MNRAS **000**, 1–6 (2018) Preprint March 24, 2018 Compiled using MNRAS LATEX style file v3.0

Magritte: a new multidimensional accelerated general-purpose radiative transfer code

F. De Ceuster[1,2⋆], J. Yates[1], L. Decin[2], T.G. Bisbas[5,6], S. Viti[1] and M. Barlow[1]

[1] *Department of Physics and Astronomy, University College London, Gower Place, London, WC1E 6BT, UK*
[2] *Department of Physics and Astronomy, Institute of Astronomy, KU Leuven, Celestijnenlaan 200D, 3001 Leuven, Belgium*
[3] *Department of Astronomy and Physics, University of Florida, Gainesville, FL 32611, USA*
[4] *Max-Planck-Institut für Extraterrestrische Physik, Giessenbachstrasse 1, D-85748 Garching, Germany*

Last updated 2011 May 22; in original form 2013 September 5

ABSTRACT

Radiative transfer is a key ingredient in understanding the dynamics, chemistry and energy balance of various astrophysical objects. Therefore it is essential in astrophysical modelling to properly account for all radiative processes and their interdependence. In this paper we present MAGRITTE: a new multidimensional accelerated general-purpose radiative transfer code. MAGRITTE is a grid-independent deterministic ray-tracing code that obtains the radiation field by solving the transfer equation along a fixed set of rays originating from each grid cell. It iteratively accounts for scattering and treats the full frequency space. Appended with a dedicated chemistry and thermal balance module MAGRITTE can self-consitently calculate the temperature field, chemical abundances and level populations.

MAGRITTE is especially designed to have a well scaling performance on various computer architectures. in stellar winds and perform synthetic observations. We will apply MARGRITTE by post-processing snapshots of hydrodynamical wind and bow-shock simulations and will present its integration in self-consistent hydro-chemical AGB wind models.

Key words: radiative transfer, astrochemistry, methods: numerical

1 INTRODUCTION

Radiative transfer processes play a central role in the dynamics, the chemistry and the energy balance of all kinds of astrophysical objects. It provides a radiative pressure that can drive the dynamics, it affects the chemistry through various photoionization and photodissociation reactions, and it can very efficiently heat or cool very specific regions. Furthermore the radiative transfer determines how these objects appear in observations: which regions are visible in what part of the spectrum. Therefore it is essential in astrophysical modelling to properly account for all radiative processes and their interdependence. This however can be highly complicated by i) an intricate 3D geometrical structure shielding or exposing specific regions to radiation, ii) the scattering of radiation by dust or free electrons yielding additional non-trivial coupling between the geometry and the radiation field, and iii) the mixing in frequency space due to Doppler shifts caused by velocity gradients in the medium. The tight coupling between the radiative processes and the often very specialized but diverse dynamical and chemical models furthermore requires a modular solution method that can easily be integrated into various frameworks.

There are two main computational strategies used to solve radiative transfer problems. There are the probabilistic (Monte Carlo) codes e.g. (Baes et al. 2011; Dullemond et al. 2012; Vandenbroucke & Wood 2018) and deterministic ray-tracing codes e.g. (Altay et al. 2011; Bisbas et al. 2012). Nowadays most radiative transfer codes use probabilistic methods. They mimic the physical photon transport by propagating a number of photon packets through the medium. The main issue with this approach is that photon packets can get trapped in optically thick regions of the medium which can severely increase the computation time. Although many techniques have been devised to avoid the trapping of photon packets it remains a serious bottleneck for probabilistic codes. Deterministic ray-tracing codes do not suffer from photon trapping. They obtain the radiation field by (iteratively) solving the transfer equation along a number of fixed rays. The fixed (deterministic) computational scheme moreover leads to various opportunities for optimization (De Ceuster et al. *in prep.*).

⋆ Contact e-mail: frederik.deceuster@kuleuven.be

© 2018 The Authors

In this paper we present MAGRITTE: a new multidimensional accelerated general-purpose radiative transfer code. MAGRITTE is a deterministic ray-tracing code that is independent of the underlying grid structure. It can trace its rays only using the positions of the cell centers and the nearest neighbours lists. MAGRITTE can iteratively account for anisotropic scattering by a range of dust species and free electrons. The methods are also flexible enough to be extended to scattering from aligned dust grains due to e.g. magnetic fields. The code treats the full frequency space including continuum radiation, and atomic and molecular lines. Appended with a dedicated chemistry and thermal balance module it can self-consitently calculate the temperature field, chemical abundances and level populations.

