \delta l \) the actual path that the packet travels inside the cell. Differently from the CRASH1 implementation, here we have introduced a new algorithm which computes the length \( \delta l \) of the ray segment within each cell (see the next section). Once the opacity of the cell is determined, the number of photons deposited is calculated according to the probability for a photon to be absorbed, \( P(\nu_i) = (1 - e^{-\tau_c(\nu_i)}) \), which depends on the atomic species and the frequency.

Summarizing, a packet is emitted from a source with \( N_{\nu}^{(0)}(\nu_j) \) photons per frequency bin at the time \( t_0 \). While travelling through the gas, photons are progressively removed from the packet as they ionize atoms along the path. The number of photons deposited in a cell is evaluated from the total optical depth, which gives the probability for an absorption event:

\[
N_{\nu}^{(c)}(\nu_j) = N_{\nu}^{(c-1)}(\nu_j) \left[ 1 - e^{-\tau_c(\nu_j)} \right], \quad (3)
\]

where \( c \) counts the number of cells crossed by the packet from its emission location, \( N_{\nu,c} \) the number of photons absorbed and \( N_{\nu,c} \) the number of photons which is transmitted to the next cell.

The absorption probability in the various frequency bins induces a modification in the spectral shape of the packet. Typically, the frequencies closer to the resonant photo-ionization thresholds are depleted first and more strongly. In Fig. 1, we show an example of how the population of the frequency bins changes while the packet travels through an H/He gas. The differently coloured curves represent the spectral shape of the packet in the different cells pierced, with the sequence from top to bottom corresponding to the order in which they are crossed. Note that for both panels each colour represents the cell after a given number of crossings, for example the transitions dark blue-to-magenta and purple-to-green roughly correspond to the 20th and 30th cell crossed. The two panels show the evolution of the spectral shape of a packet emitted (i) in a completely neutral gas (left-hand panel) and (ii) when the H/He II IFs are roughly 30 cells away from the source (right-hand panel). It can be seen how the photons close to the photo-ionization thresholds are absorbed first, due to the higher cross-sections, and how the SED of the packet becomes progressively harder as it moves far away from the source. It is interesting to note that the spectrum evolution shown in the left-hand panel does not exhibit the drop at 54.4 eV, as here the packet is emitted in a completely neutral gas, still devoid of He II absorbers.

A proper modelling of this filtering mechanism is very important, particularly to correctly estimate the gas temperature or the inner structure of the IFs.

1 The physical configuration of the simulation analysed here is the same of the CRASH/CLOUDY comparison described in Section 4.3, and the two panels correspond to packets emitted at the simulation times \( t_c = 0 \) yr (left-hand panel) and \( t_c = 6 \times 10^4 \) yr (right-hand panel).
Once the total number of photons deposited in a cell from each frequency bin $v_j$ has been determined, $N_{A,j}(v_j)$, they are distributed among the absorber species included in the calculation. This is done by assigning to each absorber species a fraction of $N_{A,j}(v_j)$ proportional to the corresponding absorption probability

$$P_i(v_j) \left[1 - e^{-\tau_i(v_j)}\right],$$

normalized to the total probability:

$$P_i(v_j) = \sum_{i=1}^{\text{th species}} P_i(v_j).$$

As the packet passes through the cell, it photo-ionizes a number of atoms of the $i$th species given by

$$N_{A,j}(v_j) = N_{A,j}(v_j) \times \frac{P_i(v_j)}{P(v_j)}$$

with a corresponding increment in the ionization fractions given by

$$\Delta x_i = \sum_j N_{A,j}(v_j) \times \frac{A,\gamma}{n_i}.$$ (7)

where $i^+$ is the singly ionized ion of the $i$th species ($i^+ \in \{\text{H}^+, \text{He}^+, \text{He}^{++}\}$ for $i \in \{\text{H}, \text{He}, \text{He}^+, \text{He}^{++}\}$), and $N_{ni}$ the total number of nuclei of a given element contained in the actual cell.

The cell temperature before the current packet crossing, $T_c$, is updated accounting for photo-heating and for the variation in the number of free particle which is given by the variation in the number density of free electrons $n_e$. The two effects contribute, respectively, with the following temperature increments:

$$\Delta T_{T^+} = \frac{2}{3\kappa_B n_e} \sum_j N_{A,j}(v_j) (h \nu_{i^+} - h \nu_{i^+}) V_{cell},$$

$$\Delta T_n = -\frac{2}{3} T_c \frac{\Delta n_e}{n_e},$$

where $h \nu_{i^+}$ is the ionization threshold for photo-ionization of the $i$th species, $n_e$ is the number density of free particles and $\Delta n_e$ is the variation of this quantity corresponding to the change in the electron number density due to photo-ionization:

$$\Delta n_e = \Delta x_e = \left[f_b \times \Delta N_{\text{H}^0} + f_{\text{He}} \times (\Delta N_{\text{He}^+} + 2\Delta N_{\text{He}^{++}})\right].$$ (10)

Each time a coloured packet crosses a new cell, the set of vectorial operations described in equations (2) through (10) is performed to get the key quantities: the number of photons which escape the $c$th cell crossed from the emission location, $N_{P,\gamma}^{c-1}(v_j) = N_{P,\gamma}^{c-1} - N_{P,\gamma}^{c-2}$, which accounts for RT, and $\Delta x_{T^+}, \Delta T_{T^+}$ and $\Delta T_n$ necessary to account for the evolution of the gas physical state.

