

Python Radiative Transfer Emission code (PyRaTE): non-LTE spectral lines simulations

A. Tritsis^{1,2}, H. Yorke^{3,4}, K. Tassis^{1,5}

¹*Department of Physics and ITCP *, University of Crete, PO Box 2208, 71003 Heraklion, Crete, Greece*

²*Research School of Astronomy and Astrophysics, Australian National University, Canberra, ACT 2611, Australia.*

³*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA*

⁴*SOFIA Science Center, NASA Ames Research Center, 94035 Moffett Field, USA*

⁵*IESL and Institute of Astrophysics, Foundation for Research and Technology-Hellas, PO Box 1527, 71110 Heraklion, Crete, Greece*

31 March 2020

ABSTRACT

We describe PYRATE, a new, non-local thermodynamic equilibrium (non-LTE) line radiative transfer code developed specifically for post-processing astrochemical simulations. Population densities are estimated using the escape probability method. When computing the escape probability, the optical depth is calculated towards all directions with density, molecular abundance, temperature and velocity variations all taken into account. A very easy-to-use interface, capable of importing data from simulations outputs performed with all major astrophysical codes, is also developed. The code is written in PYTHON using an “embarrassingly parallel” strategy and can handle all geometries and projection angles. We benchmark the code by comparing our results with those from RADEX (van der Tak et al. 2007) and against analytical solutions and present case studies using hydrochemical simulations. The code is available on GITHUB.[†]

Key words: Physical data and processes: Radiative transfer – ISM: clouds – ISM: molecules – methods: numerical

1 INTRODUCTION

Recent advancements in computer technologies, have allowed for simulations of multiphysics astrophysical problems to flourish. Numerical codes and methods are becoming increasingly sophisticated and accurate (Krumholz et al. 2007; Mocz et al. 2014a; Mocz et al. 2014b; Schaal et al. 2015; Hopkins 2015; Hopkins & Raives 2016). Astrochemical simulations in which the chemical and dynamical evolution of astrophysical objects are coupled, follow in the same direction (Tassis et al. 2012a; Tassis et al. 2012b; Clark et al. 2013; Motoyama et al. 2015; Walch et al. 2015; Tritsis et al. 2016; Seifried & Walch 2016).

Observational astrophysics is also advancing rapidly. New, very high angular and spectral resolution data obtained with telescopes such as ALMA can reveal features of astrophysical objects in unprecedented detail. Furthermore, such observations offer a unique opportunity to distinguish between different theoretical models. To do so, a connection between theoretical results and observations is

essential. Since observationally our full range of knowledge about astrophysical objects is obtained through the study of electromagnetic radiation, a direct comparison requires post-processing simulation outputs with radiative transfer (RT) codes.

Post-processing astrochemical simulations with line radiative transfer codes is not a trivial task. Local thermodynamic equilibrium (LTE) is not attained in many astrophysical problems and the population densities of the species under consideration have to be computed in non-local thermodynamic equilibrium (non-LTE). The majority of the existing radiative transfer codes (Keto 1990; Keto et al. 2004; Hogerheijde & van der Tak 2000; van der Tak et al. 2007; Brinch & Hogerheijde 2010; Dullemond 2012) do compute the population densities in non-LTE. However, the validity of certain approximations and assumptions, as for example is homogeneity and the calculation of optical depth from average quantities, may not always be appropriate. Furthermore, restrictions in the number of dimensions existing RT codes can handle and the fact that the interfaces are not always particularly practical can cause confusion and limit the number of choices a user has.

In this paper we describe the PYRATE (Python

* Institute for Theoretical and Computational Physics, formerly Institute for Plasma Physics

[†] <https://github.com/ArisTr/PyRaTE>

Radiative Transfer Emission) code. PYRATE is a user-friendly, “embarrassingly parallel”, modular, non-LTE radiative transfer code, fully written in PYTHON¹. It makes use of the YT analysis toolkit (Turk et al. 2011) which can handle simulation outputs performed with all major codes² as well as generic adaptive mesh refinement (AMR) and array data. As a result, exporting data from astrochemical simulations requires no effort. Moreover, PYRATE can handle all geometries and all projection angles. In its current version, the code is focused on star formation and molecular cloud related problems. However, it can be easily modified so that it can be used for any astrophysical problem.

In section § ?? we describe the formalism of radiative transfer used and the numerical methods, we compare our results with those from RADEX (van der Tak et al. 2007) and with analytical solutions, we outline the parallelism strategy followed and the design of the code. In § ?? we demonstrate the code’s capabilities on hydrochemical numerical simulations. We use these simulations to demonstrate the code’s capabilities. A summary of this work and future goals are given in § ??.

2 CODE DESIGN

2.1 Basic formalism

Integration of the non-relativistic, time-independent equation of radiation transfer between two grid points i and $i+1$ yields:

$$I_{i+1} = \frac{(e^{-\tau_{i+1}^C} - p)I_i + pS_i^L + qS_{i+1}^L + S^k}{1 + q} \quad (1)$$

(Yorke 1986) where the contributions of line and dust continuum emission, denoted with the superscripts L and C respectively, are considered separately. In Equation ?? I is the radiative intensity, S^L is the source function for line emission and τ^C is the optical depth for continuum emission. The quantities q , p and S^k are given by:

$$q = \frac{\tau_{i+1}^L}{1 + e^{-\tau_{i+1}^L}} \quad (2)$$

$$p = q(e^{-\tau_{i+1}^L - \tau_{i+1}^C}) \quad (3)$$

$$S^k = e^{-\tau_{i+1}^C} \int_i^{i+1} \kappa^C S^C e^{\int_i^s \kappa^C(s') ds'} ds \quad (4)$$

where S^C is the source function for dust continuum emission and τ_{i+1}^L is the optical depth of the line:

$$\tau_{i+1}^L = \int_i^{i+1} \kappa^L(s) ds \quad (5)$$

In Equations ?? and ?? s is the length along the line-of-sight (LOS), κ^C is the extinction coefficient for continuum emission which can be computed from Mie theory (see Section

??) and κ^L is the line extinction coefficient:

$$\kappa^L = n_m B_{mn} \frac{h\nu_0}{4\pi} \left[1 - \frac{n_n g_m}{n_m g_n}\right] \varphi(\nu) \quad (6)$$

(Mihalas 1978). In the latter equation, g is the statistical weight, B is the Einstein B coefficient, ν_0 is the rest frequency of the line, h is the Planck constant, n is the population density of the upper and lower level (denoted with the subscripts n and m) and $\varphi(\nu)$ is the normalized profile function such that $\int \varphi(\nu) d\nu = 1$. As a result, I can be computed by calculating the quantities that appear in Equations ?? through ??, inserting them in Equation ?? and performing the ray-tracing.