The development of the code started from the (open source) 3D astrochemistry code 3D-PDR[1] by Bisbas et al. (2012). From there it was rewritten and restructured in a more object-oriented way in C++. Starting from an existing code allowed us to heavily test and benchmark the new structure used in MAGRITTE. 3D-PDR was made to self-consitently model the chemistry, some atomic and molecular lines and the thermal balance in phododissociation regions (PDRs). In 3D-PDR the UV radiation field is obtained by direct integration, neglecting the diffusive radiation using the on-the-spot approximation (Osterbrock 1974). When only considering external UV sources, this approximation is simple enough to allow for a self-consistent treatment of the UV radiation field and the UV photochemistry. The atomic and molecular line tranfer in 3D-PDR is treated in the escape probability formalism using the large velocity gradient (LVG) or Sobolev approximation (Sobolev 1960; Castor 1970; de Jong et al. 1975; Poelman & Spaans 2005). Using this approximation the line transfer can be used to calculate the radiative cooling fast enough to self-consitently calculate the temperature field by assuming thermal balance. Although the above assumptions are very useful and reasonable in some situations, they highly limit the applicability of the code. Another limitation to 3D-PDR is the high memory cost due to the way in which the rays are stored. This limits the computational size of the models that can be run.

As a first step in the development of MAGRITTE the old ray-tracer was rewritten to make use of the neighbours lists of the grid points. This improved its speed which allowed us to trace the rays on-the-fly thus avoiding the need to store them. This drastically reduces the memory cost. Afterwards the atomic and molecular line radiative transfer was replaced by an approximated lambda iteration (ALI) scheme. The new scheme does not make any assumptions on the velocity field and was extened to account for dust scattering. Finally the thermal balance module that self-consistently calculates the temperature field was made more robust. Thanks to the new grid-independent structure of MAGRITTE we have a lot of flexibility in the way we use the grid. We can start the temperature calculations on a very coarse grid and use a slow but careful method to get an idea of the temperature field. Then we can interpolate this onto a more refined grid and recalculate the temperature using a faster method. Since we already had a reasonable idea of the temperature to start with, the faster method will still be stable. The flexibility to make the optimal trade-off between stability and performance is the guiding idea in the object-oriented design and structure of MAGRITTE.

In this paper we mainly focus on the new physics that is implemented in MAGRITTE and benchmark the accuracy of its output. In a forthcomming paper (De Ceuster et al. *in prep.*) we will investigate further optimization and parallelization techniques that can be used to obtain optimal and scalable performance of MAGRITTE on various computer architectures. Afterwards we plan to demonstrate the versatillity of the code by post-processing snapshots of hydrodynamical simulations of both stellar winds and composite galaxies. To demonstrate the modularity we plan to integrate it in AMRVAC (Xia et al. 2018) to be used together with KROME (Grassi et al. 2014) in self-consistent radiative hydro-chemical AGB wind models.

This paper is organized as follows. In section 2 we present the new computational scheme of MAGRITTE. Section 3 describes the different benchmarks that were done to test the different modules and to compare its performance against 3D-PDR. In Section 4 we present some first novel applications. Finally our conclusions are discussed in Section 5.

2 COMPUTATIONAL SCHEME

Although MAGRITTE is based on 3D-PDR (Fortran), the whole code has been rewritten and restructured in C++. As input, MAGRITTE takes an unstructured grid. For each grid cell one needs to specify the cell center, the total density and the local gas velocity. If known, one can also specify the nearest neighbours for each cell, otherwise these are calculated by MAGRITTE. Except for the neighbours no other attributes of the grid (e.g. cell edges or faces) are needed. The code does not dependent on the underlying structure (i.e. the tessellation) of the grid. This enables us to easily handle various types of input, from AMR-grids to Voronoi grids as well as SPH-particle data[2].

2.1 Ray-tracing

MAGRITTE is a deterministic ray-tracing code. The radiation field is determined by solving the radiative transfer equation on a fixed set of rays (i.e. straight lines) originating from each cell center. The mean intensity in a cell is then obtained by averaging over the radiation along the different rays. The direction of a ray is determined by the HEALPIX[3] discretization of the sphere (Górski et al. 2005). Given a level of refinement ℓ, it discretizes the unit sphere in $N_{\text{rays}} = 12 \times 4^{\ell}$ uniformly distributed pixels of equal area. For each pixel there is an associated unit vector pointing from the origin of the unit sphere to the pixel center. Our rays are thus defined as the straight lines originating from the cell centers in the directions of these unit vectors.

