MNRAS **000**, 1–6 (2018) Preprint March 10, 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] and S. Viti[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 plays a key role in the dynamics, the chemistry and the energy balance of all kinds of astrophysical objects. It provides the radiative pressure to drive stellar winds, it affects the chemistry through various photoionization and photodissociation reactions, and it can very efficiently heat or cool very specific regions in the wind. Therefore it is essential in modelling these objects to properly account for all radiative processes and their interdependence. This however can be complicated by i) an intricate 3D geometrical structure shielding or exposing specific regions to radiation, ii) the scattering by dust or free electrons, and iii) the mixing in frequency space due to Doppler shifts caused by velocity gradients in the medium. The tight coupling between radiative transfer and the often very specialized and diverse dynamical and chemical models furthermore requires a modular code that can easily be integrated. To address these needs we will present MAGRITTE: a new multidimensional accelerated general-purpose radiative transfer code. MAGRITTE is especially designed to have a well scaling performance on various computer architectures. Appended with a dedicated chemistry and thermal balance module it can self-consitently calculate the temperature field, chemical abundances and level populations. Magritte is a 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. This allows us to self-consistently model the chemistry and energy balance in stellar winds and perform synthetic observations. We will apply MARGRITTE by post-processing snapshots of hydrodynamical wind and bowshock 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 plays a key role in the dynamics, the chemistry and the energy balance of all kinds of astrophysical objects. It provides the radiative pressure to drive stellar winds, it affects the chemistry through various photoionization and photodissociation reactions, and it can very efficiently heat or cool very specific regions of a gas. Therefore it is essential in modelling these objects to properly account for all radiative processes and their interdependence. This however can be complicated by i) an intricate 3D geometrical structure shielding or exposing specific regions to radiation, ii) the scattering by dust or free electrons, and iii) the mixing in frequency space due to Doppler shifts caused by velocity gradients in the medium. The tight coupling between radiative transfer and the often very specialized and diverse dynamical and chemical models furthermore requires a modular code that can easily be integrated into various frameworks. To address these needs we will present MAGRITTE: a new multidimensional accelerated general-purpose radiative transfer code. MAGRITTE is especially designed to have a well scaling performance on various computer architectures. Appended with a dedicated chemistry and thermal balance module it can self-consitently calculate the temperature field, chemical abundances and level populations. Magritte is a deterministic ray-tracing code that obtains the

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

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. This allows us to self-consistently model the chemistry and energy balance in stellar winds and perform synthetic observations. We will apply MARGRITTE by post-processing snapshots of hydrodynamical wind and bowshock simulations and will present its integration in self-consistent hydrochemical AGB wind models.

Post-processing and synthetic observations (Haworth et al. 2017).

New C++ code based on 3D-PDR (Bisbas et al. 2012) improved performance. separate modules for chemistry, radiative transfer and thermal balance.

In this paper we will mainly focus on the physics that is implemented in MAGRITTE and how it can be used in a generic computational setup. In a forthcomming paper (De Ceuster et. al. *in prep.*) we will investigate further optimization techniques that can be used to obtain optimal performance on various computing architectures.

The paper is organized as follows. In section 2 we present the new computational scheme and the different modules of Magritte. Section 3 describes the different benchmarks that were done to test the code and to compare its performance with 3D-PDR. In Section 4 we present the new applications that have become feasible through the nove 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 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 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[1].

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[2] 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.

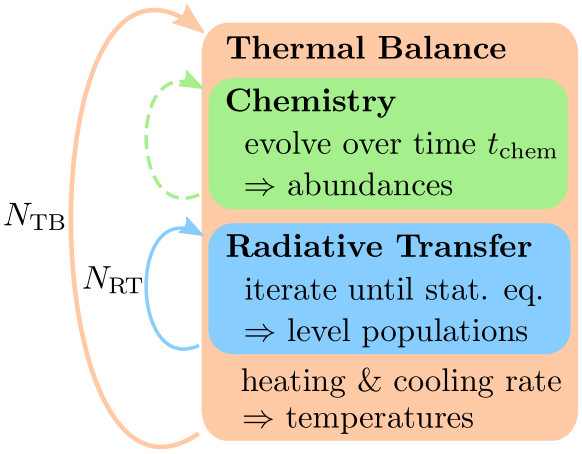

Figure 1. Iteration schemes of the Magritte workflow, different boxes represent the different modules.

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 is 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

[1] 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 might optimize for certain grids.
[2] healpix.sourceforge.net

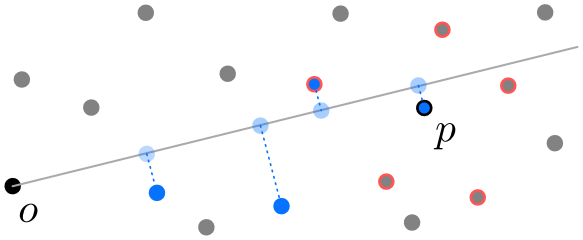

Figure 2. Sketch of the algorithm that chooses which cells to project on a ray. All cells are represented by their cell centers. The ray originates from the black cell (denoted o). The blue cells are already projected onto the ray. The blue cell with the black border (denoted p) is the last cell that 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.

be stored which scales as $N_{\text{cells}} \times N_{\text{cells}} \times N_{\text{rays}}$. 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. This reduces the scaling of the memory cost to $N_{\text{cells}} \times N_{\text{rays}}$ 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[3] 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,

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

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,

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

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

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

[3] 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.

