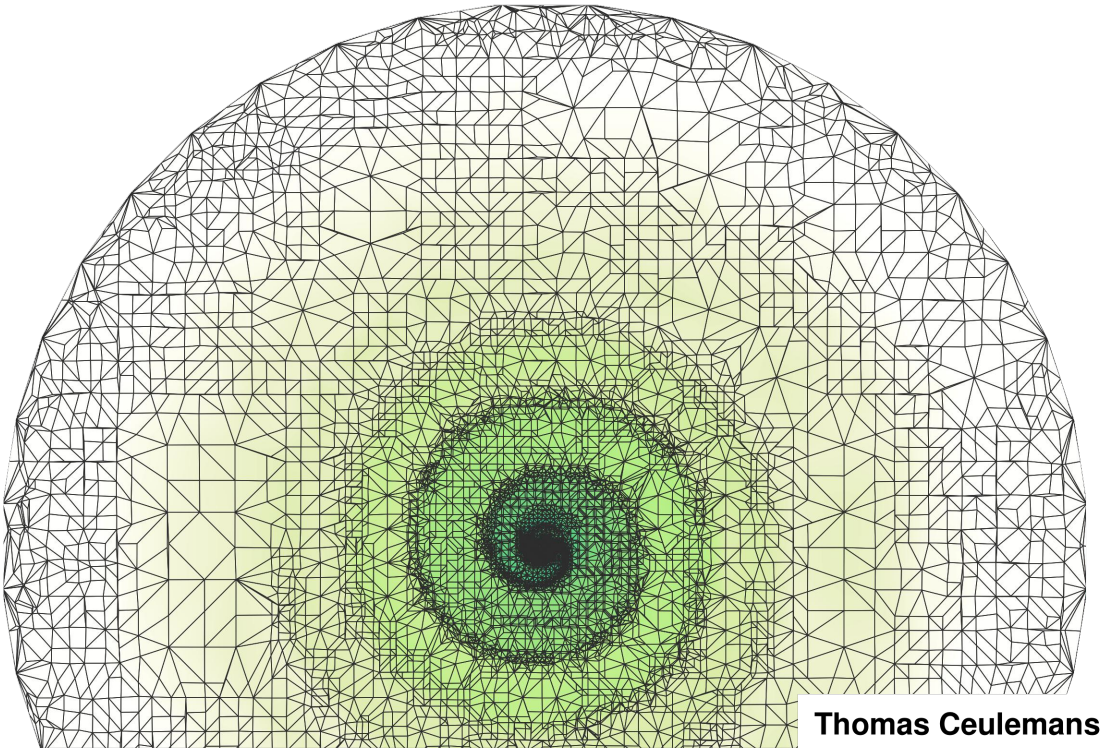
KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Science

Optimizing line radiative transfer to efficiently model the stellar environment

Synthetic observations and cooling of the stellar winds of AGB stars



Thomas Ceulemans

OPTIMIZING LINE RADIATIVE TRANSFER TO EFFICIENTLY MODEL THE STELLAR ENVIRONMENT

SYNTHETIC OBSERVATIONS AND COOLING OF THE STELLAR WINDS OF AGB STARS

Thomas CEULEMANS

Supervisors:
Prof. Leen Decin
Dr. Frederik De Ceuster

Members of the
Examination Committee:
Prof. Thomas Van Riet, chair
Prof. Jon Sundqvist
Prof. Giovanni Samaey
Prof. Maarten Baes
(UGent)
Prof. Jeremy Yates
(University College London)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Science (PhD): Astronomy and Astrophysics

September 2025

Preface

I started this PhD as continuation of my master thesis topic, which was about optimizing radiative transfer. Although the main work of this PhD is the same in concept, the scope is significantly different. Instead of solely focusing on the numerics of the computation, I now had specific goals we wanted to reach. In the research group of my supervisor Leen Decin, the goal is to better understand the origin of the chemical species in our universe. To do this, we have to simulate the chemistry in the outflow of a specific type of stars, namely AGB stars. For this, we evidently need chemistry calculations, but also require hydrodynamics and last but not least, radiative transfer. Therefore the practical goal of this thesis is to make detailed line transfer more computationally efficient[1]. For the science goals, we want to be able to compare observations of these AGB stars with simulated images, made from simulations of the AGB wind. Furthermore, the chemistry is impacted by temperature, and thus by radiative cooling.

First of all, I would like to thank my supervisors Leen and Frederik, who provided much needed advice and support, albeit each with slightly different focus. Leen focused more on the overarching context of my PhD, while Frederik was slightly more interested in the methods. You have both given me quite some freedom to explore what science I wanted to do, but when necessary, also given me direction I needed.

Secondly, I would like to the entirety of of team L.E.E.N., for the advice given during and outside of the team meetings, both on science-related topics, and non science topics. If there one last piece of advice I may give to you: remember to continue the tradition of bringing snacks to the team meeting.

Of course also colleagues of the IvS, for the generally pleasant work environment. As a relatively quiet person, I quite enjoyed hearing the conversations happening

[1] At this moment of the explanation, I feel that I should thank both of my supervisors, who more than once reminded me that I should not only focus on the (numerical) methods, but also on the science goal we aim to reach.

in the coffee room next door from my office, no matter how random they might seem to be at times. Even though I did not join many of the outside of work activities organized by you, I consistently joined the board game evenings, which I really liked.

Last but not least, I would like to thank my family for supporting me whenever I needed it throughout these 4 years, which has probably taken a bit longer in my perceptions of time than in yours. No matter how dry all this statement may seem to be: I truly appreciate all of you for being there during my PhD.

The work in this manuscript has received computational support in the form fo available computational resources and HPC training from the the Flemish Supercomputer Centrum (VSC) and their UK counterpart DIRAC. Finally, I acknowledge support from the Vlaams Wetenschappelijk fond (FWO), grants 166722N and 166724N for making this PhD possible.

Popularized Abstract

After stars come to their end, their remains can be recycled to create new stars and planets. In this way, the chemical composition of this material will be inherited by the new objects. Thus, to understand the chemical origins of the Earth and our Sun, we must figure out the chemical processes in which these chemical species are made in aging stars. An important fraction of the mass of these stars is lost in the so-called AGB phase, which low-to-intermediate mass stars (with initial mass of up to 8 solar masses) enter the end of their lives. In this phase, they expel their outer layers in so-called stellar winds, which can reach up to $\sim 15 - 30$ km/s. To simulate the chemistry, one needs the temperature, the pressure and the gas density. These parameters can be obtained from hydrodynamics simulations but are also influenced by radiation transport. Matter which emits light, loses energy and therefore cools. From the observational side, different colors of light are emitted and absorbed differently in the envelope, depending on its composition, which informs us about the presence/absence of specific molecules. Therefore, the spectrum of light contains a lot of information. Simulating line radiative transfer comes at a high computational cost. This is especially true when simulating light self-consistently on top of expansive 3D stellar winds models. As knowledge of the temperature is crucial in both chemistry and hydrodynamics, we must also be able to calculate the cooling induced by radiation. In this thesis, we explore various computational improvements to line radiative transfer, and implement them in the open-source 3D line radiative transfer library Magritte. We try optimizing model discretization and explore various improvements to the numerical implementation of radiative transfer. Furthermore, we explore a new numerical method to simulate radiative transfer, and finally also try out GPU computing, attempting to speed up our computations by utilizing more powerful compute hardware. Finally, we calculate radiative cooling rates for CO in the AGB environment, in order to improve the physics of AGB simulations.

Gepopulariseerde Samenvatting

Op het eind van hun leven, verliezen sterren het gros van hun material waaruit ze zijn opgebouwd. Deze overblijfselen vormen later nieuwe sterren en planeten. Op deze wijze hangt de samenstelling van de nieuwe objecten af van de oudere sterren. Bijgevolg, om de oorsprong van de chemische samenstelling van de aarde en de zon te achterhalen, moeten we chemie kunnen simuleren. Het blijkt dat het overgrote deel van het material afkomstig is van zogenaamde AGB-sterren, oude sterren met een initiële mass van tussen de 0,8 tot 8 zonsmassas. In deze fase verliezen ze hun massa in sterrenwinden, die snelheden tot $\sim 15 - 30$ km/u bereiken. Om de chemie te simuleren in de winden, moeten we de temperatuur, druk en dichtheid kennen. Deze parameters kunnen we afleiden uit hydrodynamica simulaties, maar worden ook beinvloed door stralingstransport. Bijvoorbeeld, wanneer materiaal licht uitstraalt, verliest het energie en dus koelt het af. Bovendien kunnen we ook veel informatie halen uit de observatie van licht, omdat elk atoom/molecule licht absorbeert en uitstraalt op unieke golflengten (∼kleur). De nodige berekeningstijd om lijnstralingtransport te simuleren, is hoog. Bovendien beinvloeden het stralingsveld en de energetische staat van het materiaal elkaar wederkerig, wat de simulaties nog duurder maken. Omdat de temperatuur een significante impact heeft op zowel chemie en hydrodynamica, is het belangrijk dat we stralingskoeling kunnen berekenen. In deze thesis onderzoeken we verscheidene optimalizaties voor het berekenen van lijn stralingstransport, en implementeren deze in de open-source lijn stralingstransfer code Magritte. We bekijken onder andere de model discretizatie en verscheidene verbeteringen met betrekking tot de numerieke implementatie van stralingstransport. Bovendien onderzoeken we een nieuwe numerieke methode om dit te simuleren. Verder bekijken we ook hoe we grafische kaarten kunnen inzetten in de berekening. Tot slot berekenen we de stralingskoeling voor het molecule CO in de AGB winden, om de simulaties

van de AGB winden te verbeteren.

Abstract

A significant fraction of gas and dust in the Interstellar medium (ISM) originates from the outflows of low to intermediate mass stars in the Asymptotic Giant Branch (AGB) phase. Therefore, the chemical composition of the ISM is closely related to the chemistry happening in the complex winds originating from these stars. This material can later on be recycled to form new stars and planets; the chemical species formed in these AGB winds therefore form the building blocks of our solar system. The chemistry depends on local conditions, such temperature, density and pressure, and is therefore influenced by thermodynamics. To understand how and where the chemical species are formed, we must simulate the AGB environment. From recent observations, the outflow shape of several AGB stars appears to be significantly different from spherical symmetry, showing arc-like structures, which might indicate the presence of a companion. In light of this, we can no longer treat AGB simulations in 1D.