[1] github.com/uclchem/3D-PDR

[2] We are aware that specifying a certain type of grid (e.g. Voronoi) can further improve the performance in terms of speed. In future versions, when MAGRITTE is used in combination with certain other codes, we will optimize for certain grids.

[3] healpix.sourceforge.net

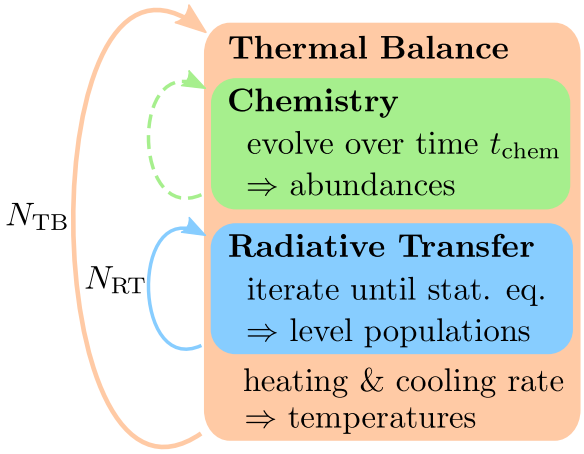

Figure 1. Iteration schemes of the Magritte workflow, different boxes represent the different modules.

2.1.1 *Constructing the rays*

In order to solve the transfer equation along a certain ray, we need to know the emissivity and opacity of the cells that are encountered along that ray. Furthermore we need to know the path length of a ray through a certain cell. This is slightly complicated since the only geometrical information we required from the cells is the location of their cell center and their nearest neighbours. The solution as implemented in Magritte is to walk along the ray and project nearest cell centers onto the ray. Consider a ray originating from a cell, say o (see figure 2). Clearly the cell o itself should lie on the ray. The next cell to be projected on the ray is the neighbour of o that lies closest to the ray. We will call this cell p_1. Now the next cell to be projected is the neighbour of p_1 that lies closest to the ray and that is further away from o than p_1. The last condition is there to ensure that one proceeds along the ray towards the boundary. This process is repeated untill the boundary of the gird is reached.

The above algorithm constructs the ray from a given cell outward to the boundary. However it is also useful to be able to do the opposite and construct a ray from the boundary inward towards a certain cell. We will refer to this cell for the moment as the origin. To do this one needs to know the end point of each ray, i.e. the final cell of the boundary that will be projected on the ray. These are calculated and stored in the setup of Magritte. Once the endpoints are known the above algorithm can be used to walk along the ray from the boundary to the origin of the ray. The only difference is that one should pick the next cell closest to the ray and that is closer to the origin instead off further away. The last condition is again there to ensure that one recedes along the ray back to the origin.

2.1.2 *Ray-tracing in* Magritte *vs.* 3d-pdr

In 3D-PDR the projections of all cells on the rays for each cell are calculated during setup (using a different algorithm) and kept in memory. This is a huge amount amount of data to be stored which scales as $N_{\rm cells} \times N_{\rm cells} \times N_{\rm rays}$. This quickly leads to memory issues, severely limiting the grid size of the simulations that can be run. In Magritte we only store the projections of the cells on the rays for one cell at the time.

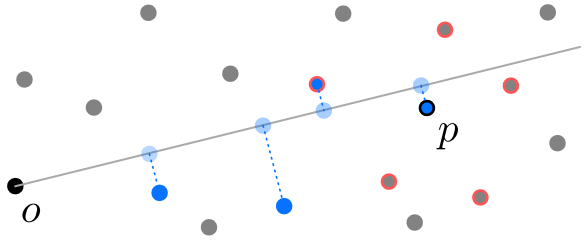

Figure 2. Sketch of the algorithm that chooses which cells to project on a ray. All cells are represented by their cell centers. The ray originates from the black cell (denoted o). The blue cells are already projected onto the ray. The blue cell with the black border (denoted p) is the last cell that was projected. The neighbours of p are indicated with red borders. The next cell to be projected is the neighbor of p with a projection on the ray further than the projection of p, that lies closest to the ray.

This reduces the scaling of the memory cost to $N_{\rm cells} \times N_{\rm rays}$ which allows us to run simulations with much larger grids on the same machine.

2.2 Chemistry

The chemistry module is largly based on the chemical gas-grain code UCLCHEM (Holdship et al. 2017). We opted to rewrite the needed parts of the code in our framework rather than using it to facilitate future extentions. In this module Magritte can determine the relative abundances of a limited number of atomic and molecular species at each cell. This is done by solving the time-dependent chemistry of a self-contained network of formation and destruction reactions. The chemical network is a subset of the most recent UMIST data base of reaction rates (Woodall et al. 2007), consisting of 320 reactions between 33 species (including electrons), and includes photoionization and photodissociation reactions in addition to the standard gas-phase chemistry.

