Astronomy and Computing 23 (2018) 40–59

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom

Full length article

# The Monte Carlo photoionization and moving-mesh radiation hydrodynamics code CMacIonize

B. Vandenbroucke *, K. Wood

*SUPA, School of Physics & Astronomy, University of St Andrews, North Haugh, St Andrews, KY16 9SS, United Kingdom*

ARTICLE INFO

*Article history:*
Received 29 September 2017
Accepted 25 February 2018
Available online 8 March 2018

*Keywords:*
Numerical
Hydrodynamics
Radiative transfer
ISM: evolution

ABSTRACT

We present the public Monte Carlo photoionization and moving-mesh radiation hydrodynamics code CMacIonize, which can be used to simulate the self-consistent evolution of HII regions surrounding young O and B stars, or other sources of ionizing radiation. The code combines a Monte Carlo photoionization algorithm that uses a complex mix of hydrogen, helium and several coolants in order to self-consistently solve for the ionization and temperature balance at any given type, with a standard first order hydrodynamics scheme. The code can be run as a post-processing tool to get the line emission from an existing simulation snapshot, but can also be used to run full radiation hydrodynamical simulations. Both the radiation transfer and the hydrodynamics are implemented in a general way that is independent of the grid structure that is used to discretize the system, allowing it to be run both as a standard fixed grid code, but also as a moving-mesh code.



## 1. Introduction

Photoionization of hydrogen and helium in the interstellar medium (ISM) by luminous UV sources has an important effect on the evolution and properties of the ISM. Absorption of ionizing radiation through ionization is an important source of energy that feeds the expansion of bubbles surrounding young O and B stars, and hence shapes the structure of the ISM on small scales (Harries et al., 2017). In cases where the dynamical effect of ionizing radiation is less pronounced, the presence of ionizing radiation will still alter the overall ionization balance, not only of hydrogen and helium, but also of other elements. This, together with an increase in temperature in ionized regions, will have a visible impact on their emission spectrum, making them stand out as HII regions. Detailed observations of HII emission spectra contain a wealth of information about the local ISM and the incident radiation field, and modelling them is important in understanding observational signatures of star formation (Klassen et al., 2012; Mackey et al., 2016), and diffuse emission from galactic discs (Barnes et al., 2014; Vandenbroucke et al., 2018).

On larger scales, photoionization also has important dynamical effects. The combined UV emission of quasars and young stars in the early Universe generates a UV background radiation field that is responsible for the reionization of the Universe by redshift 6 (Becker et al., 2001; Alvarez et al., 2009). This UV background field affects the formation of galaxies by altering the abundances of ISM coolants (De Rijcke et al., 2013), and is responsible for suppressing galaxy formation in low mass haloes (Benítez-Llambay et al., 2015; Vandenbroucke et al., 2016). Furthermore, radiative feedback might be an important mechanism to regulate star formation in galactic discs (Peters et al., 2017).

Modelling photoionization and in particular HII regions requires solving a very complex system of ionization balance equations for the various elements present in these regions, which is only possible if strong assumptions are made. The widely used code Cloudy ascl:9910.001 (Ferland et al., 2017) for example assumes a simple 1D geometry, but keeps track of a large number of elements and ionization stages. When a real 3D geometry is necessary, it is no longer feasible to keep track of so many elements, and a selection has to be made, depending on the problem at hand.

When the effect of ionizing radiation on the dynamics of the ISM is studied, further assumptions need to be made about how to deal with the coupling between radiation transfer and hydrodynamics, to get a radiation hydrodynamics (RHD) scheme. Some methods treat the radiation field as a fluid governed by diffusion equations (Kolb et al., 2013; Rosdahl et al., 2013). These methods have the advantage that they do not require extensive algorithmic changes and are relatively efficient. However, they have undesired side effects, like e.g. the absence of shadows in optically thin regions, and the fact that extra assumptions need to be made about the propagation speed of the radiation field to prevent the integration time step from getting very small. Alternative methods

* Corresponding author.
*E-mail addresses:* bv7@st-andrews.ac.uk (B. Vandenbroucke), kw25@st-andrews.ac.uk (K. Wood).

https://doi.org/10.1016/j.ascom.2018.02.005
2213-1337/

use an approximate ray tracing scheme (Pawlik and Schaye, 2008; Bisbas et al., 2009; Baczynski et al., 2015) which is more complex to implement and preserves some directional information. However, these schemes require careful fine tuning to make sure the radiation field accurately covers the density structure, especially if the density structure is asymmetric or clumpy.

A more accurate, albeit less efficient way to treat the radiation field is provided by using Monte Carlo based RHD codes like TORUS ascl:1404.006 (Harries, 2000) and MOCASSIN ascl:1110.010 (Ercolano et al., 2005). These codes have the advantage that they are also much more flexible and easier to extend with extra physics, e.g. extra chemistry (Bisbas et al., 2015a). Furthermore, Monte Carlo techniques are also widely used to model dust scattering and absorption (Steinacker et al., 2013), making it straightforward to include dust scattering in Monte Carlo based RHD modelling.