Radiative transport is the main topic of this thesis, as light impacts the dynamics through radiative cooling and radiative pressure. Furthermore, light directly provides information about the physical and chemical processes in the outflow. Of particular importance to this thesis are line absorption and line emission. The presence of molecules/atoms influences the intensity of light in narrow frequency bands around their line frequencies, unique to each species. The particular shape of these line profiles are shaped by the density, temperature and velocity profile of the wind. Thus radiative transfer is necessary for both improving simulations and understanding the observations of these distant objects. Assuming that we can efficiently simulate radiative transfer for the AGB models, we can implement additional physics in our simulations and try to better interpret the observations.

However, simulating line radiative transfer on 3D models is computationally expensive, even more so if we simulate it in a self-consistent manner, given that the radiation field depends on the energy state of the medium and vice versa.

In this thesis, we will mainly be improving the non- Local Thermodynamic Equilibrium (NLTE) line radiative transfer code MAGRITTE, exploring many approaches to optimize our simulations. We improve the individual calculation steps of the radiative transfer simulations, and explore a potentially more computationally efficient formulation of radiative transport equation. We also rewrite our simulation code in an array-oriented paradigm, using PYTORCH, a machine learning library. This approach allows us to run our codes on various hardware, including GPUs, and allows for (linear) uncertainty quantification, given that models created using machine learning libraries are fully differentiable. We apply a similar approach in order to automate retrieval modeling, which consists of comparing synthetic observation of the modeled environment against observations of an astrophysical object to find the physical parameters of the object. Finally, we compute radiative CO cooling in the AGB environment, given that CO is abundant in this regime. However, given that NLTE line radiative transfer is still too expensive to simulate for including it in a hydrodynamics simulation, we propose a prescription for the net radiative CO cooling in the AGB outflow.

With the results of this thesis, we can now more efficiently simulate NLTE line radiative transfer. These tools can be utilized to include more radiative transfer physics in simulations of the AGB outflow. This will improve our understanding of the chemistry in this environment by allowing the simulations to better match reality, which brings us one step closer to understanding the origin of the chemical composition of the ISM, and by extension the origins of the building blocks of our solar system.

Beknopte Samenvatting

Een significant gedeelte van het het gas en stof in het interstellaire medium (ISM) vindt haar oorsprong in de omgeving rond licht-tot middelmatig zware sterren die zich in de AGB fase bevinden. Bijgevolg is de chemische samenstelling van het ISM gerelateerd aan de chemie die gebeurt in de turbulente sterrenwinden van de sterren. Het materiaal in het ISM kan later worden hergebruikt om nieuwe sterren en planeten te vormen; bijgevolg beinvloedt de chemie in deze sterrenwinden de samenstelling van ons zonnestelsel. De chemie wordt beinvloed door de lokale temperatuur, dichtheid en druk, en is dus afhankelijk van de hydrodynamica. Om te begrijpen waar en hoe bepaalde moleculen tot stand komen, moeten de AGB steromgeving simuleren. Uit recente observaties weten we dat de vorm van enkele van zulke sterrenwinden ver van sferisch symmetrisch is, en boogstructuren lijkt te bevatten, wat erop kan wijzen dat er een nabije buurster is. Bijgevolg kunnen we deze niet meer modelleren met 1D modellen.

Het hoofdonderwerp van deze thesis is stralingstransport, geven dat licht de dynamica beïnvloed door middel van stralingskoeling en stralingsdruk. Bovendien bevat licht informatie over en de fysica en de chemie die zich afspeelt in deze verre objecten. In deze thesis zijn voornamelijk lijn absorptie en lijn emissie van belang, omdat de aanwezigheid van atomen en moluculen de intensiteit van licht beïnvloedt in nauwe frequentiebanden rond bepaalde lijn frequenties, die uniek zijn voor elk soort atoom/molecule. De vorm van de intensiteitscurve rond de lijnfrequentie wordt bepaald door de temperatuursprofiel, de dichtheidsverdeling, en het snelheidsveld van de AGB wind. Bijgevolg is het simuleren van stralingstransport noodzakelijk om onze simulaties te verbeteren en observaties van deze AGB winden te kunnen begrijpen.

Helaas zijn deze stralingstransport simulaties computationeel duur, zeker in 3D. Als we bovendien het stralingstransport op een consistent manier willen berekenen, moeten we rekening houden met het feit dat het stralingsveld en de

energetische staat van het medium van elkaar afhangen.

In deze thesis zullen we voornamelijk de niet-LTE lijnstralingstransport code MAGRITTE verbeteren, op verscheidene manieren. We optimalizeren de individuele berekeningsstappen in de simulaties, en bekijken een potentieel efficientere formulering van de stralingstransport vergelijking. We herschrijven ook de algoritmes in onze code om berekeningen op grote vectoren van data uit te voeren, wat efficienter geëvalueerd kunnen worden op modernere computer architecturen. We maken hiervoor gebruik van PYTORCH, een functiepakket bedoelt om aan machine learning te doen. Dit laat ons toe om code te schrijven welke werkt op een variëteit aan hardware architecturen, inclusief grafische kaarten, en om een maat van onzekerheid te berekenen op onze bekomen resultaten. We gebruiken ditzelfde functiepakket ook om model reconstructie te automatiseren. Vanuit een astrofysische observatie proberen we de fysische staat van het medium te bepalen aan de hand van het vergelijken van synthetische observaties met de observatie. Tot slot berekenen we de stralingskoeling van CO in de AGB omgeving. Omdat het simuleren van NLTE line stralingstransport nog steeds te traag is om in onze AGB modellen te steken, maken we een fit voor de CO koeling in deze omgeving.

Met de resultaten uit deze thesis kunnen we nu efficienter NLTE stralingstransport simuleren. De tools ontwikkeld in deze thesis kunnen nu gebruikt worden om meer fysica in onze modellen voor de AGB winden te steken. Dit zal ons verstaan van de chemie in deze omgeving verbeteren, wat ons een stap dichter brengt bij het begrijpen van chemische samenstelling van het ISM, en bij extensie de oorspong van de bouwstenen van ons zonnestelsel.

List of Abbreviations

The following table contains all abbreviations that will be used throughout this thesis.

Table 1: The abbreviations used in this manuscript.

Abbreviation	Full
AGB	Asymptotic Giant Branch
ALI	Accelerated Lambda Iteration
ALMA	Atacama Large Millimeter/submillimeter Array
AMR	Adaptive Mesh Refinement
API	Application/programmer interface
CPU	Central processing unit
GPU	Graphics processing unit
LAMDA	Leiden Atomic and Molecular Database
LTE	Local thermodynamic equilibrium
NLTE	non- Local thermodynamic equilibrium
PINN	Physics informed neural network
SKA	Square Kilometer Array
SPH	Smoothed Particle Hydrodynamics
TPU	Tensor processing unit
UV	Ultra violet

List of Symbols

Table 3 contains most of the symbols used in this thesis. Symbols which are only used at a single place in this thesis, or are a variation on an already defined symbol (e.g. use an extra subscript), might be excluded from this list, depending on their importance in this thesis. Note that some symbols can have multiple different meanings depending on the context. As we require different subscripts in the different chapters, we denote chapter-specific notation in Table 2.

Table 2: Chapter specific notation. The notation in this table is different compared to Table 3 in the specified chapters.

Chapter	Symbol	Description
Chapter 3	χ_{ij}	Frequency integrated line opacity
Chapter 4	$*_{ij}$	Subscripts denoting frequency index i and spatial index j
Chapter 4	$*_{ul}$	Subscripts for denoting the line transition $u \rightarrow l$
Chapter 4	$*_{\mathrm{obs}}$	Subscript denoting observer frame variable

Table 3: Symbols used in this thesis

Symbol	Description
I	Specific monochromatic intensity
$\boldsymbol{x}$	Position
ν	Frequency
η	Total emissivity
χ	Total opacity
S	Source function
B	Planck distribution
h	Planck constant
c	Speed of light
k_b	Boltzmann constant
τ	Optical depth
$\Delta\tau$	Optical depth increment
I_{bdy}	Boundary intensity
ΔE	Energy difference
ϕ_{ij}	Line profile function for transition $i \to j$
$\delta\nu_{ij}$	Line profile width
ν_{ij}	Line center frequency
A_{ij}	Einstein coefficient for spontaneous emission
B_{ij}	Einstein coefficient for stimulated emission
B_{ji}	Einstein coefficient for absorption
C_{ij}	Collisional rate coefficient
T	Temperature
m_{spec}	Species mass
v_{turb}	Turbulent velocity
η_{ij}	Line emissivity
χ_{ij}	Line opacity
n_i	Energy level population
n_*	Ng-acceleration prediction for the converged n_i
$\boldsymbol{v}$	Velocity
$\hat{\boldsymbol{n}}$	Direction unit vector
g_i	Energy level multiplicity
J_{ij}	Mean line intensity
Ω	Solid angle
ρ	Number density
Λ	Operator used in Λ-iteration
Λ^*	Approximation to the Λ-operator, used in ALI
l_{ij}^*	Local contribution to the Λ-operator
Λ	Net cooling rate
$*_c$	Subscript for denoting co-moving frame variables
Δx	Position difference
$\Delta\nu$	Frequency spacing
$\Delta\nu$	Doppler shift
$\mathcal{L}$	Loss function
N_{QUANTITY}	Amount of the specified QUANTITY in a model

Contents

List of Figures

List of Tables

CHAPTER 1
Astronomical context

Planetary systems, including the solar system, are formed by dust and gas contracting in the interstellar medium (ISM). The chemical composition of our Sun and Earth is therefore seeded from the ISM. In the same way that the birth of new stars extracts material from the ISM, the death of stars deposits new material to the ISM. Depending on how a star evolves throughout its life, different chemical elements get produced. In the final stages of their lives, stars eject most material from their envelope, which ends up in the ISM. There, the material can be reused for forming a new generation of stars and planets.