2.3 Level populations

Given the chemical abundances, Magritte can calculate the level populations for a subsset of the chemical species. The evolution of the population n_i of the level i of a species with N levels in the comoving frame is given by the transition rate to that level minus the transition rate form that level

$$\frac{\partial n_i(\boldsymbol{x})}{\partial t} = \sum_{j=1}^{N} n_j(\boldsymbol{x}) R_{ji}(\boldsymbol{x}) - n_i(\boldsymbol{x}) \sum_{j=1}^{N} R_{ij}(\boldsymbol{x}). \tag{1}$$

The transition rates in terms of Einstein coefficients read

$$R_{ij}(\boldsymbol{x}) = A_{ij} + 4\pi B_{ij} J_{ij}(\boldsymbol{x}) + C_{ij}(\boldsymbol{x}), \tag{2}$$

which account for the spontanious (A_{ij}), stimulated (B_{ij}) and collisional ($C_{ij}(\boldsymbol{x})$) transitions. Note that $A_{ij} = 0$ for $i \leq j$ and that the collisional Einstein coefficients have a position dependence because they depend on the local gas temperature. In our simulations we used the line data from the Leiden Atomic and Molecular Database (LAMDA, Schöier et al. 2005). $J_{ij}(\boldsymbol{x})$ is the local mean intensity in the spectral range of the transition $i \leftrightarrow j$. In general the intensity itself depends on the level populations. Therefore one needs

to solve iteratively for the level populations, each time using the values of the previous iteration to compute the mean intensity. In MAGRITTE this is done assuming local statistical equilibrium, i.e. $\partial n_i(\boldsymbol{x})/\partial t = 0$.

2.3.1 Radiative transfer

The local mean intensity $J_{ij}(\boldsymbol{x})$ in the spectral range of a transition $i \leftrightarrow j$ is the monochromatic specific intensity $I_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}})$ averaged over all directions and intergrated over the whole spectrum weighted by the local line profile $\phi_\nu^{ij}(\boldsymbol{x})$,

$$J_{ij}(\boldsymbol{x}) = \oint \frac{\mathrm{d}\Omega}{4\pi} \int_0^\infty \mathrm{d}\nu\ \phi_\nu^{ij}(\boldsymbol{x})\ I_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}}). \tag{3}$$

To obtain the average over all directions, MAGRITTE computes the intensity along all rays and averages over the rays. To obtain the weighted integral over the spectrum MAGRITTE computes the intensity for certain frequencies given by a quadrature scheme.

In the simulations presented in this paper we always assumed Gaussian profile functions[4] such that the integrals over the spectrum can be done using a Gauss-Hermitte quadrature. A Gaussian profile function in the comoving frame reads,

$$\phi_\nu^{ij}(\boldsymbol{x}) = \frac{1}{\sqrt{\pi}\delta\nu_{ij}} \exp\left(-\left(\frac{\nu - \nu_{ij}}{\delta\nu_{ij}}\right)^2\right), \tag{4}$$

where the width $\delta\nu$ of the profile is due to the Doppler shifts caused by motions of the atoms and molecules in a cell,

$$\left(\frac{\delta\nu_{ij}}{\nu_{ij}}\right)^2 = \frac{2k_\mathrm{B}T(\boldsymbol{x})}{\pi mc^2} + \left(\frac{v_\mathrm{turb.}}{c}\right)^2. \tag{5}$$

The first term is due to thermal motions (along the line of sight) and the second is due to the turbulent motion.

To find in each direction $(\hat{\boldsymbol{n}})$ the monochromatic specific intensity, MAGRITTE solves the radiative transfer equation in the comoving frame along each ray.

$$\hat{\boldsymbol{n}} \cdot \nabla I_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}}) = \eta_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}}) - \chi_\nu(\boldsymbol{x})\ I_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}}). \tag{6}$$

The emissivity $\eta_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}})$ and opacity $\chi_\nu(\boldsymbol{x})$ can be split into a line, a continuum and a scattering term,

$$\begin{aligned} \eta_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}}) &= \eta_\nu^{ij}(\boldsymbol{x}) + \eta_\nu^{\mathrm{con.}}(\boldsymbol{x}) + \eta_\nu^{\mathrm{sca.}}(\boldsymbol{x}, \hat{\boldsymbol{n}}) \\ \chi_\nu(\boldsymbol{x}) &= \chi_\nu^{ij}(\boldsymbol{x}) + \chi_\nu^{\mathrm{con.}}(\boldsymbol{x}) + \chi_\nu^{\mathrm{sca.}}(\boldsymbol{x}). \end{aligned} \tag{7}$$