2.2 Dust model

We use a three component dust grain model consisting of amorphous carbon grains, silicate grains and dirty ice (i.e. water, ammonia and carbon particles) coated silicate grains. Our grain model is that of Preibisch et al. (1993) which is appropriate for dense molecular cloud conditions. However, the amount of dust has been reduced to be compatible with the calculations performed by Pollack et al. (1985). The number of grains per gram gas are $\log N_d = 14.170$, 1. and 12.001 gr^{-1} for the amorphous carbon, silicate and dirty ice coated silicate grains respectively. The sublimation temperature of each component is ~ 2000 , 1500 and 125 K respectively. The total amount of amorphous carbon is kept constant by enhancing the number density of amorphous carbon grains in temperatures above 125 K. For every dirty ice coated silicate grain destroyed, a silicate grain is also added. All components are assumed to be spherical with sizes ~ 10 nm for the carbon, ~ 50 nm for the silicate and ~ 60 nm for the dirty ice coated silicate grains.

The data for the efficiency factor for extinction Q^{ext} and the albedo α ($\alpha = Q^{sca}/Q^{ext}$, where Q^{sca} is the efficiency factor for scattering) are stored in the code in tabulated form. Based on the desired frequency, linear interpolation is applied through the data for each of the dust components. The extinction coefficient for dust emission is then computed as:

$$\kappa^C = \sum_{d=1}^3 n_d Q_d^{ext} \pi r_d^2 \quad (7)$$

where n_d is the number density of each grain component which is in turn determined from the temperature and density of each grid point. The source function for continuum emission is computed as the sum of the thermal emission, weighted over the albedo, and the contribution from scattering. For the interpolation we use a standard SCIPY routine³. Higher order interpolation through the data can also be used by changing a keyword in the interpolating function. For further details regarding our dust model and figures showing the absorption efficiencies as a function of frequency for each dust component we refer the reader to Preibisch et al. (1993) and Kessel et al. (1998). Finally, by adjusting this dust grain model, PYRATE can be modified to suit a variety of astrophysical problems.

¹ <https://www.python.org/>

² For a full list of the codes that YT and thus PYRATE can handle see http://yt-project.org/docs/dev/examining/loading_data.html

³ <http://docs.scipy.org/doc/scipy/reference/interpolate.html>

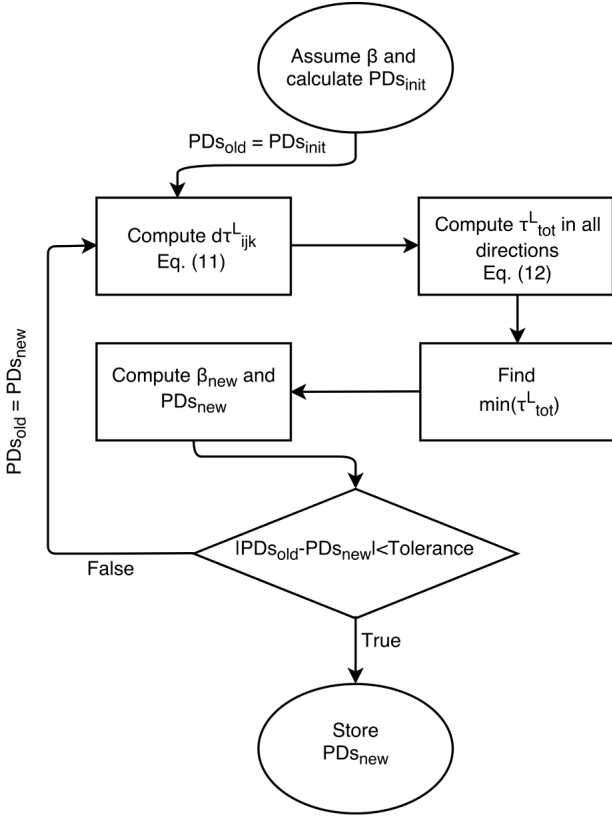


Figure 1. Flow chart of the algorithm used to compute the population densities (PDs) at each grid point. The same procedure is followed for all grid points.

2.3 Computing the population ratio (non-LTE)

We consider a multilevel system. The rate of change of the population density in level m will be the sum of the excitation (absorption and collisional excitation) and de-excitation (induced emission, spontaneous emission and collisional de-excitation) processes:

$$\begin{aligned} \frac{dn_m}{dt} = & \sum_{n>m}^N [n_n(A_{nm} + B_{nm}u_{nm}^\nu + n_{H_2}C_{nm}) \\ & - n_m(B_{mn}u_{mn}^\nu + n_{H_2}C_{mn})] + \\ & \sum_{n<m}^N [-n_m(A_{mn} + B_{mn}u_{mn}^\nu + n_{H_2}C_{mn}) \\ & + n_k(B_{nm}u_{nm}^\nu + n_{H_2}C_{nm})] \end{aligned} \quad (8)$$

In Equation ??, u_{nm}^ν is the radiative energy density between levels n and m , C_{nm} and C_{mn} (hereafter $n > m$) are respectively the de-excitation and excitation coefficients due to collisions, n_{H_2} is the number density of the H_2 molecule which is the dominant collisional partner, and A_{nm} and B_{nm} are the Einstein A and B coefficients for spontaneous and induced emission. De-excitation coefficients are computed from the detailed balance relation:

$$C_{mn} = C_{nm} \frac{g_n}{g_m} e^{-\Delta E/k_B T} \quad (9)$$

where k_B is the Boltzmann constant, T is the temperature and ΔE is the energy difference between levels. In order to

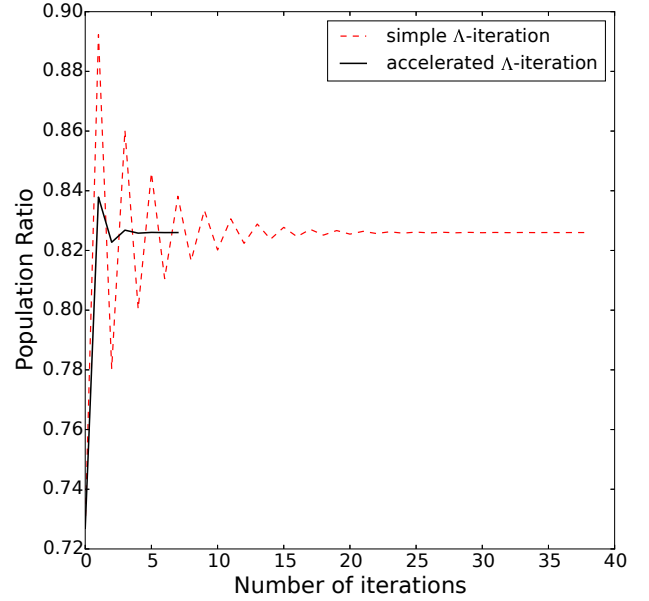


Figure 2. The black solid line shows the performance of the accelerated Λ -iteration algorithm currently in use for calculating the population densities. The red dashed line shows results using a simple Λ -iteration. The current algorithm converges in just a few iterations. Calculations were performed considering the optically thick ^{12}CO molecule ($J = 1 - 0$ transition) and the physical parameters shown in Figure ??.