During their lifetime, stars nucleosynthesize chemical elements in their core and envelope, when the temperature and density are sufficiently high. Depending on the mass of the star, heavier elements can be created. Most stars are low-to-intermediate mass stars, with a mass between 0.8-8 solar masses. At the end of their lives, they enter the so-called AGB (Asymptotic Giant Branch) phase. In this phase, they eject most of their mass in stellar winds. Furthermore, some reactions in the (unstable) layers of the star, create an excess an excess of neutrons, which allows heavier s-process elements to form through (slow) neutron capture. These AGB stars provide most of the Li, C and N in the universe and produce significant amounts of s-process elements, ranging from ^{38}Sr to ^{83}Bi. Furthermore, the physical conditions within their stellar winds allow for complex chemistry to occur [Van de Sande et al., 2019, Maes et al., 2023]. The created molecules and dust end up in the ISM, forming the building blocks to create new stars and planetary systems. In total, AGB stars provide $\sim 85\%$ of the gas and $\sim 35\%$ of the dust [Tielens, 2005] available in the ISM.

In this chapter, we will mainly be focusing on AGB star physics, given that their winds are an important source of molecules for the ISM. However, the other chapters of this thesis are about (computing) radiation transport. Some details about the radiation transport described in this thesis are specific to the

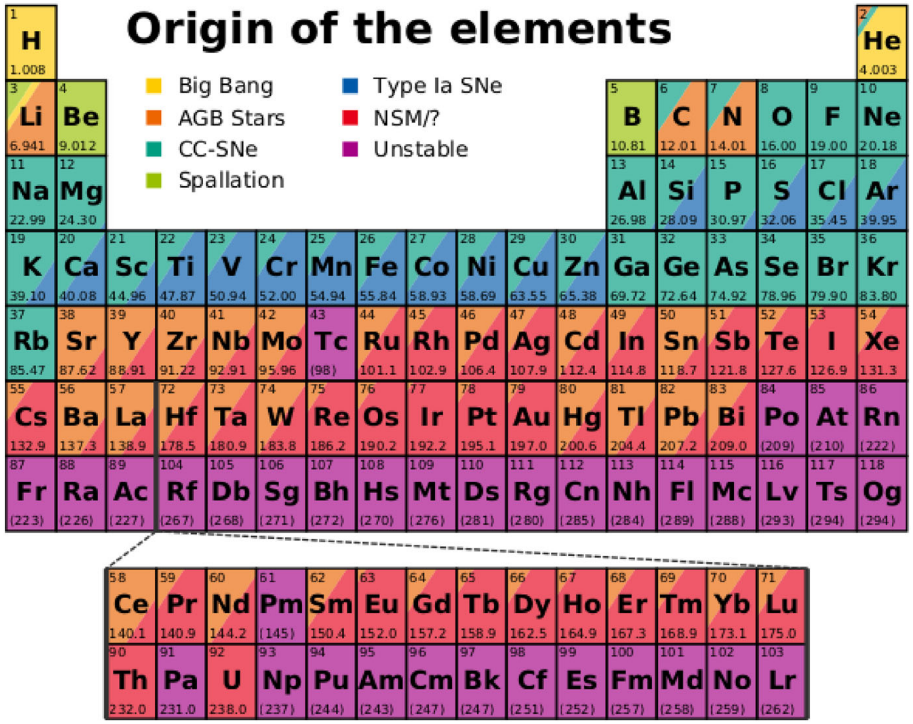


Figure 1.1: Periodic table of elements colored according to their origin. The colored area is proportional to fraction of the element originating from the different astrophysical origins. Figure taken from Arcones and Thielemann [2022].

AGB enviroment, but the majority of the content in this thesis is also applicable to different astrophysical regimes. For this introductory chapter, we first start by explaining how low-to-intermediate mass stars reach the AGB phase in Section 1.1, before explaining the characteristics of the phase itself in Section 1.2. Afterwards, we briefly touch upon the relevant physics for understanding the dynamics of the stellar wind in Section 1.3.

Stellar evolution 1.1

This section is based on standard textbooks [Aerts, 2021, Kippenhahn et al., 2012]. Before we explain the physics concerning Asymptotic Giant Branch (AGB) stars, we first explain how a star ends up in this phase. We will limit

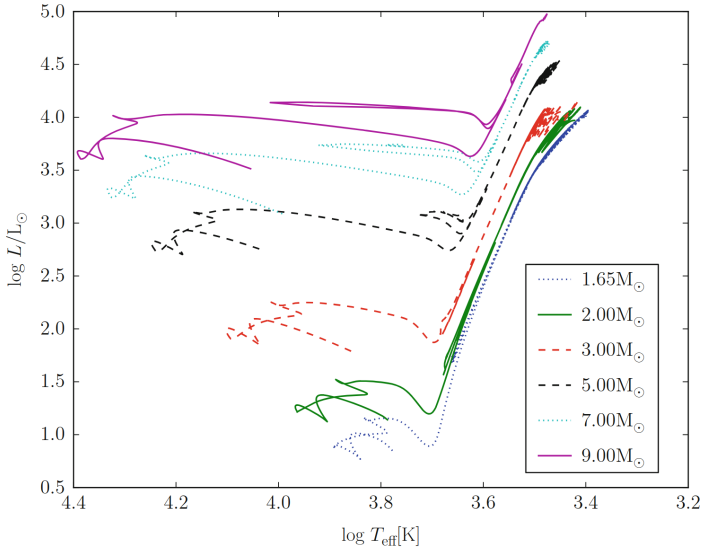


Figure 1.2: Hertzsprung-Russel diagram for various initial masses, using a solar-like metallicity of $Z = 0.02$. Each evolution track starts in the pre-main sequence, and ends in the AGB phase. Figure taken from Fig. 8.1 of Oswalt and Barstow [2013].

the scope of this chapter to low-to-intermediate mass stars, which eventually evolve into the AGB phase.

A protostar is born from a collapsing cloud of dust and gas. When the cloud contracts, the density, temperature and luminosity increase. Eventually the temperature in the core becomes hot enough for hydrogen burning to start, dramatically increasing the emitted radiation, enabling the radiation pressure to counteract gravity, thereby forming a stable star. Stars with insufficient mass, below $0.08M_\odot$, never reach hydrogen burning and end up as a brown dwarf instead. The start of hydrogen burning puts the star on the main sequence of the Hertzsprung-Russel diagram (see Fig. 1.2). For 90% of its life, it will remain in this phase, until most of the hydrogen in its core is exhausted. Afterwards, the hydrogen burning produces too little energy, while the temperature in the core is too low to initiate helium burning. Therefore, the star contracts due to gravity. Slowly, the temperature in the hydrogen shell rises, and eventually becomes high enough for hydrogen burning to start, which deposits extra helium in the core. This marks the beginning of the sub-giant phase. Because of the shell hydrogen burning, the envelope expands due to the increased luminosity,

moving the star along the red giant branch. At the same time, the core keeps contracting, as the insufficient amount of hydrogen burning no longer generates enough radiation pressure to counteract gravity. Eventually, the core becomes hot enough to enable helium burning, which signals the end of the red giant phase. For stars below $\sim 2.3 M_\odot$, the matter in the core has become degenerate, and helium burning occurs without expanding the radius of the core. Therefore a thermal runaway occurs, creating the so-called helium flash. Afterwards, the star enters the horizontal branch phase, in which stable helium burning occurs, together with hydrogen burning in the hydrogen shell. In the end, most matter in the core will be converted to carbon and oxygen, but the star will never reach the temperatures required to initiate carbon burning. The star now enters the AGB phase.

AGB stars 1.2

Although the core no longer participates in nucleosynthesis, temperatures in the envelope can become high enough to initiate hydrogen and helium burning at different radii. This is an unstable process, as the helium burning provides energy for the upper layers, which causes them to expand, and therefore cools the upper layers. This temporarily deactivates the hydrogen shell burning, which results in less helium being available for the lower layers, which eventually halts the helium burning. At this point the upper layers may contract again, restarting the hydrogen burning. Thus the cycle can begin again, until not enough matter remains for it to restart. Most of the envelope is then blown away, and what remains is a CO core, which signals the start of the post-AGB phase.

These pulsations in the AGB phase launch strong winds, with terminal velocities up to 30 km/s [Ramstedt et al., 2008], resulting in additional mass loss for the AGB star, with rates ranging from $\sim 10^{-8}$ to $\sim 10^{-5} M_\odot$/yr [Kippenhahn et al., 2012]. In these winds, different dynamical and chemical processes take place, depending on the distance to the star (as illustrated in Fig. 1.3). The winds start in the outer atmosphere of the star. In this region, chemistry can take place, as the temperature is low enough to prevent molecules immediately dissociating after formation (more in Section 1.3.1). Eventually the material reaches a cooler region, below 1500 K, in which dust can condense [Gail and Sedlmayr, 2014]. This dust then gets accelerated by radiation pressure, dragging along the gas [see e.g. Höfner and Olofsson, 2018]. Eventually, we get a steady outflow[1], in which the density drops, such that high-energy UV-photons can

[1]Gravitational force falls off as r^{-2} and radiative pressure falls off as $\rho r^{-2} \sim r^{-4}$, so almost no external forces act on the outflow in the outer regions.