The line emissivity and opacity are caused by the transitions between the level populations and can be expressed in terms of the Einstein coefficients,

$$\begin{aligned} \eta_\nu^{ij}(\boldsymbol{x}) &= \frac{h\nu}{4\pi} A_{ij}\ n_i(\boldsymbol{x})\ \phi_\nu^{ij}(\boldsymbol{x}), \\ \chi_\nu^{ij}(\boldsymbol{x}) &= \frac{h\nu}{4\pi} \left(B_{ji}\ n_j(\boldsymbol{x}) - B_{ij}\ n_i(\boldsymbol{x})\right) \phi_\nu^{ij}(\boldsymbol{x}). \end{aligned} \tag{8}$$

The continuum terms are due to dust. MAGRITTE contains one dust species with a certain dust temperature $T_\mathrm{dust}(\boldsymbol{x})$. The cosmic microwave background (CMB) is treated as a boundary condition.

[4] MAGRITTE can cope with any user defined profile function as long as the appropriate roots and weights for the quadrature are also provided by the user.

MAGRITTE solves the monochromatic transfer equation in Feautrier form (Feautrier 1964) using the improved numerical scheme suggested by Rybicki & Hummer (1991). In this method the transfer equation is rewritten as a second-order differential equation

$$u_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}}) = \frac{1}{2}\Big(I_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}}) + I_\nu(\boldsymbol{x}, -\hat{\boldsymbol{n}})\Big) \tag{9}$$

$$\frac{\mathrm{d}^2 u_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}})}{\mathrm{d}\tau_\nu^2} = u_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}}) - S_\nu^{\mathrm{eff.}}(\boldsymbol{x}, \hat{\boldsymbol{n}}) \tag{10}$$

In 3D-PDR the radiative transfer is solved in the escape probability formalism and applying the large velocity gradient (LVG) or Sobolev approximation (Sobolev 1960; Castor 1970; de Jong et al. 1975; Poelman & Spaans 2005). This is still an option in Magritte. The mean intensity is given by

2.3.2 Accelerated Lambda Iterations (ALI)

The convergence of the level populations can be notoriously slow. In order to account for this, many acceleration methods have been devised over the years. In MAGRITTE we used an approximated lambda operator scheme introduced by Rybicki & Hummer (1991), complemented by the Ng-acceleration method (Ng 1974).

The approximated lambda operator scheme of Rybicki & Hummer reoders the transition coefficients as

$$R_{ij}(\boldsymbol{x}) = A_{ij}\left(1 - \Lambda_{ij}^*(\boldsymbol{x})\right) + B_{ij} J_{ij}^{\mathrm{eff.}}(\boldsymbol{x}) + C_{ij}(\boldsymbol{x}), \tag{11}$$

where the approximated lambda operator $\Lambda_{ij}^*(\boldsymbol{x})$ and the resulting effective mean intensity $J_{ij}^{\mathrm{eff.}}(\boldsymbol{x})$ are given by

$$\begin{aligned} \Lambda_{ij}^*(\boldsymbol{x}) &= \\ J_{ij}^{\mathrm{eff.}}(\boldsymbol{x}) &= \end{aligned} \tag{12}$$

In the Ng-acceleration method (Ng 1974) a new educated guess for the level populaions is made based on the level populations in previous iterations.

The combination of both acceleration schemes yields a significant speed-up in the convergence of the level populations. In the Sobolev case the number of iterations over the level populations is reduced to $\mathcal{N}_\mathrm{pop} \approx 3$, which is an improvment by a factor 10 (?), see Appendix A. Note that the Ng-acceleration is redundant in this case since there are not enough iterations. Since this is such a simple way to obtain a such a drastic improvment in performance it was also implemented in a new update to 3D-PDR[5].

In the general case theIn the general case the

2.4 Thermal balance

MAGRITTE can iteratively determine the gas temperature, consistent with the chemical and radiative state, by demanding local thermal balance, i.e. equal heating and cooling rates in each cell. This assumes that the dynamical timescale of the system is large enough to be negligible. This is for instance the case in simulations of photodissociation regions (PDRs), see (citations!).