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

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

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

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

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

$$R_{ij}(\boldsymbol{x}) \;=\; A_{ij}\left(1-\Lambda^*_{ij}(\boldsymbol{x})\right) \;+\; B_{ij}J^{\text{eff.}}_{ij}(\boldsymbol{x}) \;+\; C_{ij}(\boldsymbol{x}), \tag{11}$$

where the approximated lambda operator $\Lambda^*_{ij}(\boldsymbol{x})$ and the resulting effective mean intensity $J^{\text{eff.}}_{ij}(\boldsymbol{x})$ are given by

$$\begin{aligned} \Lambda^*_{ij}(\boldsymbol{x}) \;&=\; \\ J^{\text{eff.}}_{ij}(\boldsymbol{x}) \;&=\; \end{aligned} \tag{12}$$

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 $\mathcal{N}_{\text{pop}} \approx 3$, 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[4].

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!).

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.

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

References

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

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

Haworth T. J., Glover S. C. O., Koepferl C. M., Bisbas T. G., Dale J. E., 2017, preprint, (arXiv:1711.05275)

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

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

Woodall J., Agúndez M., Markwick-Kemper A. J., Millar T. J., 2007, A&A, 466, 1197

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

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

Substituted in equation (11) this yiels

$$R_{ij}(\boldsymbol{x}) \;=\; \beta_{ij}(\boldsymbol{x}) \Big(A_{ij} \;+\; B_{ij} J_{ij}^{\text{con.}}(\boldsymbol{x})\Big) \;+\; C_{ij}(\boldsymbol{x}), \tag{A5}$$

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 the specific monochromatic intensity $I_\nu(\boldsymbol{x},\hat{\boldsymbol{n}})$ along a ray in direction $\hat{\boldsymbol{n}}$ to the local emissivity and opacity,

$$\hat{\boldsymbol{n}}\cdot\nabla I_\nu(\boldsymbol{x},\hat{\boldsymbol{n}}) \;=\; \eta_\nu^{\text{eff.}}(\boldsymbol{x},\hat{\boldsymbol{n}}) \;-\; \chi_\nu^{\text{eff.}}(\boldsymbol{x})\; I_\nu(\boldsymbol{x},\hat{\boldsymbol{n}}). \tag{B1}$$

It is usually fair to assume that both the emissivity $\eta_\nu(\boldsymbol{x})$ and the opacity $\chi_\nu(\boldsymbol{x})$ are isotropic, i.e. independent of the direction $\hat{\boldsymbol{n}}$. However, if we add the contributions for scattering (see e.g. Steinacker et al. 2013), the effective emissivity gains a direction dependence.

$$\begin{aligned} \eta_\nu^{\text{eff.}}(\boldsymbol{x},\hat{\boldsymbol{n}}) \;&=\; \eta_\nu(\boldsymbol{x}) \;+\; \eta_\nu^{\text{sca.}}(\boldsymbol{x},\hat{\boldsymbol{n}}) \\ \chi_\nu^{\text{eff.}}(\boldsymbol{x},\hat{\boldsymbol{n}}) \;&=\; \chi_\nu(\boldsymbol{x}) \;+\; \chi_\nu^{\text{sca.}}(\boldsymbol{x},\hat{\boldsymbol{n}}), \end{aligned} \tag{B2}$$

where the emissivity due to scattering is given by

$$\eta_\nu^{\text{sca.}}(\boldsymbol{x},\hat{\boldsymbol{n}}) \;=\; \chi_\nu^{\text{sca.}}(\boldsymbol{x}) \oint \mathrm{d}\Omega\; \Phi(\hat{\boldsymbol{n}}\cdot\hat{\boldsymbol{n}}')\; I_\nu(\boldsymbol{x},\hat{\boldsymbol{n}}'). \tag{B3}$$

$\chi_\nu^{\text{sca.}}(\boldsymbol{x})$ is the extra opacity solely dus to the scattering and $\Phi(\hat{\boldsymbol{n}}\cdot\hat{\boldsymbol{n}}')$ is the scattering phase function which gives the probability of an incoming photon along direction $\hat{\boldsymbol{n}}'$ to be scattered in direction $\hat{\boldsymbol{n}}$. Note that we assumed the scattering opacity to be isotropic and the phase function to only depend on the angle between the incoming and outgoing photon. These are valid assumptions if there is no significant alignment in the dust grains. For notational convenience we will drop all position dependences from here on. Now we can proceed, as suggested by Feautrier (1964), by defining

$$\begin{aligned} u_\nu(\hat{\boldsymbol{n}}) \;&=\; \frac{1}{2}\Big(I_\nu(\hat{\boldsymbol{n}}) \;+\; I_\nu(-\hat{\boldsymbol{n}})\Big), \\ v_\nu(\hat{\boldsymbol{n}}) \;&=\; \frac{1}{2}\Big(I_\nu(\hat{\boldsymbol{n}}) \;-\; I_\nu(-\hat{\boldsymbol{n}})\Big). \end{aligned} \tag{B4}$$