compute the terms containing the Einstein B coefficients on the right hand side of Equation ?? a priori knowledge of the radiation field is required. As a result, the entire problem has to be solved iteratively, a computationally very demanding task. An alternative solution to the problem is to approximate u_ν as $S_\nu^L(1 - \beta)$ where β is the probability of a photon escaping the cloud. Following the same reasoning, the radiation field due to external photons penetrating the cloud will be, $S_\nu\beta$.

If we now further assume statistical equilibrium $\frac{dn_m}{dt} = 0$ and take into account the conservation equation:

$$\sum_m^N n_m = 1$$

we have a system of non-linear equations which PYRATE solves using an optimization routine from SCIPY and which supports multiple methods for solution of non-linear systems⁴.

In order to determine the population densities we also have to compute the escape probabilities between radiatively coupled levels at every grid point. To do so, we use an accelerated Λ -iteration described in the following steps:

- (i) First, we assume some initial values for the escape probabilities (equal to 0.5 for all levels) and calculate the initial population densities for grid point $(i'j'k')$ from Equation ?? and the conservation equation.

⁴ <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.root.html>

(ii) Based on that initial guess for the population densities, the infinitesimal optical depth for all lines is computed at all grid points as:

$$d\tau_{ijk}^L = \frac{c^3}{8\pi\nu_0^3} \frac{A_{nm}n_{ijk}^{molec}}{\Delta v_{ijk}^{th}} \left(\frac{g_n}{g_m} n_m - n_n \right) ds \quad (10)$$

(van der Tak et al. 2007). In Equation ??, c is the speed of light, n_{ijk}^{molec} is the number density of the molecule at grid point (i, j, k) , Δv_{th} is the thermal width of the line and $n > m$.

(iii) Next, the total optical depth for all lines is computed as the sum of the infinitesimal optical depth of the grid points for which their absolute velocity difference with grid point $(i'j'k')$ is smaller than their thermal width.

(iv) The algorithm then searches the direction towards which the optical depths of the lines are minimum and the photons are more likely to escape the cloud (Poelman & Spaans 2006; Perego et al. 2014; Scarlata & Panagia 2015). For example, the minimum optical depth for a 3D cartesian grid is computed from six sums as:

$$\tau_{i'j'k'}^L = \min \left(\begin{aligned} & \sum_{i=i'+1}^A d\tau_{ijk}^L [|v_{i'j'k'} - v_{ijk}| < \Delta v_{ijk}^{th}], \\ & \sum_{i=i'-1}^0 d\tau_{ijk}^L [|v_{i'j'k'} - v_{ijk}| < \Delta v_{ijk}^{th}], \\ & \sum_{j=j'+1}^B d\tau_{ijk}^L [|v_{i'j'k'} - v_{ijk}| < \Delta v_{ijk}^{th}], \\ & \sum_{j=j'-1}^0 d\tau_{ijk}^L [|v_{i'j'k'} - v_{ijk}| < \Delta v_{ijk}^{th}], \\ & \sum_{k=k'+1}^C d\tau_{ijk}^L [|v_{i'j'k'} - v_{ijk}| < \Delta v_{ijk}^{th}], \\ & \sum_{k=k'-1}^0 d\tau_{ijk}^L [|v_{i'j'k'} - v_{ijk}| < \Delta v_{ijk}^{th}] \end{aligned} \right) \quad (11)$$

where A , B and C are the sizes of the grid in the i , j and k directions, respectively.

(v) With that optical depths, new escape probabilities and population densities are computed.

(vi) If the relative change of the previously and newly computed population densities is smaller than some tolerance, then the values of the population densities are stored in computer memory. If this condition is not satisfied steps (ii) through (vi) are repeated until the population densities converge to some values.

A flow chart of the algorithm that summarizes these steps is shown in Figure ?. With this procedure, density, temperature, velocity and molecular abundance variations are all taken into account. After the population densities have been computed, the line source function is calculated as:

$$S^L = \frac{2h\nu^3}{c^2} \frac{1}{\frac{n_m g_n}{n_n g_m} - 1} \quad (12)$$

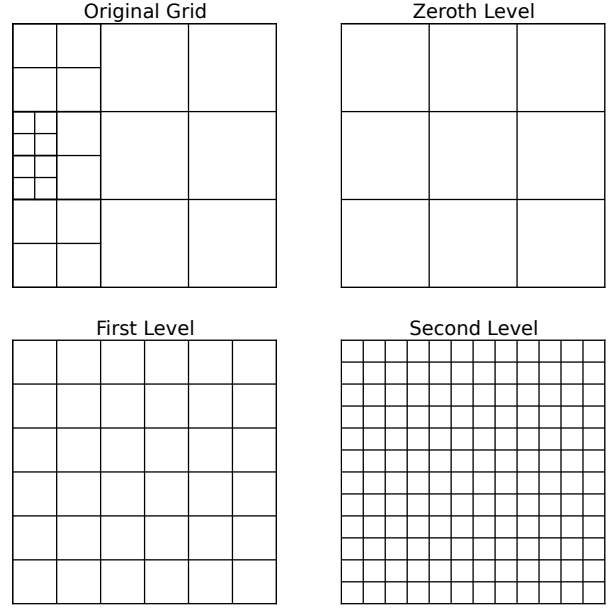


Figure 3. Treatment of AMR simulation outputs. Depending on the desired resolution, from the original AMR grid, a new uniform grid is created. When the resolution of the maximum level of refinement is selected, new grid points are created with values equal to the one of the grid point of the parent level.