[5] github.com/uclchem/3D-PDR

The temperature is iteratively calculated using the following scheme: Assuming a certain temperature field, the chemical abundances and level populations are calculated using their respective modules. These can be used to calculate the heating and cooling rates in each cell. Then a new guess can be made

The modular design of the code and the relatively simple structure of our cells allows us to do the

3 BENCHMARKS

3.1 Ray-tracing

Angular resolution tests

3.2 Line radiative transfer

In order to test MAGRITTE's new radiative transfer module, it was compared against the results of the van Zadelhoff et al. (2002) benchmark.

3.3 Dust scattering

3.4 MAGRITTE vs. 3D-PDR

3.5 MAGRITTE vs. LIME

In order to test MAGRITTE's new radiative transfer module, it was benchmarked against LIME (Brinch & Hogerheijde 2010).

4 APPLICATIONS

The modular character of Magritte allows it to be easily used in various astrophysical simulations.

5 CONCLUSIONS

We have presented Magritte: a new multidimensional accelerated general-purpose radiative transfer code.

The source code for Magritte and its separate modules will eventually be made freely available on github.com/Magritte-code. In the mean time various parts can be made available upon request.

ACKNOWLEDGEMENTS

The authors would like to thank C.P. Dullemond and C. Brinch for helpful and encouraging discussions.

FDC is supported by the EPSRC iCASE studentship programme, Intel Corporation and Cray Inc.

References

Altay G., Croft R. A. C., Pelupessy I., 2011, SPHRAY: A Smoothed Particle Hydrodynamics Ray Tracer for Radiative Transfer, Astrophysics Source Code Library (ascl:1103.009)
Baes M., Dejonghe H., Davies J., 2011, SKIRT: Stellar Kinematics Including Radiative Transfer, Astrophysics Source Code Library (ascl:1109.003)
Bisbas T. G., Bell T. A., Viti S., Yates J., Barlow M. J., 2012, Mon. Not. R. Astron. Soc., 427, 2100
Brinch C., Hogerheijde M. R., 2010, A&A, 523, A25
Castor J. I., 1970, MNRAS, 149, 111
Chandrasekhar S., 1960, Radiative transfer
Dullemond C. P., Juhasz A., Pohl A., Sereshti F., Shetty R., Peters T., Commercon B., Flock M., 2012, RADMC-3D: A multi-purpose radiative transfer tool, Astrophysics Source Code Library (ascl:1202.015)
Feautrier P., 1964, Comptes Rendus Academie des Sciences (serie non specifiee), 258
Górski K. M., Hivon E., Banday A. J., Wandelt B. D., Hansen F. K., Reinecke M., Bartelmann M., 2005, ApJ, 622, 759
Grassi T., Bovino S., Schleicher D. R. G., Prieto J., Seifried D., Simoncini E., Gianturco F. A., 2014, MNRAS, 439, 2386
Holdship J., Viti S., JimÃl'nez-Serra I., Makrymallis A., Priestley F., 2017, The Astronomical Journal, 154, 38
Ng K.-C., 1974, J. Chem. Phys., 61, 2680
Osterbrock D. E., 1974, Astrophysics of Gaseous Nebulae
Poelman D. R., Spaans M., 2005, A&A, 440, 559
Rybicki G. B., Hummer D. G., 1991, Astron. Astrophys., 245, 171
Schöier F. L., van der Tak F. F. S., van Dishoeck E. F., Black J. H., 2005, A&A, 432, 369
Sobolev V. V., 1960, Moving envelopes of stars
Steinacker J., Baes M., Gordon K. D., 2013, ARA&A, 51, 63
Vandenbroucke B., Wood K., 2018, CMacIonize: Monte Carlo photoionisation and moving-mesh radiation hydrodynamics, Astrophysics Source Code Library (arXiv:1802.09528)
Woodall J., Agúndez M., Markwick-Kemper A. J., Millar T. J., 2007, A&A, 466, 1197
Xia C., Teunissen J., El Mellah I., Chané E., Keppens R., 2018, ApJS, 234, 30
de Jong T., Dalgarno A., Chu S.-I., 1975, ApJ, 199, 69
van Zadelhoff G.-J., et al., 2002, A&A, 395, 373