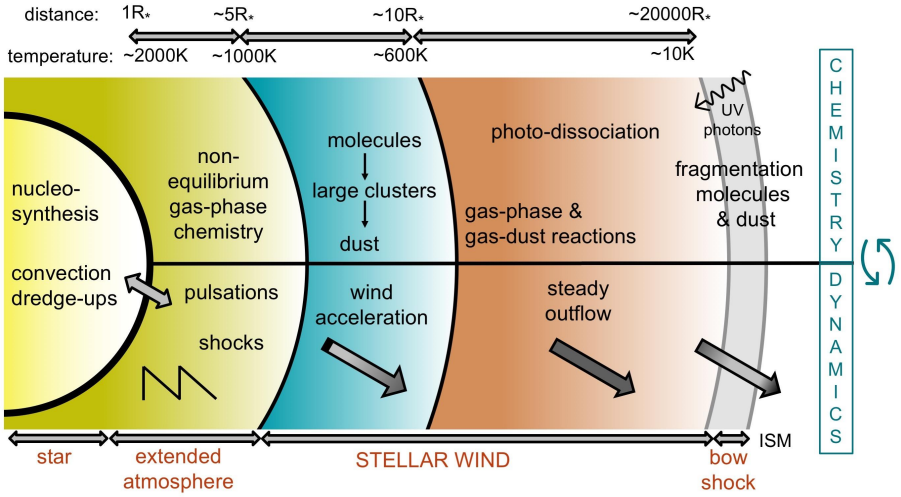


Figure 1.3: Schematic representation of the physics and chemistry in a 1D stellar wind originating from an AGB star. The puffed-up AGB star has a stellar radius $R*$ of about 1AU. Figure taken from Decin [2020]

penetrate and cause photo-dissociation.

Furthermore, some AGB outflows are far from symmetric (see Fig. 1.4). Because of the spiral-like structures observed, it is therefore hypothesised that some of these stars might have a binary companion, which interacts with the outflow to create intricate shapes [see Malfait et al., 2021, Maes et al., 2021].

Physics and chemistry in stellar winds 1.3

This section contains a brief overview of the physics responsible for the shape and contents of the AGB outflows (possibly impacted by a binary companion).

For modeling the physics concerning the outflow an AGB star, we need 3 main components: chemistry, hydrodynamics, and radiative transfer. Chemistry decides which molecules can form in the stellar wind (see Section 1.3.1). Hydrodynamics shapes the stellar wind and provides densities and temperatures for the chemistry, as explained in Section 1.3.2. Last but not least, radiative transfer impacts the pressure and temperature, through radiative pressure and radiative cooling, respectively (refer to Section 1.3.3). It also provides a means to compare our simulations to observations, as we can use it to create synthetic

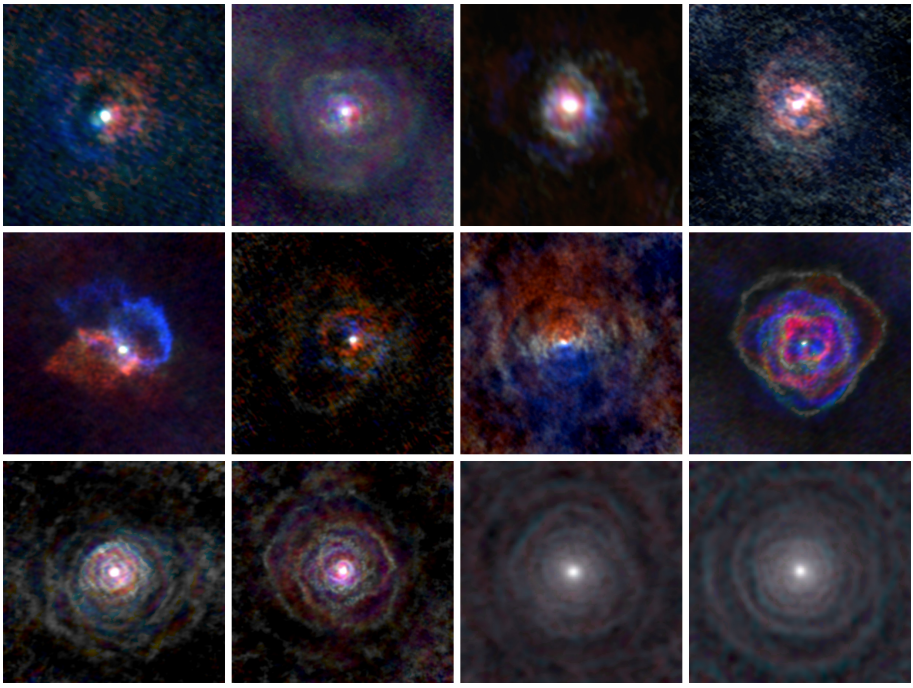


Figure 1.4: Non-symmetric outflows of AGB stars, as observed by the Atomium ALMA large program. Blue and red correspond respectively to material moving towards and away from the observer. © L. Decin, ESO, ALMA

observations from hydrodynamics models augmented with a description of the chemistry.

Chemistry, dust and gas 1.3.1

Depending on the carbon to oxygen ratio, C/O, of the star, different chemical reactions may take place in the stellar wind. This is due to the least abundant element being eventually fully depleted forming CO, and therefore almost not available for other reactions. In case that the C/O ratio is close to one, both carbon and oxygen remain available in more significant amounts. CO is the most common molecule in the stellar wind, except for H_2. For O-rich stars, assuming chemical equilibrium, models predict H_2O to be the (third) most abundant molecule, while in C-rich stars C_2H_2, HCN are expected. However, from observations, we see that O-bearing molecules (such as H_2O) have been detected in C-rich stars and vice versa [Agúndez et al., 2020]. As illustrated in Fig. 1.3, the assumption of chemical equilibrium is not valid in the region just outside the AGB star [Maes et al., 2023]. Therefore, the chemical composition of the wind varies throughout the stellar envelope. As the stellar wind travels outwards, the physical conditions, such as temperature and density, vary. This causes different chemical reactions to take place at varying distance from the star. In cooler regions, further away from the star, starting at a few stellar radii, the temperature drops such that dust grains can start to condensate [Höfner and Olofsson, 2018]. The dust absorbs light at infrared wavelength, accelerating outwards because of radiation pressure, dragging gas along with it. Because of collisions with dust, the gas also gets accelerated. Even further away, the dust density decreases, such that UV photons can penetrate and interact with the dust grains, causing photo-dissociation. A more detailed description of the chemistry around AGB stars can be found in Maes [2024].

Hydrodynamics 1.3.2

In absence of a companion object, the stellar wind structure is approximately spherically symmetric. Deviations from spherical symmetry can arise, for example due to clumping, or the influence of magnetic fields. The physical properties of the wind are determined by conservation laws: the conservation of mass, momentum and energy, determine the density, velocity and internal energy of the gas. Thus by simulating hydrodynamics, we obtain the densities[2] and temperatures[3], needed for the chemistry modeling.

[2]Technically, you will need to assume initial abundance ratios for the different chemical species, and treat the input density as total density of all the chemical species.

[3]Which is related to the change in internal energy, through for example adiabatic cooling.

However, we have seen in Fig. 1.3 that the outflows originating from some AGB stars are not at all spherically symmetric. In this case, the presence of a binary companion can explain the correlated density structures. Depending on the orbital parameters (eccentricity, period, and semi-major axis) of a binary system, many structures can appear, ranging from almost unperturbed spherically symmetric outflows to spiral-shaped outflows, arcs and smaller scale accretion discs around the companion [Maes et al., 2021, Malfait et al., 2021, Malfait, 2024].

Radiative transfer 1.3.3

Light contains a wealth of information about the environment from which it originates. Many different observables exist: for example interferometry provides precise information about the location of an object in the plane of the sky, photometry offers brightness data in various spectral bands, aiding in the determination of an object's temperature and distance. Furthermore, polarization provides insights, for instance, into the surrounding magnetic field of an object, as it reflects the scattering of light by polarized dust grains. Finally, spectral lines are sharp features in the spectral energy distribution of radiation. They originate from quantized energy transitions in the atoms and molecules in the medium. The frequencies at which these lines appear are specific to the chemical species that produce them, and thus can be used to deduce their origin in astrophysical observations. As physical experiments are not feasible in astronomy due to the scales involved, all theory must be verified using observations. Therefore it is not exaggerated to state that most of astronomy theory is built on the detection of light.[4]

The paragraph above may lead to the impression that light is merely the messenger of the physical and chemical processes in the star. However, it plays an important role in the energy and momentum transport. For instance, radiation pressure can shape the dynamics of stellar atmospheres, as it can allow a stellar outflow to form. In AGB stars, infrared radiation on dust grains launches the wind [Höfner and Olofsson, 2018]. In contrast, in the hotter, higher mass stars, metal lines[5] absorb UV photons originating from the star [Sander et al., 2017], also launching an outflow. Radiative cooling has a significant impact on the physical properties of the AGB outflow. For example, in Malfait et al. [2024b], the inclusion of atomic H I atomic cooling in their AGB binary hydrodynamics simulations significantly reduced the temperatures in the hot regions around the binary companion. As a consequence, an accretion disk can

[4] For this thesis, we ignore gravitational waves and high-energy particles, as these are not significantly present in the AGB stage.

[5] In astronomy, metals are defined as: all elements heavier than helium.

form around the companion, which was not the case in Malfait et al. [2021]. Finally, some chemical reactions are influenced by high-energy photons. For example, UV-photons can dissociate molecules in the AGB wind [Khouri et al., 2014] or interact with the dust grains [Van de Sande et al., 2020]. In contrast to the ISM, the material in the inner parts of the wind is shielded from UV-photons. Thus radiative transfer is required to evaluate the intensity of the high-energy photons.

In hydrodynamics, the cooling prescription used can have significant impact on the resulting geometry [see e.g. Malfait et al., 2024b]. The cooling contributions can originate from either hydrodynamics or radiative transfer. Radiative cooling arises from a net emission of radiation by the chemical species and dust of the environment. In the AGB outflow, CO can provide a significant amount of cooling, as it the second most abundant molecule in that regime. Although H_2 is the most abundant molecule, it has no dipole moment. As a consequence, it has no rotational energy transitions. Thus the temperature regime in which H_2 cooling is relevant starts therefore significantly higher, at approximately ~ 6000 K[6]. In this thesis, we will therefore aim to calculate radiative cooling rates for CO, as currently, no CO cooling rate prescription has been proposed for AGB environment. We do note however that several CO cooling prescriptions exist for the ISM [see e.g. Neufeld and Melnick, 1991, Whitworth and Jaffa, 2018].