The described procedure is a simple Λ -iteration. To increase the computational speed we implement an accelerated scheme. In each successive iteration, the new population densities are a weighted mean of the two previously computed values instead of just the previously computed value. Furthermore, since escape probabilities are not expected to vary much between adjacent grid points, results for one grid point are used as initial guesses for the neighbouring cell. In Figure ?? we show an example where the population ratio for a grid point is calculated using simple (red dashed line) and accelerated (black solid line) Λ -iterations. In the latter case the algorithm converges within a few iterations. Both tests were performed using the 1D isothermal spherical collapse model (the physical parameters of which are shown in Figure ??) with 304 grid points, a tolerance of 10^{-5} and the optically thick ^{12}CO molecule.

2.4 Loading data and constructing the grid

Data from astrochemical simulations and from files containing molecular data⁵ are initially loaded into the code and stored in computer memory in tabulated form. These include the mass density, the abundance of the molecule under consideration, the Einstein coefficients, collisional excitation/de-excitation coefficients, the dimensions of the grid and velocities in all directions. Magnetic fields for Zeeman splitting measurements will also be included in a future version.

⁵ Molecular data are taken from the Leiden Atomic and Molecular Database (Schöier et al. 2005) <http://home.strw.leidenuniv.nl/~moldata/>

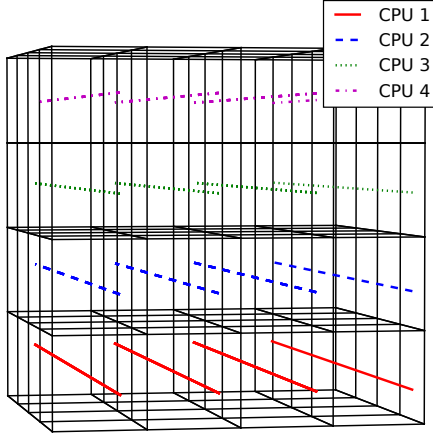


Figure 4. Strategy followed to parallelize the algorithms in our code. During ray tracing each processor is responsible only for a set of rays (coloured lines) passing through the computational region.

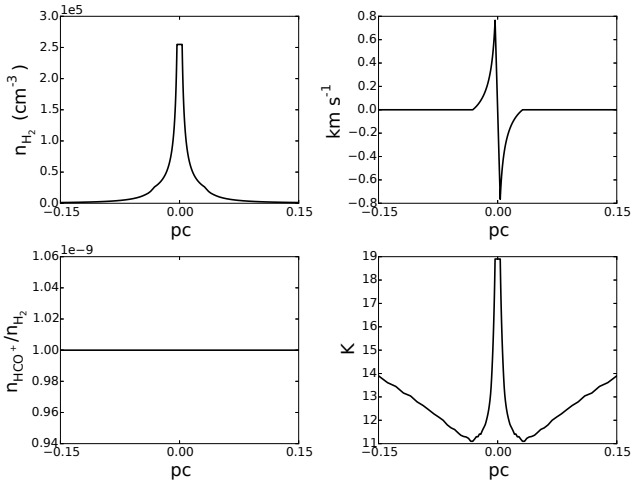


Figure 5. Parameters of the spherical model adopted from RADEX’s test problem web page (van Zadelhoff et al. 2002) to investigate the accuracy of the algorithm developed here for computing the population ratio. In the upper left panel we plot the density, in the upper right the velocity, in lower left the molecular abundance and in the lower right the temperature.

When loading data we assume spherical symmetry for 1D simulation outputs, plane and axial symmetries for 2D simulation outputs in cylindrical geometry and no symmetries for 3D simulations. These symmetries are accounted for in our calculations. However, the module responsible for exporting simulation data can be easily modified such that the desired symmetries and velocity conventions are followed. Where necessary, volume rendering is performed simultaneously with ray tracing. Throughout the code, we use cgs units. Velocities with a minus sign represent motions

towards the observer and vice versa. All other conventions followed in the code (e.g. conversion of frequencies to velocities) are consistent with the ones followed in radio astronomy.

Simulation outputs performed both in adaptive mesh refinement (AMR) and in uniform grids are supported. However, AMR grids are re-sampled into uniform grids based on the resolution of a refinement level. In the upper left panel of Figure ?? we show an example of an AMR grid with two levels of refinement. In the upper right, lower left and lower right panels we show the resulting grids constructed from the original grid, based on the zeroth, first and second level of refinement respectively. The values of additional grid points created in the lower panels of Figure ?? have the same values as the ones of their parent level. This treatment has the advantage of simplifying the algorithms and making them easy to read and modify. However, adding more grid points can substantially increase the computational time, especially for AMR simulations with many levels of refinement. In a future version we plan to upgrade the treatment of AMR grids.

2.5 Parallelization

In order to reduce computational time we have parallelized the majority of the code’s modules using an “embarrassingly parallel” strategy. “Embarrassingly parallel” or naturally parallel problems are problems that can be decomposed in identical subtasks all of which can be solved independently without the need for communication between processors. In Figure ?? we show an example of a 3D grid in which ray tracing is performed on four CPUs. Each CPU is responsible only for a set of rays (coloured lines in Figure ??). Similarly, in the modules computing the population ratios, the grid is split over the number of available processors and is then reconstructed. Given this strategy the code is linearly scalable. PYRATE uses the package MPI4PY⁶ which provides functions and bindings for the Message Passing Interface (MPI) and which can be easily installed alongside YT.

2.6 Auxiliary modules

A number of modules are available for pre-processing the simulation data and post-processing the results after ray-tracing is performed. The user has the option to add micro-turbulent broadening:

$$\Delta v_{tot} = \sqrt{\Delta v_{turb}^2 + \Delta v_{th}^2} \quad (13)$$

where Δv_{th} is the thermal linewidth and Δv_{turb} may be a constant number for all grid points or random numbers drawn from a flat or a Gaussian distribution. Additional modules also exists for debugging purposes and for creating simple models so there is no need for a priori astrochemical simulations.

To facilitate comparison with observations the user has a number of options. Gaussian noise can be added to spectral lines, based on a user-defined signal-to-noise ratio (SNR) and emission maps can be convolved with Gaussian filters

⁶ <https://pypi.python.org/pypi/mmpi4py>

based on some desired distance and telescope's beam size. Spectral lines can be plotted in antenna or brightness temperature, mJy or in cgs units. Furthermore, lines can be smoothed to match the spectral resolution of observations. The resulting output format of PyRATE is equivalent to a position-position-velocity (PPV) cube from observations which can also be saved in FITS format for further analysis. Options for creating integrated emission maps are also available.