The radiative processes which are not associated with the ionizing radiation are reproduced, as done in CRASH1, by integrating the corresponding temperature-dependent rates over the elapsed time interval between the previous packet crossings and the current one.

### 3.2 Ray casting

To limit the computational time, in the previous versions of CRASH ray casting is not implemented. Rather, the opacity of a single cell is evaluated using as the length travelled by the packet within the cell the fixed value of 0.56 (in units of the linear dimension of a cell). This number corresponds to the median value of the PDF for the length of randomly oriented paths within a cubic box of unit size (see C01).

Although this approximation is acceptable for several applications, it does not always result in enough accuracy, occasionally showing spurious geometrical patterns in the ionization and temperature maps (see the rings evident in the bottom right-hand panel of Fig. 2). For this reason, a new algorithm for ray casting has been implemented, which is similar to the Fax Voxel Traversal Algorithm for Ray Tracing by Amanatides & Woo (1987). Ray casting is performed by computing the distances that need to be travelled in order to intersect the boundaries (along the $x$-, $y$- and $z$-axes) of the grid cell pierced by the ray. The minimum of these distances gives

**Figure 2.** Test 2 of the RT Comparison Project (H region expansion in a uniform H gas with varying temperature). The upper (lower) panels show maps of the H fraction (temperature) cut through the simulation volume at coordinate $z = 0$ and time $t = 10$ Myr. Left: CRASH2. Right: CRASH1.
the length of the path travelled within the cell, and the boundary intersected first determines which index has to be incremented to get the coordinate of the next cell through which the ray passes. For more technical details about the algorithm, we refer the reader to Amanatides & Woo (1987).

The test cases described in Section 4 show the improvement resulting from the introduction in CRASH2 of the ray-casting calculation described above. This improvement in performance comes at the expense of computational efficiency, which nevertheless is not dramatically affected. The fraction of total computational time spent in computing ray casting strongly depends on the specifics of a given run, particularly on the physics included. In all the cases analysed here, we find it to be almost negligible, being at the most 5 per cent of the total computational time.

3.3 Tables and rates

In the new version of the code, we have updated the expressions adopted for the coefficient rates. The references for each specific coefficient can be found in table 1 of I06. We have furthermore introduced the use of pre-compiled tables for the rate coefficients with the aim of improving efficiency. The temperature dependence of the various coefficient rates is now computed once at the beginning of a given run, stored in dedicated tables and read during the run. This allows us to speed up the calculation significantly with respect to the previous implementation where the coefficient rates are functions evaluated at each cell crossing. Although a quantitative estimate of the improvement in the efficiency could be given, this is highly dependent on the problem at hand and would be significant only in specific cases. Typically though, we find that the efficiency is increased up to \( \sim 20 \) per cent. The implementation is worked out to allow for adaptive tables, by assigning as initial conditions the range of temperatures and the refinement of sampling according to the physical configuration associated with a given run. This is done once, at the beginning of each run, with the aim of adapting the range and refinement of the tables to the physical configuration of the simulation at hand: for example if the gas in the simulation has been shock heated, we assign an upper limit on the temperature range of the order of \( T_{\text{max}} \approx 10^7 \) K, which is not the case if we know that the run will process only photo-ionized gas, for which we adopt \( T_{\text{max}} \approx 10^8 \) K.

4 RESULTS AND PERFORMANCE

In the following, we will discuss some test cases run to study the performance of the new implementation described above.

4.1 Comparison Project Test 2

As a first test, we have repeated Test 2 of the RT Comparison Project (I06). This test is the classical problem of an H\( \text{II} \) region expansion in a uniform gas around a single ionizing source. A steady source with a \( 10^5 \) K blackbody spectrum, emitting \( N_{\gamma} = 5 \times 10^{48} \) ionizing photons per unit time (s\(^{-1}\)), is turned on in an initially neutral, uniform-density, static environment with hydrogen number density \( n_{\text{H}} = 10^{-3} \) cm\(^{-3}\). The computational box dimension is \( L = 6.6 \) kpc and the source is positioned at its corner. We follow the evolution of the neutral hydrogen fraction and of the gas temperature (initially set to 100 K), running the CRASH1 version with \( 10^7 \) photon packets and the CRASH2 one with \( 2 \times 10^8 \) photon packets.