In this chapter, we explained the astronomical context of this thesis. In short, low-mass stars produce significant parts of the dust and gas in the ISM, from which new stars and planets are formed. These low-mass stars contribute this material to ISM during the AGB phase, at the end of their lives. At this stage, the stars lose a significant amount of mass in their stellar wind (up to $10^{-5} M_{\odot}/\mathrm{yr}$). These winds are characterized by complex non-equilibrium chemistry, the products of which end up in the ISM. For accurately computing the chemistry occurring in these winds, we need temperatures and densities from hydrodynamics, augmented with heating and cooling contributions from radiative transfer. One promising candidate, which has not yet been explored in the AGB environment, is radiative cooling from CO. Furthermore, to compare any hydrodynamics model with an astronomical observation, we need to simulate radiation transport inside the modeled object. Thus, starting from the next chapter, we explain the physics concerning radiation transport, and how to numerically solve the corresponding equations.

[6] This estimate uses the energy difference corresponding to the 1-0 S(0) vibrational transition of H_2.

CHAPTER 2

Introduction

In this chapter, we will provide all theory required to simulate self-consistent 3D line radiative transfer, working towards calculating radiative cooling rates in 3D hydrodynamics models. We start from the radiative transfer theory in Section 2.1, and then explore the numerical implementation in Section 2.2. As radiative transfer is computationally expensive, we explain the standard literature optimizations in Section 2.3. Finally, in Section 2.4, we explain the features of the open-source radiative transfer code MAGRITTE, which I have further developed during my PhD.

Radiative transfer theory 2.1

Prior to explaining how to numerically compute radiative transfer, we will first explore the theory behind it. We start by explaining basic radiative transfer theory in Section 2.1.1. Afterwards in Section 2.1.2, we specialize the theory towards spectral line features, originating from atomic/molecular energy transitions. Velocity differences (see Section 2.1.3) further complicate the radiation field, as they separate the line features in frequency space, due to Doppler shifts. In general, the energy state of atoms and molecules depends on the local radiation field, which in turn depends on the energy state atoms and molecules in the environment. Thus we will explore self-consistent line radiative transfer in Section 2.1.4. We take a brief detour to explore another, potentially more computationally efficient formulation of the radiative transfer equation in Section 2.1.5. Finally, as we have described all prerequisite knowledge, we can move our attention to applications of radiative transfer. We explain how synthetic observations are used to connect theory to observation in Section 2.1.6. Afterwards, we describe how radiative cooling works in Section 2.1.8.

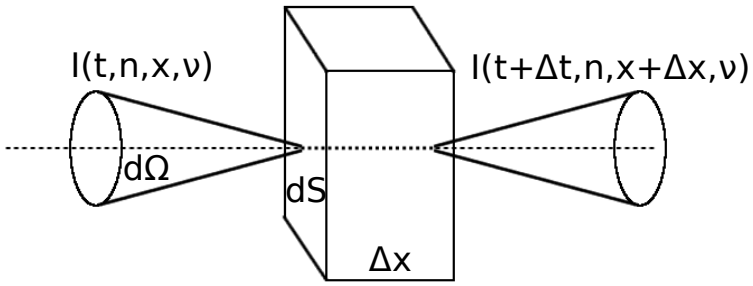


Figure 2.1: Illustration of the specific monochromatic intensity I, passing through a material which can absorb and emit radiation. Figure adapted from Fig. 11.1 of Hubený and Mihalas [2014].

Basic equations of radiative transfer 2.1.1

Light travels in a straight line, increasing and decreasing in intensity depending on the properties of the local medium. When light passes through a region, a part of the photons is absorbed by the material inside that region. Conversely, the material can also emit photons, increasing the intensity of the light.

Assuming no relativistic effects are present, the energy transport by radiation is described by the following equation [Hubený and Mihalas, 2014]

$$\frac{\partial I}{\partial t} + \hat{\boldsymbol{n}} \cdot \nabla I(t, \hat{\boldsymbol{n}}, \boldsymbol{x}, \nu) = \eta(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu) - \chi(\boldsymbol{x}, \nu) I(t, \hat{\boldsymbol{n}}, \boldsymbol{x}, \nu) \tag{2.1}$$

in which I[W sr^{-1} m^{-2} Hz^{-1}] denotes the specific monochromatic intensity, $\boldsymbol{x}$[m] represents the position, ν[Hz] denotes the frequency, η[W sr^{-1} m^{-3} Hz^{-1}] represents the total emissivity and χ[m^{-1}] corresponds to the total opacity. In radiative transfer, many related, but slightly different quantities can be denoted by the same terminology, which can cause confusion in notation. With the specific monochromatic intensity, we mean the energy transported per unit of time, which passes through a unit surface, for photons of a specific frequency originating from a given angle (see Fig. 2.1). As the emissivity η can be similarly confusing, we define it to be the energy emitted per unit time, in a given direction, per unit volume, per photon frequency. Finally, the opacity is the fraction of light absorbed per unit distance.

The timescale which light needs to travel throughout the AGB envelope is significantly shorter than the timescale in which the AGB environment changes

significantly. Therefore, we can assume a steady state for the radiation field, reducing Eq. 2.1 to

$$\hat{\boldsymbol{n}} \cdot \nabla I(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu) = \eta(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu) - \chi(\boldsymbol{x}, \nu) I(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu). \tag{2.2}$$

In general, the emissivity and opacity terms can consist of slowly varying continuum contributions (such as black body radiation), scattering, which absorbs and re-emits light in a different direction, based on some correlation function and line contributions, the latter of which we will explain in Section 2.1.2. The scattering of light in the AGB environment is mainly due to dust. Given that we do not simulate dust formation in our current AGB hydrodynamics models, we do also not implement scattering in our radiative transfer simulations.

In the simplest case, one can assume the medium to be in local thermodynamic equilibrium (LTE). The source function S[W sr^{-1} m^{-2} Hz^{-1}], is defined as

$$S(\boldsymbol{x}, \nu) = \frac{\eta(\boldsymbol{x}, \nu)}{\chi(\boldsymbol{x}, \nu)}, \tag{2.3}$$

is given by the Planck distribution $B(T, \nu)$

$$B(T, \nu) = \frac{2h\nu^3}{c^2} \frac{1}{\exp(h\nu / k_b T) - 1}, \tag{2.4}$$

in which h is the Planck constant, c is the speed of light and k_b is the Boltzmann constant.

If one knows the source function S, the intensity can be derived using the formal solution of the time-independent 1D radiative transfer equation without scattering [see e.g. Kunasz and Auer, 1988]

$$I(x, \nu) = I_{\mathrm{bdy}}(\nu) e^{-\tau(x,\nu)} + \int_0^{\tau(x,\nu)} S(t, \nu) e^{t - \tau(x,\nu)} dt, \tag{2.5}$$

in which I_{bdy}[W sr^{-1} m^{-2} Hz^{-1}] is the boundary intensity and the optical depth τ[.] is defined as

$$\tau(x, \nu) = \int_0^x \chi(s, \nu) ds. \tag{2.6}$$

To gain more intuition in how equation 2.5 determines the intensity, we give a simple example, using a uniform medium, with constant source function S for all positions x and zero boundary intensity. The equation then simplifies to

$$I(x, \nu) = S(\nu) \left(1 - e^{-\tau(x,\nu)}\right) \tag{2.7}$$

Filling this in, we see that the source function S act as an upper bound for the computed intensity. Furthermore, the optical depth τ how close the intensity becomes to the source function. In optically thin regimes ($\tau << 1$), the intensity scales linearly with the optical depth τ, $I \sim S\tau$. In optically thick regimes ($\tau > 1$), the intensity is very close to the source function, $I \simeq S$.

Line radiative transfer 2.1.2

Lines are narrow intensity features in frequency space arising from transitions between quantized electronic, vibrational, and rotational states of atoms and molecules in the medium. In order to change the energy state of a molecule or atom (as illustrated in Fig. 2.2), a photon will be absorbed or emitted, which results in higher or lower energy state respectively. The frequency ν of the photon, is given by Planck's law:

$$\Delta E = h\nu, \tag{2.8}$$

in which ΔE[J] is the energy difference and $h = 6.63 \cdot 10^{-34}$J Hz^{-1} is the Planck constant. In quantum mechanics, the possible states of a molecule or atom are discrete. Therefore the corresponding energy levels and transitions between them are also discrete. This implies in turn that the frequencies of emitted and absorbed photons should also be quantized.

If an object has a non-zero velocity with respect to an observer, the photon frequency in the observer frame of reference will be different from their frequency in the reference frame comoving with the emitting material. This phenomenon is called Doppler shifting (see also Section 2.1.3). Because of thermal motions, matter at any given position has a range of different velocities. Therefore, an observer will see a wider band of frequencies originating from a single line transition than just a single frequency. Another effect which can broaden the line profile is called pressure broadening. Collisions with nearby particles can ever so slightly perturb the energy state of molecules, which in turn impacts the energy difference between energy levels and thus can broaden the line shape. Finally, the Heisenberg uncertainty principle also predicts a variation in the energy differences between level transitions, given that the time when a energy transition happens is not deterministic. However, this last term is negligible in all circumstances except in the coldest astrophysical regimes [Tennyson et al., 2014].

In line radiative transfer, we must use a distribution for the frequencies at which light can be absorbed and emitted, which we call the line profile function ϕ. Depending on which broadening effects are included, its formula gets more complicated [see Tennyson et al., 2014, for a comprehensive overview].

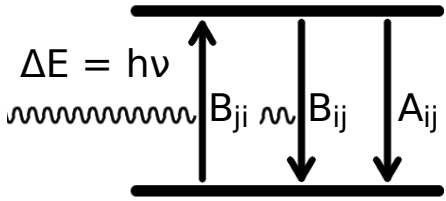


Figure 2.2: Energy level transitions occur when absorbing/emitting a photon at the correct frequency ν, corresponding to the energy difference ΔE. The transition rates are governed by their respective Einstein coefficients.