APPENDIX A: ALI IN THE ESCAPE PROBABILITY FORMALISM

In 3D-PDR the radiative transfer is solved in the escape probability formalism and applying the large velocity gradient (LVG) or Sobolev approximation (Sobolev 1960; Castor 1970; de Jong et al. 1975; Poelman & Spaans 2005). In this approximate formalism the mean intensity in the spectral range of a transition $i \leftrightarrow j$ is given by

$$J_{ij}(\boldsymbol{x}) = \Big(1 - \beta_{ij}(\boldsymbol{x})\Big)\, S_{ij}^{\text{line}}(\boldsymbol{x}) + \beta_{ij}(\boldsymbol{x}) J_{ij}^{\text{con.}}(\boldsymbol{x}) \tag{A1}$$

where the escape probability $\beta_{ij}(\boldsymbol{x})$ is cumputed as,

$$\beta_{ij}(\boldsymbol{x}) = \oint \frac{\mathrm{d}\Omega}{4\pi} \frac{1 - e^{-\tau_{ij}(\boldsymbol{x},\hat{\boldsymbol{n}})}}{\tau_{ij}(\boldsymbol{x},\hat{\boldsymbol{n}})} \tag{A2}$$

in terms of the line optical depth

$$\tau_{ij}(\boldsymbol{x},\hat{\boldsymbol{n}}) = \int_0^\infty \mathrm{d}s\ \chi_{ij}(\boldsymbol{x} + s\hat{\boldsymbol{n}}) \tag{A3}$$

The approximated lambda operator and associated effective mean intensity, as defiined by Rybicki & Hummer (1991) is

in this case simply given by

$$\begin{aligned} \Lambda^*_{ij}(\boldsymbol{x}) &= 1 - \beta_{ij}(\boldsymbol{x}) \\ J^{\text{eff.}}_{ij}(\boldsymbol{x}) &= \beta_{ij}(\boldsymbol{x})\ J^{\text{con.}}_{ij}(\boldsymbol{x}). \end{aligned} \tag{A4}$$

Substituted in equation (11) this yiels

$$R_{ij}(\boldsymbol{x}) = \beta_{ij}(\boldsymbol{x}) \left(A_{ij} + B_{ij} J^{\text{con.}}_{ij}(\boldsymbol{x})\right) + C_{ij}(\boldsymbol{x}), \tag{A5}$$

This simple reorderig of terms in the transition rates yields a significant improvment in the convergence of the level populations. The number of iterations is reduced to $\mathcal{N}_{\text{pop}} \approx 3$, which is an improvment by a factor 10 (?). Since this simple change leads to such a major performance improvment it will also be included in 3D-PDR.

APPENDIX B: SCATTERING IN THE FEAUTRIER FORMALISM

The radiative transfer equation relates the change in specific monochromatic intensity $I_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}})$ along a ray in direction $\hat{\boldsymbol{n}}$ to the local emissivity $\eta_\nu(\boldsymbol{x})$ and opacity $\chi_\nu(\boldsymbol{x})$. For local radiative processes one can assume that both the emissivity $\eta_\nu(\boldsymbol{x})$ and the opacity $\chi_\nu(\boldsymbol{x})$ are isotropic, i.e. independent of the direction $\hat{\boldsymbol{n}}$. When considering scattering (see e.g. Chandrasekhar 1960; Steinacker et al. 2013) there is an extra direction-dependent emmisivity term in the transfer equation

$$\begin{aligned} \hat{\boldsymbol{n}} \cdot \nabla I_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}}) &= \eta_\nu(\boldsymbol{x}) - \chi_\nu(\boldsymbol{x})\ I_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}}) \\ &\quad + \oint \mathrm{d}\Omega\ \Psi_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}}, \hat{\boldsymbol{n}}')\ I_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}}'). \end{aligned} \tag{B1}$$

where $\Psi(\boldsymbol{x}, \hat{\boldsymbol{n}}, \hat{\boldsymbol{n}}')$ is the scattering redistribution function which gives the probability of an incoming photon along direction $\hat{\boldsymbol{n}}'$ to be absorbed and then scattered in direction $\hat{\boldsymbol{n}}$. To solve the transfer equation we can proceed, as suggested by Feautrier (1964), by defining

$$\begin{aligned} u_\nu(\hat{\boldsymbol{n}}) &= \frac{1}{2}\left(I_\nu(\hat{\boldsymbol{n}}) + I_\nu(-\hat{\boldsymbol{n}})\right), \\ v_\nu(\hat{\boldsymbol{n}}) &= \frac{1}{2}\left(I_\nu(\hat{\boldsymbol{n}}) - I_\nu(-\hat{\boldsymbol{n}})\right). \end{aligned} \tag{B2}$$

Adding and subtracting the transfer equation (B1) for $\hat{\boldsymbol{n}}$ and $-\hat{\boldsymbol{n}}$ yields a coupled set of equation for $u_\nu(\hat{\boldsymbol{n}})$ and $v_\nu(\hat{\boldsymbol{n}})$.