2.7 Benchmarking

In order to test the validity of the algorithm developed we compare our results with those of RADEX (van der Tak et al. 2007). To do so, we adopt the set-up of a test-problem of RATRAN (Hogerheijde & van der Tak 2000; van Zadelhoff et al. 2002), the parameters of which are shown in Figure ???. In this model, the abundance of HCO^+ is low ($\sim 10^{-9}$), so that the emission is optically thin, and the radial profiles of the physical parameters were created based on the model by Shu (1977). Results for the excitation temperature for $J = 1 - 0$ and $J = 2 - 1$ obtained with both codes, are shown in Figure ?? and are identical. For producing Figure ??, instead of computing the optical depth as described in section ?? we have made the same approximations as in RADEX, namely that the optical depth can be computed from average quantities as:

$$d\tau^L = \frac{c^3}{8\pi\nu_0^3} \frac{A_{nm}N^{molec}}{1.064\Delta V} \left(\frac{g_n}{g_n} n_m - n_n \right) \quad (14)$$

where ΔV is the full width at half-maximum (FWHM) of the line and N^{molec} is the total molecular column density both of which are given as a model parameters. This formalism leads to an overestimation of the optical depth, low escape probabilities and thus more molecules in excited states.

In order to demonstrate this effect we plot in Figure ?? the population ratios obtained with RADEX and PyRATE for the same physical model. However, HCO^+ abundance is increased by 10^2 . For calculating the optical depth we have used the algorithm described in section ??. Because of the fact that the optical depth computed with PyRATE is lower than that computed from average quantities there is not enough photon trapping and thus less molecules are in excited states. The kinks in the population ratios (solid and dashed red lines) are due to abrupt changes in the escape probabilities. In turn, this abrupt change is caused by the velocity profile (see Figure ??). Beyond a radius of ~ 0.025 pc the absolute velocity difference between grid points is always less than the thermal width and as a result the optical depth rises. Thus, the kinks shown in Figure ?? are because of the adopted test problem and *not* because of numerical resolution effects. The same behaviour is not seen when the abundance is 10^{-9} and the emission is optically thin everywhere.

To further test the developed algorithms we compare results from PyRATE against the analytical solutions derived in Kylafis (1983) for an one dimensional problem applicable to CO. In this problem, the velocity field is of the form $\vec{v} = (v_0/L)z\hat{k}$ and the temperature, CO abundance and density are uniform everywhere. In Figure ?? we show the

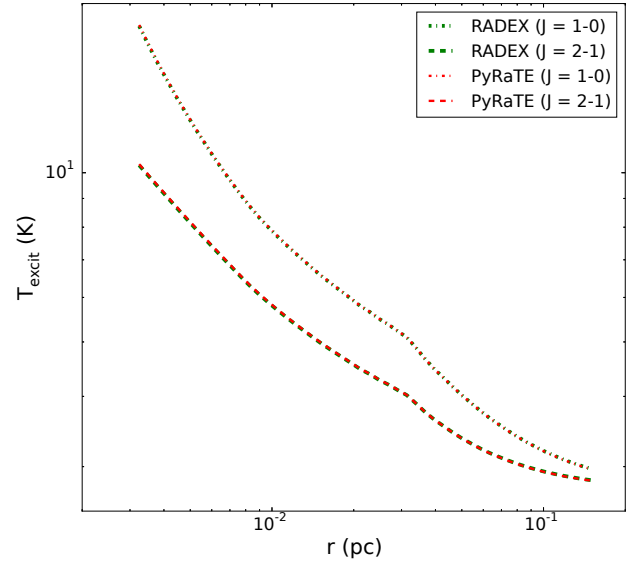


Figure 6. Comparison of the excitation temperatures obtained with PyRATE (dashed-dotted and dashed red lines) and RADEX (dashed-dotted and dashed green lines). When we follow the same approximations with RADEX results obtained with the two codes are identical. The parameters of the model used to compare the population densities between the two codes are shown in Figure ??.

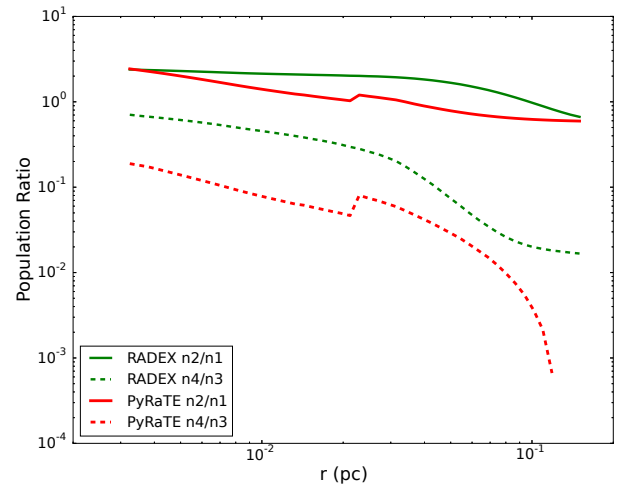


Figure 7. Comparison of population densities obtained with PyRATE (solid and dashed red lines) and RADEX (solid and dash-dotted green lines) for the same physical model as in Figure ?? but with 10^2 higher HCO^+ abundance. Using the algorithm described in ?? the optical depth computed with PyRATE is smaller and thus less molecules are in excited states.

intensity at the centre of the line ($J = 1 - 0$) as this is calculated from the analytical expressions (solid black line) and numerically (red squares) as a function of the "mean" optical depth defined in Kylafis (1983). Results from the code are in perfect agreement with the analytical method.

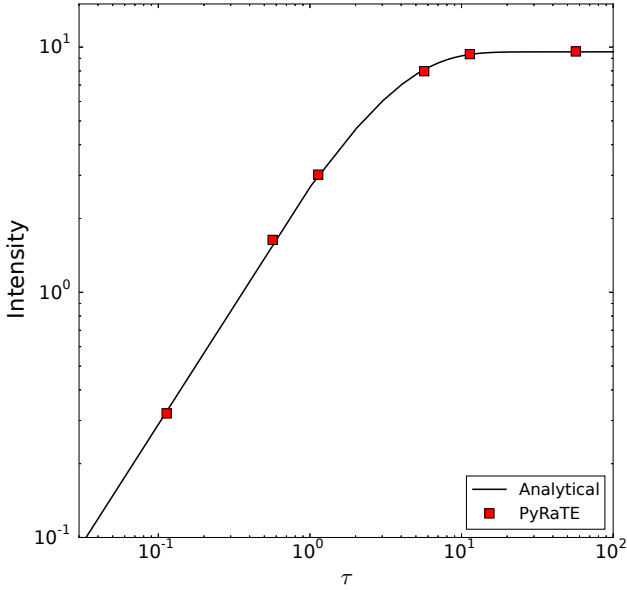


Figure 8. The intensity at the centre of the line CO (1 – 0) as a function of the “mean” optical depth, as this is defined in Kylafis (1983). With the solid black line we plot the results from the analytical solution and with the red squares the results from the code. The intensity is plotted in units of $h\nu^3/c^2$.