Commonly used ones in astronomy are the Doppler/Gaussian profile, which only considers turbulent broadening, and the Voigt profile, which also includes pressure broadening. For this thesis, we stick to the simplest formulation, as pressure broadening can be ignored in the AGB winds. The Gaussian line profile function ϕ_{ij} for a line transition $i \to j$ used throughout this thesis is given by

$$\phi_{ij}(x,\nu) = \frac{1}{\delta\nu_{ij}(x)\sqrt{\pi}} \exp\left[-\frac{(\nu-\nu_{ij})^2}{\delta\nu_{ij}(x)^2}\right]. \tag{2.9}$$

The line profile width $\delta\nu_{ij}$ arises from the combined effects of thermal and turbulent motions in the medium. It is given by

$$\delta\nu_{ij}(x) = \frac{\nu_{ij}}{c}\sqrt{\frac{2k_{\mathrm{B}}T(x)}{m_{\mathrm{spec}}} + v_{\mathrm{turb}}^2(x)}, \tag{2.10}$$

where $k_{\mathrm{B}}[\mathrm{J\ K^{-1}}]$ represents Boltzmann's constant, $T[\mathrm{K}]$ is the local gas temperature, $m_{\mathrm{spec}}[\mathrm{kg}]$ is the gas species mass, and $v_{\mathrm{turb}}[\mathrm{m\ s^{-1}}]$ is the turbulent velocity. The last parameter signifies the turbulence which is not captured by the used numerical discretization.

Given the line profile function ϕ_{ij}, we can now define the emissivity η_{ij} and opacity χ_{ij} for a single line transition ij of a species (where i denotes the upper energy level and j denotes the lower energy level), which depend on the populations of the energy levels $n_i(x)[\mathrm{m^{-3}}]$. They are given by

$$\eta_{ij}(x,\nu) = \frac{h\nu}{4\pi} n_i(x) A_{ij} \phi_{ij}(x,\nu), \tag{2.11}$$

$$\chi_{ij}(x,\nu) = \frac{h\nu}{4\pi} \left(n_j(x)B_{ji} - n_i(x)B_{ij}\right) \phi_{ij}(x,\nu), \tag{2.12}$$

in which A_{ij}, B_{ij}, B_{ji} are the Einstein coefficients, and respectively denote the spontaneous emission, the induced emission and the photon absorption rates.

The total emissivity η and opacity χ can be obtained by summing all individual contributions.

Note that we have assumed the line profile functions ϕ_{ij} of the emissivity and opacity to have the exact same shape. For this, we have assumed complete frequency redistribution; i.e. the frequency of emitted photons being unrelated to the frequency of absorbed photons. This significantly simplifies solving the resulting equations. It is possible to drop this assumption, see e.g. Paletou and Peymirat [2021].

The more observant readers have noticed that Eq. 2.12 implies that the line opacity χ_{ij} can become negative, if the net induced emission rate $n_i B_{ij}$ is higher than the net absorption rate $n_j B_{ji}$. When this happens, we have a maser[1]. This happens when the upper level n_i of the line transition $i \rightarrow j$ is overpopulated. The optical depth τ becomes negative, therefore the intensity I increases exponentially, see the first term of equation 2.5. Masers have been observed for various molecular species, including H_2O [Nesterenok, 2013], SiO [Yun and Park, 2012], OH [Field et al., 1994] and CO [McKee et al., 1982]. Although one can model masers [see e.g. Yates et al., 1997, Gray et al., 2018], an accurate treatment requires a significantly dense spatial and angular resolution[2]. Given the additional computational constraints, we will not be considering masers in this thesis. Aside from this, (near) maser-like conditions can also cause numerical instabilities, as opacities χ nearing zero can cause large variations in the source function $S = \eta/\chi$, and in turn can result in a poorly computed intensities. To prevent such instabilities, we impose a minimum value for the frequency-integrated line opacity χ_{ij}/ϕ_{ij} and implement interpolation, as explained in Section 3.4.5.

Doppler shifts 2.1.3

Light can be blue- or red-shifted, depending on whether the source of the light is moving towards or away from us. If we have an observer, looking in direction $\hat{\boldsymbol{n}}$ and we define $\boldsymbol{v}$ to be the velocity of the observer relative to the emitting object, then the observed frequency of light ν is related to the emitted frequency

[1] Which stands for Microwave Amplification by Stimulated Emission of Radiation. When a photon with the correct frequency hits a material which has a population inversion, more photons will be released at that frequency. This effect can cascade, generating a bright light.

[2] In order to see a significant increase in intensity, material in a large region in a straight line towards the observer must have the same velocity towards the observer. Otherwise, Doppler shifts will result in a limited negative optical depth.

ν_0 in the following manner,

$$\nu = \nu_0 \left(1 + \frac{\hat{\boldsymbol{n}} \cdot \boldsymbol{v}}{c}\right). \tag{2.13}$$

In this equation, we have assumed non-relativistic velocities. In the AGB outflows, which is a relatively slow environment[3], the winds have terminal velocities on the order of 10 km/s. This results in a change in frequency due to Doppler shifts on the order of $(\nu - \nu_0)/\nu_0 \simeq 10^{-5}$. This seems insignificant, until we realize that emission and absorption lines may have a relative line widths $\delta\nu_{ij}/\nu_{ij}$ on the order[4] of 10^{-6}. Therefore, when calculating line radiative transfer over an AGB model, the frequency of light emitted by a line transition may be shifted over multiple line widths when travelling through the stellar wind.

NLTE 2.1.4

The emissivities and opacities, as shown in equations (2.11) and (2.12), depend on the level populations n_i. In the assumption of Local Thermodynamic Equilibrium (LTE), these are fully determined by the local gas temperature using the Boltzmann distribution:

$$\frac{n_i}{n_j} = \frac{g_i}{g_j} \exp\left[-\frac{\Delta E}{k_b T}\right], \tag{2.14}$$

in which g_i is the multiplicity of the corresponding energy level, as different quantum configurations can lead to the same energy state. Finally, ΔE is the energy difference. However, this assumption is only valid in case of high densities, and high optical depths, such that the energy level distribution is fully determined by particle collisions. In this case, the radiation field is fully described by Planck distribution $B(T, \nu)$. In absence of this assumption, we have non- Local Thermodynamic Equilibrium (NLTE), in which the level populations n_i depend on the local radiation field, which can no longer be assumed to be given by $B(T, \nu)$. The relation between the radiation field and the level

[3] In the sense that the velocity is many orders of magnitude lower than the speed of light, and thus we do not need to take into account any relativistic effects.

[4] Taking the CO molecule as an example, and filling in a temperature of 500K in Eq. 4.19, we get $\delta\nu_{ij}/\nu_{ij} \simeq 4 \cdot 10^{-6}$, when assuming $v_{\text{turb}} = 0$.

populations is given by the statistical equilibrium equations:

$$\begin{aligned}\frac{\partial n_i}{\partial t} =& \sum_{j<i} \left(n_i A_{ij} - (n_j B_{ji} - n_i B_{ij}) J_{ij}\right) \\ &- \sum_{j>i} \left(n_j A_{ji} - (n_i B_{ij} - n_j B_{ji}) J_{ij}\right) \\ &+ \sum_{j=1}^{N} \left(n_i C_{ij} - n_j C_{ji}\right), \end{aligned} \tag{2.15}$$

where N is the number of energy levels considered, $C_{ij}[\mathrm{s}^{-1}]$ denotes the temperature-dependent collisional rates and J_{ij} is the mean line intensity corresponding to line transition ij:

$$J_{ij}(\boldsymbol{x}) = \frac{1}{4\pi} \oint_{\Omega} \int_{0}^{+\infty} I(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu) \phi_{ij}(\nu) \mathrm{d}\Omega \mathrm{d}\nu, \tag{2.16}$$

which represents a weighted average of the intensity over the frequency ν[Hz], integrated across the solid angle Ω[sr]. To solve the statistical equilibrium equations (Eq. 2.15), we assume a steady state solution. This is valid for AGB outflows, as the timescale which light needs to travel throughout the AGB outflow ($\sim$ 10 mins per AU), is significantly shorter than shortest relevant hydrodynamics timescale, which for AGB stars is the pulsation period (order of 100 days). Note that this argument does not hold for more explosive events such as supernovae. We can therefore assume

$$\frac{\partial n_i}{\partial t} = 0. \tag{2.17}$$

As the resulting system of equations is linearly dependent, we replace the last equation by the physical requirement that the sum of all level populations n_i must be equal to the species' abundance $\rho[\mathrm{m}^{-3}]$. The equation for the last level population in Eq. 2.15 is therefore replaced by

$$\sum_{i=1}^{N} n_i(\boldsymbol{x}) = \rho(\boldsymbol{x}). \tag{2.18}$$

in which $\rho[m^{-3}]$ is the number density of the atomic/molecular species.

From these equations, we see that the level populations depend on the radiation field, which itself depends on the level populations n_i. Thus to obtain a self-consistent radiation field, one can solve these equations iteratively, obtaining an iterative procedure called Λ-iteration. Λ-iteration starts by assuming an initial

distribution for the level populations n_i, for example assuming LTE. Then we compute the intensity field, and use the statistical equilibrium equation to obtain new values for the level populations n_i. This loop repeats until the level populations converge. Note that this Λ-iteration procedure converges slowly [van Zadelhoff et al., 2002]. A numerical improvement to this method can be found in Section 2.3.1.