$$\begin{aligned} \hat{\boldsymbol{n}} \cdot \nabla u_\nu(\hat{\boldsymbol{n}}) &= \frac{1}{2}\left(\eta_\nu(\hat{\boldsymbol{n}}) - \eta_\nu(-\hat{\boldsymbol{n}})\right) - \chi_\nu\ v_\nu(\hat{\boldsymbol{n}}) \\ \hat{\boldsymbol{n}} \cdot \nabla v_\nu(\hat{\boldsymbol{n}}) &= \frac{1}{2}\left(\eta_\nu(\hat{\boldsymbol{n}}) + \eta_\nu(-\hat{\boldsymbol{n}})\right) - \chi_\nu\ u_\nu(\hat{\boldsymbol{n}}) \end{aligned} \tag{B3}$$

Defining the monochromatic optical depth as usual

$$\mathrm{d}\tau_\nu = \chi_\nu\ \hat{\boldsymbol{n}} \cdot \mathrm{d}\boldsymbol{x}, \tag{B4}$$

equation (B3) can be solved for $u_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}})$.

$$\begin{aligned} \frac{\mathrm{d}^2 u_\nu(\hat{\boldsymbol{n}})}{\mathrm{d}\tau_\nu^2} &= u_\nu(\hat{\boldsymbol{n}}) - \frac{\eta^{\text{eff.}}_\nu(\hat{\boldsymbol{n}}) + \eta^{\text{eff.}}_\nu(-\hat{\boldsymbol{n}})}{2\chi^{\text{eff.}}_\nu} \\ &\quad + \frac{\mathrm{d}}{\mathrm{d}\tau_\nu}\left(\frac{\eta^{\text{eff.}}_\nu(\hat{\boldsymbol{n}}) - \eta^{\text{eff.}}_\nu(-\hat{\boldsymbol{n}})}{2\chi^{\text{eff.}}_\nu}\right) \end{aligned} \tag{B5}$$

Substituting the effective emissivities obtained in equation (B9) gives the Feautrier equation including scattering,

$$\frac{\mathrm{d}^2 u_\nu(\hat{\boldsymbol{n}})}{\mathrm{d}\tau_\nu^2} = u_\nu(\hat{\boldsymbol{n}}) - S^{\text{eff.}}_\nu(\hat{\boldsymbol{n}}), \tag{B6}$$

where the effective source function is defined as

$$\begin{aligned} S^{\text{eff.}}_\nu(\hat{\boldsymbol{n}}) &= \frac{\eta_\nu}{\chi^{\text{eff.}}_\nu} - \frac{\chi^{\text{sca.}}_\nu}{\chi^{\text{eff.}}_\nu} \oint \mathrm{d}\Omega\ \Phi(\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{n}}')\ u_\nu(\hat{\boldsymbol{n}}') \\ &\quad + \frac{\mathrm{d}}{\mathrm{d}\tau_\nu}\left(\frac{\chi^{\text{sca.}}_\nu}{\chi^{\text{eff.}}_\nu} \oint \mathrm{d}\Omega\ \Phi(\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{n}}')\ v_\nu(\hat{\boldsymbol{n}}')\right). \end{aligned} \tag{B7}$$

Solving the full Feautrier equation with scattering is intractable because of the $u_\nu(\hat{\boldsymbol{n}})$ and $v_\nu(\hat{\boldsymbol{n}})$ dependence in $S^{\text{eff.}}_\nu(\hat{\boldsymbol{n}})$. However, since we already have to solve the transfer iteratively because of the level populations, we can replace the effective source function with the one calculated in the previous iteration.

When discretizing the integrals over the directions in the transfer equation it is clearely beneficial to move the forward scattering term from the emissivity to the opacity,

$$\begin{aligned} \eta^{\text{eff.}}_\nu(\hat{\boldsymbol{n}}) &= \eta_\nu + \eta^{\text{sca.}}_\nu(\hat{\boldsymbol{n}}) \\ \chi^{\text{eff.}}_\nu &= \chi_\nu + \left(1 - \Phi(\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{n}})\right)\chi^{\text{sca.}}_\nu \end{aligned} \tag{B8}$$

such that now the emissivity reads

$$\eta^{\text{sca.}}_\nu(\hat{\boldsymbol{n}}) = \chi^{\text{sca.}}_\nu \sum_{\hat{\boldsymbol{n}}' \neq \hat{\boldsymbol{n}}} \Phi(\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{n}}')\ I_\nu(\hat{\boldsymbol{n}}'). \tag{B9}$$

Since the scattering phase function is strongly peaked around $\hat{\boldsymbol{n}}' = \hat{\boldsymbol{n}}$ this reduces the contributions of the scattering terms in the effective source function. This will improve the convergence of the iterations over the level populations.

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