3 TEST CASES

The numerical simulations used to demonstrate PYRATE’s capabilities were performed with the astrophysical code FLASH4.0.1 (Fryxell et al. 2000, Dubey et al. 2008) and were described in detail in Tritsis et al. (2016). Here we give a brief overview of these simulations. In Tritsis et al. (2016) we employed hydrodynamic simulations of prestellar cores in spherical and cylindrical symmetry. The radius of our spherically symmetric models was 0.55 pc. The axial and radial dimensions of one of our cylindrically symmetric models were 0.72 and 0.48 pc respectively. The initial number density was 10^3 cm^{-3} . In all models, all velocity components were initially set to zero and the cores were allowed to collapse freely under their self-gravity. The dynamical models were isothermal and coupled with non-equilibrium chemical network consisting of 214 gas-phase and 82 dust grain species with each species treated as a different fluid. Our chemical network consisted of ~ 14000 reactions with reaction rates adopted from the fifth release of the UMIST database (McElroy et al. 2013).

In Figure ?? we show the density (upper left), velocity (upper right), CO abundance (lower right) and H₂O abundance distribution for our spherical model when the central density is $4 \cdot 10^5 \text{ cm}^{-3}$. The temperature is constant and equal to 7 K everywhere. This evolutionary stage corresponds to a time of ~ 1 Myrs. In the left panel Figure ?? we show a ^{12}CO ($J = 1 - 0$) emission line from a LOS threading the middle of our spherical core. With the red line we plot our results assuming LTE and the black line non-LTE calculations. The spectral resolution used to produce the synthetic lines was $\sim 0.05 \text{ km/s}$. Random, gaussian noise was added to the line with a signal-to-noise ratio (SNR) of ~ 5 . PYRATE correctly reproduces a line profile characteristic of infall motions (Evans 1999). The emission line

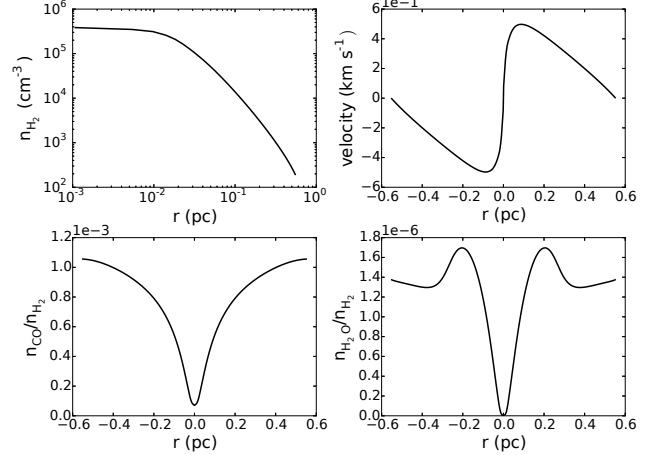


Figure 9. Physical and chemical parameters used to produced the synthetic lines shown in Figure ?? and Figure ?. The parameters are taken from a hydrodynamical simulation of a collapsing isothermal sphere. The upper left and right panels show the radial profiles of number density and velocity respectively. In the lower left panel we show the CO abundance profile and in the lower right that of H₂O.

is double-peaked with the blue component being stronger due to absorption along the LOS. The strength of the line changes depending on whether LTE is assumed. Since in non-LTE we expect that the upper level will be less populated than in LTE, the source function of the line, and thus the intensity, will decrease. This feature is clearly seen in Figure ??, especially for the blue shifted component of the line since for the red shifted component the decrease in the source function is counteracted by the decrease in the absorption coefficient. Furthermore, from Figure ?? it can be seen that the line profile changes depending on whether LTE is assumed. That is due to the velocity profile. The velocity differences between the centre of the core and the outer parts are larger than the thermal width of the line. Consequently, the total optical depth decreases, the escape probability increases and CO is only subthermally excited. However, if we were to “observe” the two lines, taking resolution restrictions into account, both profiles could probably be fitted by two Gaussian functions. In the right panel of Figure ?? we show excited-state lines for the same ray as in the left panel.

In Figure ?? we show a synthetic absorption line of H₂O ($1_{10} - 1_{01}$) from a LOS threading the middle of our spherical core. The spectral resolution was 0.1 km/s but the line was then smoothed to a spectral resolution of 0.6 km/s. The continuum level was $\sim 11 \text{ mK}$ (i.e. the same as in Caselli et al. 2010 for L1544) and was subtracted. The CO ($J = 1 - 0$) emission line combined with the absorption line for H₂O for the same physical parameters is in agreement with expectations from both observations and previous line radiative transfer simulations (Caselli et al. 2010; Caselli et al. 2012).

In order to demonstrate the capabilities of PYRATE to post-process simulations with more complex geometries we also produce a synthetic PPV cube from hydrodynamical simulations of a collapsing isothermal cylindrical/filamentary core. In Figure ?? we show the physical parameters of the core. In the left panel we show the number

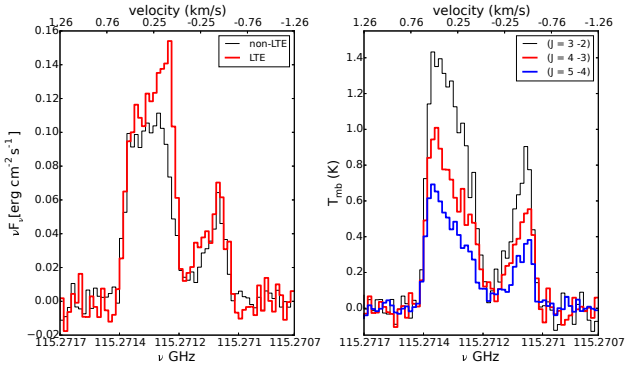


Figure 10. Left panel: ^{12}CO ($J = 1 - 0$) emission line from a ray passing through the centre of a simulated spherical core under the LTE approximation (bold red line) and with non-LTE (black line). Right panel: excited-state lines for the same ray. Gaussian noise with a signal-to-noise ratio of ~ 5 was added to the lines. The code correctly reproduces the expected infall asymmetry in the line profiles.