Comoving frame radiative transfer 2.1.5

In stellar outflows, non-zero velocity gradients are present. Therefore these reference frames are shifted in frequency space with respect to each other. Gluing them together for an entire astrophysical environment, we obtain a non-inertial frame, called the comoving frame. We thus need to evaluate emissivities and opacities in this frame. For this, we can Doppler shift the frequency at which we evaluate these quantities and apply appropriate transformations [See Appendix B of Hubený and Mihalas, 2014] to be able to evaluate them in the inertial observer frame. Note that in a non-relativistic scenario, we can limit ourselves to only transforming the evaluation frequency, as the relative change in the other quantities (I, η and χ) depend on v/c. Another option is to write the standard radiative transfer equation (Eq. 2.2) in the comoving frame. In general, this translation requires a Lorentz transformation, though if we limit ourselves to non-relativistic velocities, we can explain the resulting equation only using Doppler shifts. The comoving frame version of the radiative transfer equation [Eq. 19.156 from Hubený and Mihalas, 2014, modified from 1D sperical geometry to a ray] is given by

$$\frac{\partial I(x,\nu_\mathrm{c})}{\partial x} - \frac{\nu_\mathrm{c}}{c}\left(\frac{\partial \mathrm{v}}{\partial x}\right)\frac{\partial I(x,\nu_\mathrm{c})}{\partial \nu} = \eta(x,\nu_\mathrm{c}) - \chi(x,\nu_\mathrm{c})I(x,\nu_\mathrm{c}). \tag{2.19}$$

For convenience, we denote the comoving frame frequency in this section with ν_c. If we ignore which frame we are using, Eq. 2.19 looks exactly like the standard radiative transfer equation, except for the extra term $\frac{\nu}{c}\left(\frac{\partial \mathrm{v}}{\partial x}\right)\frac{\partial I(x,\nu_\mathrm{c})}{\partial \nu}$ on the left hand side. When using the comoving frame, we continuously change which observer frame frequency we are using. This corresponds to adding a term $\frac{d\nu}{dx}\frac{\partial I}{\partial \nu}$ to the observer frame equation, in order to change which frequency we are using to evaluate the intensity. Filling in $\frac{d\nu}{dx}$ using the Doppler shift (see Eq. 2.13), we obtain the aforementioned extra term.

The comoving frame radiative transfer equation is more complex than the observer frame version, but still used in the literature [see e.g. Hauschildt, 1992, Castor, 2004]. The main computational advantage of this method is found when evaluating the mean line intensity J_{ij} (Eq. 4.20) for computing NLTE line radiative transfer. As the line profile function ϕ_{ij} has a limited range of

influence in frequency space, we can limit our computations to only compute intensities at the relevant frequencies, located around the line center. Due to Doppler shifts, this frequency changes in the observer frame as the velocity (projected to the line of sight) changes. Therefore, using the comoving frame formulation allows us to save computation time by only evaluating the intensity at relevant frequencies, close to the line centers.

For discretizing this equation, the previous works treat the position as space-like variable and the frequency as time-like variable[5], in order to overcome stability issues. The stability criterion described in Castor [2004] when using the position as a time-like variable instead, is given by

$$\Delta x \frac{|\partial v/\partial x|}{c} \frac{\nu}{\Delta\nu} < 1, \tag{2.20}$$

in which $\Delta\nu$ denotes the frequency spacing. A stability criterion denotes in which parameter regime a discretization of an equation is numerically stable, i.e. where any errors in the solution get damped. When computing detailed line radiative transfer (in which case $\Delta\nu/\nu$ is small), this stability criterion can indeed be violated. However, using frequency as time-like variable restricts the method to using monotonic velocity fields. In Baron and Hauschildt [2004], Hauschildt and Baron [2004], the authors proposed a different solution method, using position as time-like variable, which also allows for a non-monotonic velocity field. Although their solution method can be applied to models with a non-monotonic velocity field, it may encounter numerical instabilities [see e.g. Sampoorna and Nagendra, 2016, Hauschildt and Baron, 2004] in case of Doppler shifts larger than the spacing of the utilized frequency discretization. However, the comoving formulation of the radiative transfer equation offers a significant computational speedup for computing the intensity field for line radiative transfer. The computation time for calculating mean line intensities using long characteristics (see Section 2.2.1) scales with $O(N_{\text{points}}^{4/3})$ in a 3D model[6]. This scaling can be reduced to $O(N_{\text{points}})$ using the comoving formulation.

Synthetic observations 2.1.6

In the previous sections, we have described the physics concerning the transport of radiation. Now the question remains, what can we learn by simulating light? As mentioned in Section 1.3.3, many different types of astronomical observables exist, mostly based on the observation of light. Simulating radiative transfer allows us to create a synthetic observable from a model of an astrophysical

[5] Thus for each frequency, they solve the equation for all positions at once.

[6] For every position, long characteristics needs to trace a seperate ray, which has an average length of $O(N_{\text{points}}^{1/3})$.

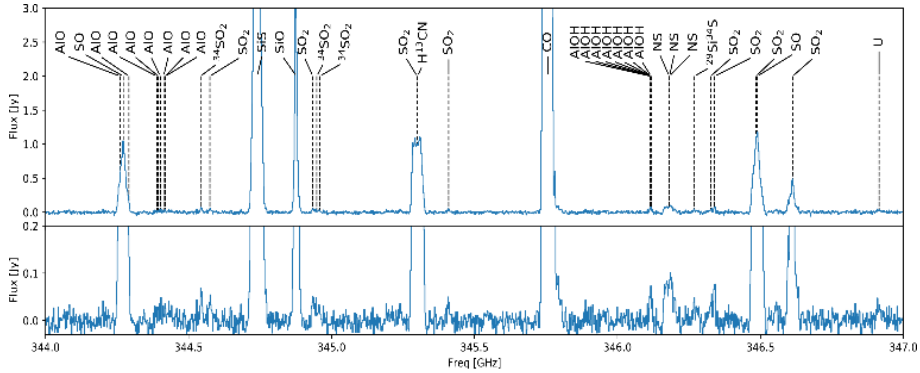


Figure 2.3: Extract from an ALMA observed spectrum of the AGB star IK Tau. The text explains which molecule causes a specific feature. As an example, the CO $J = 3 - 2$ line transition is found around 345.8 GHz. Figure taken from Decin et al. [2018].

object. This can then be compared with actual observations in order to try to model the astrophysical environment, which helps us understand what physics and chemistry take place.

For stellar environments, a particularly interesting observable are spectra (see Fig. 2.3). Spectra count the total intensity arriving from an object at narrowly spaced frequency bins. As stated in Section 2.1.2, atoms and molecules absorb and emit light in specific, narrow frequency bands. The presence of such lines then indicates whether a chemical species is present. From the height of the lines and ratios between these heights, one can derive temperatures and densities for the corresponding species. Furthermore, from the width and specific shape of these lines, one can infer a velocity distribution, because the Doppler shifts modify the frequency of light, depending on the velocity of the emitting matter. By modeling the environment using line spectra, one can obtain physical parameters of the stellar environment, including mass loss rates [Decin et al., 2006], and molecular abundances [Danilovich et al., 2014].

Another observable of interest to this thesis, are channel maps (see Fig. 2.4). Channel maps are similar to line profiles, but instead obtaining a single spectrum for an entire object, we have a 2D image for each frequency bin. This gives us almost a 3D view of the astronomical object, as the frequency distribution at each pixel can translated to a velocity distribution[7]. Therefore, by using channel

[7]To furthermore translate the velocity to a position, assumptions need to be made on the velocity field of the astronomical object.

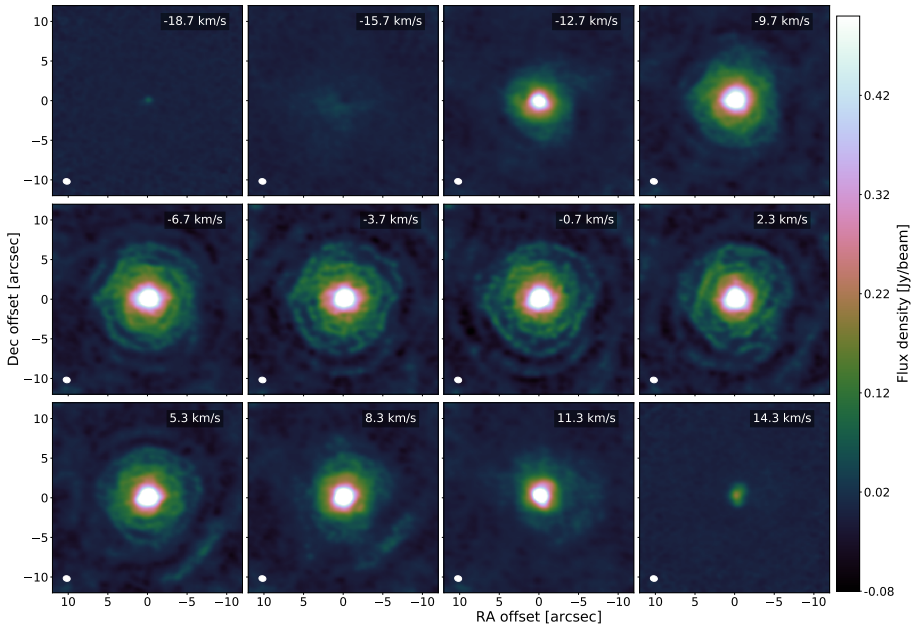


Figure 2.4: Channel map of the CO J=2-1 v=0 line emission from the star IRC-10529, observed using ALMA. The dot on the bottom left gives an indication of the angular resolution. Figure taken from Gottlieb et al. [2022].

maps, we can model the 3D structure of the AGB environment [Coenegrachts et al., 2023, Van de Sande et al., 2024].

Converting intensities to observables 2.1.7

Telescopes observe objects very far away. How sensitive they are to light originating from different directions is given by the beam response function G. For single dish telescopes, the beam response function is maximal in the direction at which these instruments point, but it quickly falls off for slightly different angles. For objects which are not just a point source, the total flux $F(\nu)$ observed by a telescope is therefore not just a summation over the intensities, but needs a varying weighting for the different angles. The observed flux F by a single dish telescope is therefore given by [Eq. 6 in Schoenberg, 1988]

$$F(\nu) = \int_{-\pi/2}^{\pi/2} \int_{0}^{2\pi} I(\nu, \theta, \phi) G(\theta, \phi) \cos(\theta) \sin(\theta) d\phi d\theta, \tag{2.21}$$

in which the $\cos(\theta)$ arises from the difference between the intensity direction and the flux direction and we integrate over half of the solid angle[8], and take into account the angle θ between telescope pointing direction and the current part we are integrating over. The beam response function G is typically approximated by a Gaussian, normalized at 1 in the center,

$$G(\theta, \phi) = e^{-\left(\frac{\theta}{\sigma/\left(2\sqrt{ln(2)}\right)}\right)^2}, \tag{2.22}$$

in which σ is the full width at half maximum (FWHM), (twice) the angle at which only half of the response remains. For a single dish instrument, the FWHM is linearly proportional to the measured wavelength, divided by the diameter D of the dish

$$\sigma \sim \frac{\lambda}{D}. \tag{2.23}$$

The FWHM is directly related to the minimum angular distance at which one can differentiate between objects, and thus determines the resolution of an observation. To obtain higher angular resolution observations, one can observe with interferometers, such as ALMA, given that their FWHM depends on the measured wavelength, divided by the baseline B, which is the distance between individual dishes,

$$\sigma \sim \frac{\lambda}{B}. \tag{2.24}$$

Impact on hydrodynamics and chemistry 2.1.8

As stated in the Section 1.3.3, light is not only the messenger of physics happening in faraway objects, but it also actively participates. Radiative pressure impacts the momentum in hydrodynamics, while radiative heating/cooling impacts the temperature, which is used in both hydrodynamics and chemistry. Finally, high-energy photons can impact the chemistry.