The H\( \text{II} \) fraction and temperature maps, 10 Myr after the source has turned on, are shown in Fig. 2 for CRASH1 (right) and CRASH2 (left) (same cuts in figs 11 and 12 of I06). From a comparison between the left- and right-hand panels, differences are evident. More specifically, the spikes present in the CRASH1 maps, particularly in the temperature, are almost completely vanished. This is due to the coloured packets implementation which allows for a much better sampling of the high-energy tail of the spectrum, for the same number of packets emitted (and thus for the same resolution). The weak spike-like features, which are still visible in the CRASH2 temperature map, are due to the noise intrinsic to MC algorithms. Also, the artificial ring-like patterns clearly visible in the CRASH1 maps have completely disappeared, thanks to the new ray-casting algorithm implemented. Finally, a visual inspection of the maps in Fig. 2 reveals that the volume inside the I-fronts is both hotter and more highly ionized in CRASH2 than in CRASH1.

This can be better seen in the more quantitative comparison shown in Fig. 3, which plots the spherically averaged temperature (left-hand panel) and ionized and neutral fractions (right-hand panel) as a function of the distance from the source 100 Myr after the source has been turned on. CRASH2 (solid red lines) produces a higher temperature and a lower neutral fraction compared to CRASH1 (dashed blue lines). This result, which brings a better agreement between CRASH and other codes (e.g. \( c^2\)-RAY, whose results are shown

![Fig. 3](https://example.com/figure3.png)

**Figure 3.** Test 2 of the RT Comparison Project (H\( \text{II} \) region expansion in a uniform H gas with varying temperature). The left-hand (right-hand) panel shows the spherically averaged temperature (ionized and neutral H fraction) as a function of the distance from the source, 100 Myr after the source has turned on. Solid line: CRASH2. Dashed line: CRASH1. Dash–dotted line: \( c^2\)-RAY.
for comparison by the dash–dotted green curves), is once again due to the improved modelling of the filtering which is achieved thanks to a better sampling of the high energy tail of the spectrum: photons in this region of the SED can heat more and ionize less the gas with respect to lower energy photons which are oversampled in CRASH1. Despite this, as a result of the increased temperature which inhibits recombinations, the ionization fraction within the ionized sphere is higher in CRASH2. One can also see that the IF lies slightly closer to the ionizing source in the new implementation: this happens because, once the luminosity is assigned as energy emitted in the unit time, the oversampling of the lower energy part of the SED (which affects the CRASH1 implementation) results in a higher number of ionizing photons. Also, we still obtain a wider IF with respect to other codes (see fig. 17 in I06), which results from the better sampling of the spectrum in the frequency space.

4.2 Comparison Project Test 4

In this section, we describe the results from the Test 4 of the RT Comparison Project. In this case, the propagation of ionization fronts from multiple sources in a static cosmological density field is followed. The initial conditions are provided by a time-slice (at redshift $z = 9$) from a cosmological N-body and gas-dynamic simulation. The simulation box size is $0.5 h^{-1}$ comoving Mpc. The gas is assumed to be initially neutral and the temperature is initialized at 100 K everywhere in the box. The ionizing sources are chosen so as to correspond to the 16 most massive haloes in the box, which are identified with a friends-of-friends algorithm. We assume that the sources have a blackbody spectrum with effective temperature $T_{\text{eff}} = 10^5$ K. The ionizing photon production rate for each source is constant and is assigned assuming that each source lives for 3 Myr and emits 250 ionizing photons per atom during its lifetime. For simplicity, all sources are assumed to switch on at the same time. The boundary conditions are transmissive (i.e. photons leaving the computational box are lost, rather than coming back as for periodic boundary conditions). The maps in Fig. 4 show the H\textsc{i} fraction (upper panels) and temperature (lower panels) cuts through the simulation volume at $z = z_{\text{box}}/2$ and at time 0.05 Myr (same cuts as in figs 31 and 32 of I06). Both CRASH1 (right-hand panels) and CRASH2 (left-hand panels) outputs are shown in the figure. The differences between the two implementations are strikingly evident: as already found in the Test 2 above, the IFs are much larger in the CRASH2 implementation which accounts properly for
CRASH2: coloured packets and other updates

Figure 6. Comparison between the equilibrium configurations obtained by CRASH2 (points) and CLOUDY94 (solid lines) for different physical quantities, as a function of distance from the point source in cell units. The errors refer to the CLOUDY94 results and are evaluated as explained in the text.