In this work, we present our own Monte Carlo RHD code called CMACIONIZE, that couples a basic finite volume hydrodynamics scheme with a Monte Carlo photoionization code. Our code can use a variety of different grid types to discretize the ISM, and can run with both a fixed grid and a fully adaptive moving mesh. Apart from running as a RHD code, CMACIONIZE can also be run as a pure Monte Carlo photoionization code, and can be used to post-process density fields from other simulations.

Our code is written in modular C++ and is meant to be both user-friendly and efficient by combining a well-structured and documented design with an implementation that makes use of new features of modern C++11. The code has a limited number of dependencies and can be run in parallel using a hybrid OpenMP and MPI parallelization strategy. Some parts of the code are wrapped into a Python library using Boost Python.[1] The photoionization part of the code can also be used as an external C or Fortran library, facilitating coupling our code to other simulation codes.

This paper is structured as follows: in Section 2 we discuss the physics that has been implemented in CMACIONIZE, and give a short overview of the Monte Carlo photoionization technique and the finite volume hydrodynamics scheme. In Section 3 we describe the design considerations that were used during the development of the code, and detail their implementation. We conclude in Section 4 with the results of a number of benchmark tests that are part of the public code repository and that show its accuracy and performance.

## 2. Physics

The emission line spectrum of a star forming nebula is determined by its thermal equilibrium, which is a steady-state equilibrium between heating through photoionization by UV sources, and cooling by various atomic processes in the nebula. Osterbrock and Ferland (2006) identify four important sources of cooling:

1. Energy loss by recombination of hydrogen and helium, i.e. the reverse of the photoionization process,
2. energy loss by bremsstrahlung emitted by free electrons,
3. energy loss by collisionally excited line radiation from some abundant metals, and
4. energy loss by collisionally excited line radiation from hydrogen.

In order to compute photoionization and recombination rates, we need to know the ionization structure of the gas in the nebula, and the temperature of the nebula. Since the temperature itself is the solution of the thermal equilibrium, this can only be solved for iteratively.

The thermal and ionization equilibrium is also important for the dynamics of the nebula: ionized regions have more free particles and hence a higher specific energy than neutral regions, so that photoionization effectively acts as a heating term in the hydrodynamics of the gas. In order to properly model this effect, combined radiation hydrodynamics (RHD) simulations are necessary.

CMACIONIZE can be run in two different modes: either as a pure Monte-Carlo photoionization code that ray traces the radiation of an ionizing UV radiation field through a density field and self-consistently solves for the ionization and temperature structure, or as a radiation hydrodynamics code that uses the output of the photoionization code as a heating source in a hydrodynamical simulation. The former is essentially a completely rewritten version of the photoionization code of Wood et al. (2004), while the latter combines this code with a standard finite volume method which is a simplified version of the algorithm implemented in SHADOWFAX ascl:1605.003 (Vandenbroucke and De Rijcke, 2016). We will summarize the most important physical ingredients of both methods below.

Note that in the current version of the code, we do not include a treatment of non-ionizing radiation, nor do we take into account dust scattering and the dynamical effect of radiation pressure on dust. The treatment of these processes uses algorithms that are very similar to the ones used for photoionization, and it is straightforward to extend the code with these processes in the future.

### 2.1. Photoionization

As our initial research focusses on diffuse ionized gas in star forming nebulae, we only model photoionization of hydrogen and helium self-consistently, for UV radiation in the energy range [13.6, 54.4] eV, corresponding to the ionization threshold for hydrogen and the second ionization threshold for helium. As in Wood et al. (2004), we do not trace double ionized helium, and we only care about photons that are energetic enough to ionize hydrogen.

To model the various cooling mechanisms correctly, we also need to know the ionization structure of a number of coolants, i.e. $C^+$, $C^{++}$, $N^0$, $N^+$, $N^{++}$, $O^0$, $O^+$, $O^{++}$, $Ne^+$, $Ne^{++}$, $S^+$, $S^{++}$, and $S^{+++}$ (see 2.3). These are treated approximately, where we make the assumption that the number of free electrons released by photoionization of these elements is negligible compared to the total number of free electrons, which allows us to use a simplified ionization balance equation.

This approximation only holds in regions that are sufficiently ionized, as the total number of free electrons is mainly determined by the ionization of hydrogen and helium for realistic elemental abundances.

Note that Wood et al. (2004) did not include cooling due to $S^{+++}$, and does not mention the use of carbon cooling rates (although they were used). However, we found that not including these coolants leads to excessively high temperatures in the Lexington benchmark tests (see 4.2).

#### 2.1.1. Monte Carlo technique

The local photoionization rate depends on various factors, the most important of which are the position, direction and energy of the incoming ionizing radiation, and the local ionization state. Due to the strong non-linearity of the photoionization process, it is impossible to exactly solve for the ionization balance except for a very limited number of cases, so that approximate techniques are required.

As a first step, we discretize the density field of interest on a geometrical grid structure consisting of a (large) number of small cells. Each cell contains a compact subset of the total physical region of interest and is bounded by a discrete number of planar faces, which separate it from neighbouring cells. Our grid can be a regular Cartesian grid consisting of cubical cells, but can also be a hierarchical adaptive mesh refinement (AMR) grid (Saftly et al., 2014), or an unstructured grid (Camps et al., 2013).

[1] http://www.boost.org/doc/libs/release/libs/python/doc/html/index.html.