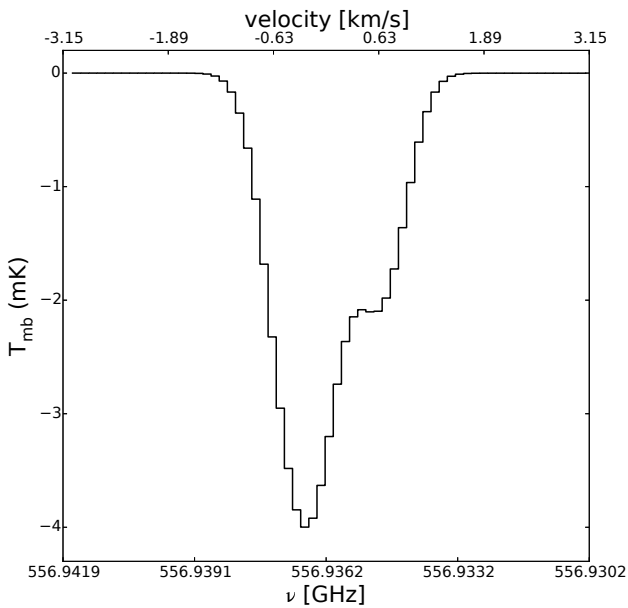


Figure 11. H_2O ($1_{10} - 1_{01}$) absorption line from a ray passing through the centre of our simulated spherical core. The continuum level has been subtracted. The line was produced using a spectral resolution of 0.1 km/s and was then smoothed to a spectral resolution of 0.6 km/s.

density distribution, in the middle panel we plot the radial velocity and in the right panel we show the N_2H^+ abundance distribution. The temperature is equal to 10 K. In Figure ?? we show four velocity slices from our synthetic PPV cube of the core as observed in N_2H^+ ($J = 1 - 0$) and with a spectral resolution of ~ 0.05 km/s. In the upper panel we plot the emission map in each velocity slice and in the lower panel a spectral line (black line) threading the centre of the map (blue points in the upper panels). The red line in the lower panel traces the velocity shown in the upper panel. The core is observed such that its axis of symmetry is perpendicular to the LOS (edge-on).

4 SUMMARY

Radiative transfer is essential when comparing results from astrochemical simulations with radio observations. To this end, we have developed an easy-to-use non-LTE line radiative transfer code. The code can be used to post-process results from astrochemical simulations performed with all major astrophysical codes and can handle all geometries and projection angles. The population densities are computed using the escape probability method with variations in density, molecular abundance, temperature and velocities taken into account.

Compared to existing radiative transfer codes the populations densities in PYRATE are *not* computed from average quantities and velocity variations are taken into account. Its simple interfaces both for importing and exporting data and the fact that it is written in PYTHON make the code easy-to-use, flexible and capable of producing publication quality figures effortlessly.

In a future version we plan to include a full treatment of the Stokes parameters in our code. Thus, simulations of the polarization of spectral line emission will also become possible. The algorithms for ray-tracing and for computing the population densities will continue to be developed both in terms of performance and accuracy. We think that PYRATE can be proven an important tool in the effort of connecting theory with observations. The code is publicly available for the community at <https://github.com/ArisTr/PyRaTE>.

ACKNOWLEDGEMENTS

We thank N. Kylafis for useful comments and discussions. We also thank the anonymous referee for useful comments that helped improved this paper. The software used in this work was in part developed by the DOE NNSA-ASC OASCR Flash Center at the University of Chicago. For post processing our results we partly used yt analysis toolkit (Turk et al. 2011). K.T. and A.T. acknowledge support by FP7 through Marie Curie Career Integration Grant PCIG-GA-2011-293531 “SFOnset”. A.T. and K.T. would like to acknowledge partial support from the EU FP7 Grant PIRSES-GA-2012-31578 “EuroCal”. Usage of the Metropolis HPC Facility at the CCQCN of the University of Crete, supported by the European Union Seventh Framework Programme (FP7-REGPOT-2012-2013-1) under grant agreement no. 316165, is also acknowledged.

REFERENCES

- Brinch, C., & Hogerheijde, M. R. 2010, *A&A*, 523, A25
- Caselli, P., Keto, E., Pagani, L., et al. 2010, *A&A*, 521, L29
- Caselli, P., Keto, E., Bergin, E. A., et al. 2012, *ApJL*, 759, L37
- Clark, P. C., Glover, S. C. O., Ragan, S. E., Shetty, R., & Klessen, R. S. 2013, *ApJL*, 768, L34
- Dubey, A., Fisher, R., Graziani, C., et al. 2008, *Numerical Modeling of Space Plasma Flows*, 385, 145
- Dullemond, C. P. 2012, *Astrophysics Source Code Library*, ascl:1202.015
- Evans, N. J., II 1999, *Annu. Rev. Astron. Astrophys.*, 37, 311
- Fryxell, B., Olson, K., Ricker, P., et al. 2000, *ApJS*, 131, 273
- Goldsmith, P. F., & Li, D. 2005, *ApJ*, 622, 938
- Hogerheijde, M. R., & van der Tak, F. F. S. 2000, *A&A*, 362, 697

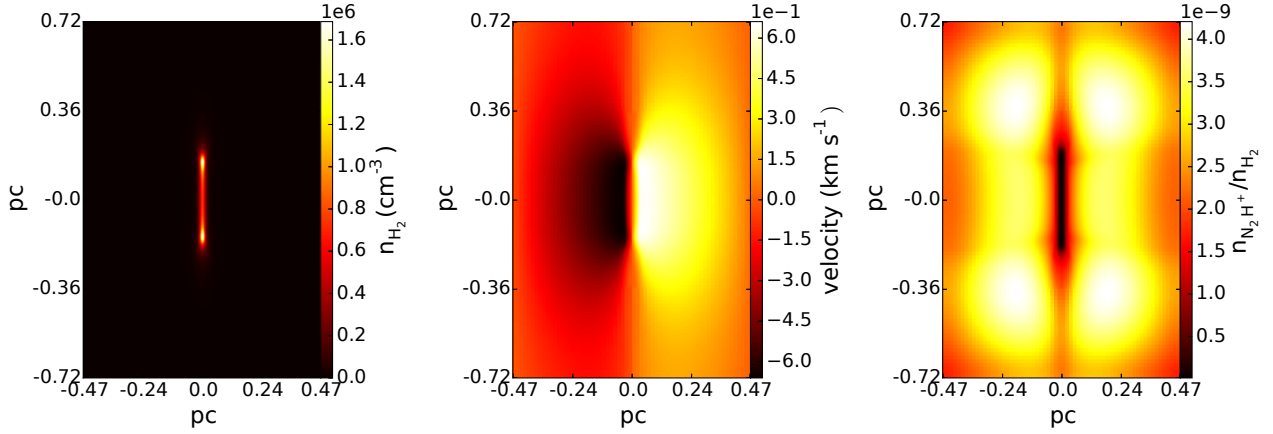


Figure 12. Physical and chemical parameters used to produced the synthetic PPV cube shown in Figure ???. The parameters are taken from a hydrodynamical simulation of a collapsing isothermal cylinder. The left, middle and right panels show the number density, radial velocity and N_2H^+ abundance distribution respectively.