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 $v_\nu(\hat{\boldsymbol{n}})$.

$$\begin{aligned} \hat{\boldsymbol{n}}\cdot\nabla u_\nu(\hat{\boldsymbol{n}}) \;&=\; \frac{1}{2}\Big(\eta_\nu^{\text{eff.}}(\hat{\boldsymbol{n}}) \;-\; \eta_\nu^{\text{eff.}}(-\hat{\boldsymbol{n}})\Big) \;-\; \chi_\nu^{\text{eff.}}\; v_\nu(\hat{\boldsymbol{n}}) \\ \hat{\boldsymbol{n}}\cdot\nabla v_\nu(\hat{\boldsymbol{n}}) \;&=\; \frac{1}{2}\Big(\eta_\nu^{\text{eff.}}(\hat{\boldsymbol{n}}) \;+\; \eta_\nu^{\text{eff.}}(-\hat{\boldsymbol{n}})\Big) \;-\; \chi_\nu^{\text{eff.}}\; u_\nu(\hat{\boldsymbol{n}}) \end{aligned} \tag{B5}$$

Defining the effective monochromatic optical depth

$$\mathrm{d}\tau_\nu \;=\; \chi_\nu^{\text{eff.}}\; \hat{\boldsymbol{n}}\cdot\mathrm{d}\boldsymbol{x}, \tag{B6}$$

equation (B5) can be solved for $u_\nu(\boldsymbol{x},\hat{\boldsymbol{n}})$.

$$\begin{aligned} \frac{\mathrm{d}^2 u_\nu(\hat{\boldsymbol{n}})}{\mathrm{d}\tau_\nu^2} \;&=\; u_\nu(\hat{\boldsymbol{n}}) \;-\; \frac{\eta_\nu^{\text{eff.}}(\hat{\boldsymbol{n}}) \;+\; \eta_\nu^{\text{eff.}}(-\hat{\boldsymbol{n}})}{2\chi_\nu^{\text{eff.}}} \\ &\quad+\; \frac{\mathrm{d}}{\mathrm{d}\tau_\nu}\left(\frac{\eta_\nu^{\text{eff.}}(\hat{\boldsymbol{n}}) \;-\; \eta_\nu^{\text{eff.}}(-\hat{\boldsymbol{n}})}{2\chi_\nu^{\text{eff.}}}\right) \end{aligned} \tag{B7}$$

Substituting the effective emissivities obtained in equation (B11) gives the Feautrier equation including scattering,

$$\frac{\mathrm{d}^2 u_\nu(\hat{\boldsymbol{n}})}{\mathrm{d}\tau_\nu^2} \;=\; u_\nu(\hat{\boldsymbol{n}}) - S_\nu^{\text{eff.}}(\hat{\boldsymbol{n}}), \tag{B8}$$

where the effective source function is defined as

$$\begin{aligned} S_\nu^{\text{eff.}}(\hat{\boldsymbol{n}}) \;&=\; \frac{\eta_\nu}{\chi_\nu^{\text{eff.}}} \;-\; \frac{\chi_\nu^{\text{sca.}}}{\chi_\nu^{\text{eff.}}} \oint \mathrm{d}\Omega\; \Phi(\hat{\boldsymbol{n}}\cdot\hat{\boldsymbol{n}}')\; u_\nu(\hat{\boldsymbol{n}}') \\ &\quad+\; \frac{\mathrm{d}}{\mathrm{d}\tau_\nu}\left(\frac{\chi_\nu^{\text{sca.}}}{\chi_\nu^{\text{eff.}}} \oint \mathrm{d}\Omega\; \Phi(\hat{\boldsymbol{n}}\cdot\hat{\boldsymbol{n}}')\; v_\nu(\hat{\boldsymbol{n}}')\right). \end{aligned} \tag{B9}$$

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 $S_\nu^{\text{eff.}}(\hat{\boldsymbol{n}})$. 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 clearly beneficial to move the forward scattering term from the emissivity to the opacity,

$$\begin{aligned} \eta_\nu^{\text{eff.}}(\hat{\boldsymbol{n}}) &= \eta_\nu + \eta_\nu^{\text{sca.}}(\hat{\boldsymbol{n}}) \\ \chi_\nu^{\text{eff.}} &= \chi_\nu + \Big(1 - \Phi(\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{n}})\Big) \chi_\nu^{\text{sca.}} \end{aligned} \tag{B10}$$

such that now the emissivity reads

$$\eta_\nu^{\text{sca.}}(\hat{\boldsymbol{n}}) = \chi_\nu^{\text{sca.}} \sum_{\hat{\boldsymbol{n}}' \neq \hat{\boldsymbol{n}}} \Phi(\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{n}}')\ I_\nu(\hat{\boldsymbol{n}}'). \tag{B11}$$

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.