The higher energy photons associated with larger mean free paths. The better sampling of the harder tail part of the spectrum results also in higher temperature and lower neutral hydrogen fractions as already discussed above. The new ray-casting implementation corrects furthermore for the artificial patterns visible close by the IFs in CRASH1. The difference in the internal structure of the ionized region can be better appreciated in Fig. 5 where the histograms of the temperature at time $t = 0.05$ Myr obtained with the two implementations of the code are compared: the solid (dotted) line is for CRASH2 (CRASH1). With the coloured packets implementation we find on average higher temperatures. The fact that regions at temperature above $10^3$ K are more densely populated reflects the higher temperature of ionized regions produced by the contribution of harder photon packets which are undersampled in CRASH1. At the same time, the lack of the peak found at low temperatures in CRASH1 is due to the wider IFs obtained with CRASH2, which extend much further from the sources and affect almost the whole computational volume. As in the previous test, with CRASH2 we found a better agreement with C2-RAY results shown in the figure by the dash–dotted curve.\footnote{It should be noted that in fig. 37 of I06, the purple and green lines have been erroneously exchanged.}

4.3 CRASH/ CLOUDY comparison

Finally, similarly to what was done in MFC03, we compare our code with the publicly available one-dimensional RT code CLOUDY94.\footnote{http://www.nublado.org.} With this test, we seek to check for the performance of the new algorithm in dealing with helium physics, which has not been included in the comparison project tests. We consider the case of a point source with a blackbody spectrum at $T = 6 \times 10^4$ K and luminosity $L = 10^{38}$ erg s$^{-1}$, ionizing a uniform medium with density $n = 1$ cm$^{-3}$ in a gas composed of hydrogen (90 per cent in number density) and helium.

Fig. 6 shows the profiles at equilibrium for temperature, neutral and ionization fractions. Compared to the earlier results of CRASH1 (see fig. 4 in MFC03), the new version of the code provides an overall much better agreement with CLOUDY94. In particular the neutral hydrogen fraction inner profile resulting from the new CRASH2 implementation fits much better with reduced dispersion the results from CLOUDY94. The same is true for the neutral helium fraction profile which is, however, slightly higher in CRASH2 than in CLOUDY94. Nevertheless, both the temperature and the He II profiles are slightly lower than the ones derived with CLOUDY94 and the IF extends somewhat further in CLOUDY94. The two behaviours are likely to...
be correlated and might be due to differences in the expressions adopted for the coefficient rates associated with helium together with the fact that CRASH neglects the effects of heat conduction.

5 SUMMARY

In this paper, we introduce the new version of the RT code CRASH. The details of the code implementation have been previously described in MFC03. Here, we present the new elements introduced to improve performance and efficiency. The main new ingredients implemented in CRASH2 are as follows:

(i) the introduction of coloured photon packets, which brings a significant improvement in the sampling of source spectra both in space and in frequency;
(ii) a new model for propagating the photon packets through the Cartesian grid, which includes a proper ray-casting algorithm;
(iii) the use of pre-compiled adaptive tables for the coefficient rates which result in more efficient computing.

The CRASH2 implementation has been tested and compared with the earlier versions of the code by performing a set of reference tests taken from the RT codes Comparison Project (I06) and from MFC03. CRASH2 is more accurate both in the calculation of the temperature and in the ionization fractions, thanks to the improved sampling of the hard tail part of ionizing sources spectra, which is undersampled in the monochromatic packet implementations. Furthermore, the better sampling of hard radiation together with the new ray-casting algorithm corrects for spurious geometrical features like spikes in Strömgren sphere boundaries or artificial geometrical patterns that are found in the CRASH1 results.

ACKNOWLEDGMENTS

We are grateful to Patrik Jonsson and Adrian Partl for useful discussions. We thank Marco Pierleoni for his sharp comments on the technical details of the implementation. AM is supported by the DFG Priority Program 1177.

REFERENCES

Altay G., Croft R. A. C., Pelupessy I., 2008, MNRAS, 386, 1931
Amanatides J., Woo A., 1987, Eurographics, 87, 3
Ciardi B., Ferrara A., Marri S., Raimondo G., 2001, MNRAS, 324, 381 (C01)
Iliev I. T. et al., 2006, MNRAS, 371, 1057 (I06)
Finlator K., Özel F., Davé R., 2008, preprint (arXiv:0808.3578)
Maselli A., Ferrara A., 2005, MNRAS, 364, 1429
Maselli A., Ferrara A., Ciardi B., 2003, MNRAS, 345, 379 (MFC03)
Pawlik A., Schaye J., 2008, MNRAS, 389, 651
Pierleoni M., Maselli A., Ciardi B., 2008, MNRAS, in press (arXiv:0712.1159)
Verhamme A., Schaerer D., Maselli A., 2006, A&A, 460, 397

This paper has been typeset from a TeX/LaTeX file prepared by the author.