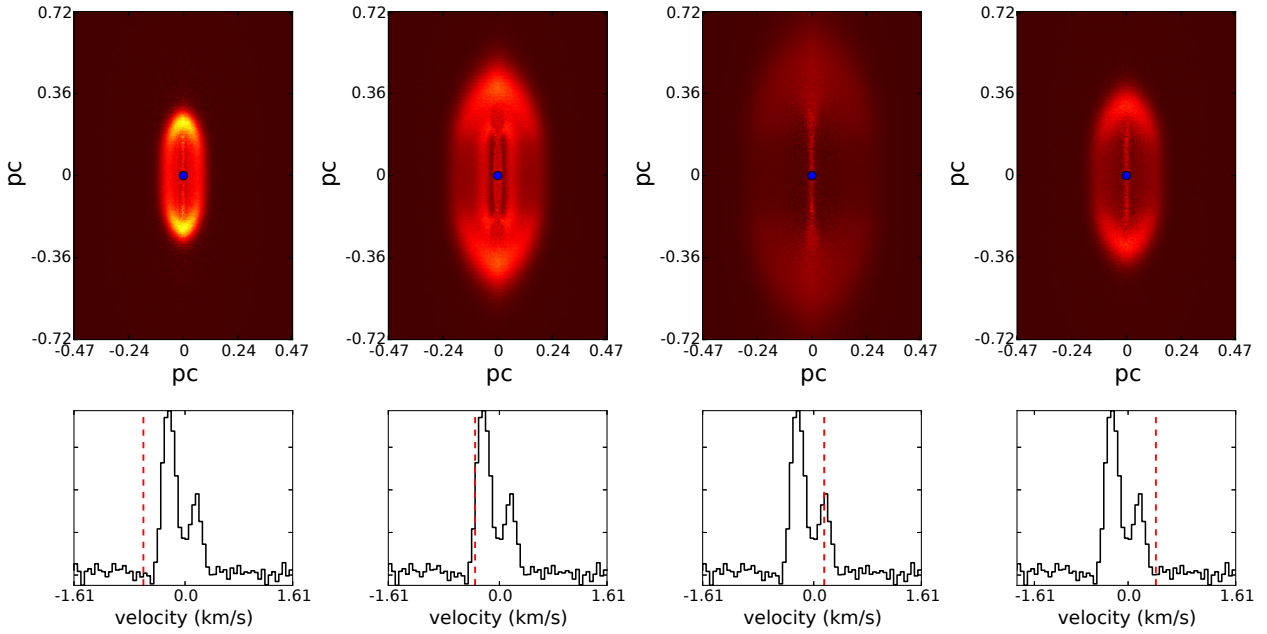


Figure 13. Four velocity slices (upper row) from a synthetic PPV cube of a collapsing isothermal cylinder seen in N_2H^+ ($J = 1 - 0$). The lower row shows the spectrum from a LOS passing through the centre of the map (blue points in upper row) while the red line traces the velocity of each slice map.

Hopkins, P. F. 2015, MNRAS, 450, 53
 Hopkins, P. F., & Raives, M. J. 2016, MNRAS, 455, 51
 Kessel, O., Yorke, H. W., & Richling, S. 1998, A&A, 337, 832
 Keto, E. R. 1990, ApJ, 355, 190
 Keto, E., Rybicki, G. B., Bergin, E. A., & Plume, R. 2004, ApJ, 613, 355
 Krumholz, M. R., Klein, R. I., McKee, C. F., & Bolstad, J. 2007, ApJ, 667, 626
 Kylafis, N. D. 1983, ApJ, 267, 137
 McElroy, D., Walsh, C., Markwick, A. J., et al. 2013, A&A, 550, A36
 Mihalas, D. 1978, San Francisco, W. H. Freeman and Co., 1978. 650 p.,
 Mocz, P., Vogelsberger, M., Sijacki, D., Pakmor, R., & Hernquist, L. 2014a, MNRAS, 437, 397

Mocz, P., Vogelsberger, M., & Hernquist, L. 2014b, MNRAS, 442, 43
 Motoyama, K., Morata, O., Shang, H., Krasnopolsky, R., & Hasegawa, T. 2015, ApJ, 808, 46
 Perego, A., Gafton, E., Cabezón, R., Rosswog, S., & Liebendörfer, M. 2014, A&A, 568, A11
 Poelman, D. R., & Spaans, M. 2006, A&A, 453, 615
 Pollack, J. B., McKay, C. P., & Christofferson, B. M. 1985, Icarus, 64, 471
 Preibisch, T., Ossenkopf, V., Yorke, H. W., & Henning, T. 1993, A&A, 279, 577
 Scarlata, C., & Panagia, N. 2015, ApJ, 801, 43
 Schaal, K., Bauer, A., Chandrasekhar, P., et al. 2015, MNRAS, 453, 4278
 Schöier, F. L., van der Tak, F. F. S., van Dishoeck, E. F., & Black,

- J. H. 2005, *A&A*, 432, 369
 Seifried, D., & Walch, S. 2016, *MNRAS*, 459, L11
 Shirley, Y. L. 2015, *PASP*, 127, 299
 Shu, F. H. 1977, *ApJ*, 214, 488
 Tassis, K., Willacy, K., Yorke, H. W., & Turner, N. J. 2012a, *ApJ*, 753, 29
 Tassis, K., Willacy, K., Yorke, H. W., & Turner, N. J. 2012b, *ApJ*, 754, 6
 Tritsis, A., Tassis, K., & Willacy, K. 2016, *MNRAS*, 458, 789
 Turk, M. J., Smith, B. D., Oishi, J. S., et al. 2011, *ApJS*, 192, 9
 van der Tak, F. F. S., Black, J. H., Schöier, F. L., Jansen, D. J., & van Dishoeck, E. F. 2007, *A&A*, 468, 627
 van Zadelhoff, G.-J., Dullemond, C. P., van der Tak, F. F. S., et al. 2002, *A&A*, 395, 373
 Walch, S., Girichidis, P., Naab, T., et al. 2015, *MNRAS*, 454, 238
 Yorke, H. W. 1986, *NATO Advanced Science Institutes (ASI) Series C*, 188, 141