Radiative pressure

Light carries a momentum with magnitude $h\nu/c$. If light gets absorbed, its momentum gets transferred to the absorbing material. The reverse happens

[8]No light enters the telescope from behind, so we can maximally integrate over half of the solid angle. Note that the beam response function G is only relevant in a fraction of this angle.

when matter emits light. The total photon momentum flux $\boldsymbol{M}$ is given by [based on Eq. 3.67 from Hubený and Mihalas, 2014]

$$\boldsymbol{M} = \frac{1}{c} \int_0^\infty \oint_\Omega \hat{\boldsymbol{n}} I(\boldsymbol{x}, \nu, \hat{\boldsymbol{n}}) d\Omega d\nu. \tag{2.25}$$

From this, we see that a net flux arises when there is an imbalance in the intensity coming from different directions. In a stellar environment, the main source of radiation is the star. Thus the absorption of radiation will attempt to push matter away from the star. However, evaluating the intensity field I, is expensive, when compared to a typical hydrodynamics simulation step. Thus several prescriptions have been made to evaluate it efficiently [see e.g. Esseldeurs et al., 2023].

Radiative cooling

Radiation transport impacts the energy balance, as photons contain energy $h\nu$. Thus, when matter emits radiation, it loses energy, and thus cools. Similarly, matter absorbing radiation heats up. By calculating the difference between how much energy is emitted and absorbed, we obtain the net cooling rate Λ [Carlsson and Leenaarts, 2012]:

$$\Lambda = \int \chi(\boldsymbol{x}, \nu)(S(\boldsymbol{x}, \nu) - I(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu)) d\nu d\Omega. \tag{2.26}$$

In this equation, we integrate the difference between emitted and absorbed intensity over the entire solid angle and frequency spectrum. For line radiative transfer, we can instead look at the energy rate required for the collisional transitions to sustain the current energy state [Sahai, 1990]:

$$\Lambda = \sum_{\text{transitions J=ij}} (n_j C_{ji} - n_i C_{ij}) h\nu_{ij}. \tag{2.27}$$

For NLTE line radiative transfer, we prefer the latter formulation, as we do not need to discretize the integral in Eq. 2.26. Furthermore, cancellation errors can occur in the former in optically thick regions, which are nearly in LTE, such that $I \simeq S$, but not exactly. During testing, we found Eq. 2.27 to produce more reliable results. For obtaining accurate cooling rates, we must calculate the radiation field and corresponding level populations self-consistently. Therefore, in the next part (Section 2.2), we explore how to practically compute NLTE line radiative transfer.

Photochemistry

AGB stars are relatively cool, with effective temperatures[9] ranging from around 2000 K to 3000 K [Bladh et al., 2019]. They therefore do not produce significant amounts of ultraviolet (UV) radiation. Any high-energy photons in the AGB environment originate instead from the ISM. These UV photons can interact with the gas and dust chemistry in the AGB outflow [Van de Sande et al., 2020]. They can dissociate molecules and impact the composition of dust which gets deposited in the ISM. For example, CO molecules get dissociated on the outside of the stellar wind. The dissociation rate drops exponentially with the encountered optical depth, as it is proportional to the UV intensity, which drops as $\exp(-\tau)$, assuming no source of UV photons is present in the outflow. Therefore, thanks to the high abundance of CO in the outflow, the CO in the inner layers gets shielded from UV radiation by the CO in the outer layers [Groenewegen, 2017][10]. Given its impact on the molecular abundances, radiative transfer is required to accurately simulate the chemistry in the AGB environment.

Numerical radiative transfer 2.2

In this section, we will explain how to numerically solve radiative transfer. We start by giving an overview of the most commonly used radiative transfer methods in Section 2.2.1. In general, most methods are based on a discretization of the formal solution, which is explained in Section 2.2.2. Afterwards, we explain another approach to solve the radiative transfer equation: the Feautrier method (see Section 2.2.3).

Radiative transfer solution methods 2.2.1

Many variations exist both in the details of the radiation transport physics we can try to simulate and on how to solve the resulting equations. In this section, we will give a brief overview of the most popular solution methods, and give examples of simulation codes which use them.

[9] In astronomy, the effective temperature of a star is defined as the temperature of a black body with the same luminosity. This black-body spectrum gets further modified by material in higher layers of the star.

[10] The self-shielding is applicable as long as no significant variation in velocity exists throughout the envelope. Doppler shifts significantly reduce the frequency-dependent optical depth, and thus increase how far UV photons can penetrate.

The main result computed by most radiative transfer simulations, is an intensity-like quantity or a quantity which can be derived from this. The possible quantities range from the specific intensity, depending on direction, position, and frequency[11], to more averaged out quantities, which integrate out their frequency and/or angle dependence. The choice of which to use depends on the considered physics and required computational demands.

Long characteristics

We start by explaining long characteristics, a spatial discretization for the radiative transfer equation, which can be used in both the Feautrier and formal solution type methods. In long characteristics, to compute an intensity, you trace a ray through each position, in each desired direction, until you hit the model boundary. The ray then discretizes the medium through which a photon flies, when traveling to that position in the given direction. The main upside of this discretization is its accuracy, as it simulates the physics along the exact path which light follows. The main downside is the computation time required to compute the intensity field, as it scales proportionally to the number of model points, times the average length of a ray, as we are required to compute the intensity along the entire ray. The long characteristics method has been used for example in Magritte [De Ceuster et al., 2022] and CO5BOLD [Freytag et al., 2012].

Short characteristics

In contrast, short characteristics traces rays for each individual point up until the next spatial position. The resulting segments are stitched together in order to compute intensities[12]. It is therefore only applicable to formal solution type methods. One starts by computing the intensity at positions directly next to the model boundary. These intensities can subsequently be used as initial condition for computing the intensity at nearby positions. This repeats until the intensity for a given direction has been computed in the entire model. The main advantage, when compared to long characteristics, is the increase of computational efficiency, as we only need to compute the intensity for a given position once (per direction). The main downside of this method is that the intensity is dispersive. A beam of intensity originating from a small source can become wider, due to the interpolations required during computation [de Vicente

[11] Time can also be included if using time-dependent radiative transfer.
[12] The segments might not line up exactly, thus interpolation is required.

et al., 2021], see Fig. 2.5. Short characteristics has been used in e.g. PORTA [Štěpán and Bueno, 2013] and ANTARES [Kostogryz et al., 2021].

Moment method

Often, one is not particularly interested in the specific intensity, but one only requires averages, integrated over the solid angle and possibly the frequency spectrum, because these averages determine the energy and momentum transport. The first three angular moments are the mean intensity J, the intensity flux $\boldsymbol{H}$, and the tensor $\overline{\overline{\boldsymbol{K}}}$ [see Eqs. 4.9 to 4.11 from Castor, 2004]:

$$J(\boldsymbol{x},\nu) = \frac{1}{4\pi}\oint_\Omega I(\hat{\boldsymbol{n}},\boldsymbol{x},\nu)d\Omega \tag{2.28}$$

$$\boldsymbol{H}(\boldsymbol{x},\nu) = \frac{1}{4\pi}\oint_\Omega \hat{\boldsymbol{n}} I(\hat{\boldsymbol{n}},\boldsymbol{x},\nu)d\Omega \tag{2.29}$$

$$\overline{\overline{\boldsymbol{K}}}(\boldsymbol{x},\nu) = \frac{1}{4\pi}\oint_\Omega \hat{\boldsymbol{n}}\hat{\boldsymbol{n}} I(\hat{\boldsymbol{n}},\boldsymbol{x},\nu)d\Omega. \tag{2.30}$$

Such integrated quantities can for example be used to calculate the radiation momentum (Eq. 2.25) and radiative cooling (Eq. 2.26). To save computation time, one can drastically reduce the number of equations to solve by integrating the standard radiative transfer equation over the solid angle and the entire frequency range. This results in a new equation,

$$\int_0^\infty \oint_\Omega \hat{\boldsymbol{n}}\cdot\nabla I(\hat{\boldsymbol{n}},x,\nu)d\Omega d\nu = \int_0^\infty \oint_\Omega \eta(x,\nu) - \chi(x,\nu) I(\hat{\boldsymbol{n}},x,\nu)d\Omega d\nu \tag{2.31}$$

which contains (the divergence of) the first moment of the intensity $\nabla\cdot\boldsymbol{F} = \oint_\Omega \hat{\boldsymbol{n}}\cdot\nabla I d\Omega$ on the left hand side and the zeroth moment of the intensity $J = \oint_\Omega I d\Omega$ on the right hand side. Any higher order moment can be obtained by multiplying both sides with an extra direction vector $\hat{\boldsymbol{n}}$ before integrating. Doing so results in an equation with an $(n+1)$-th moment of the intensity on the left hand side and an n-th moment of the intensity on the right hand side. To solve the resulting system of equations, we require a closure relation, an assumption which relates the n-th moment of the intensity to the lower order moments. With this significant reduction in parameter space, radiation transport can be evaluated sufficiently fast, in order to be used on the fly in radiation-hydrodynamics (RHD) simulations. The main downside of this method, is that the obtained solution may depend on which closure relation is used [Menon et al., 2022], as a local approximation may not capture asymmetries in the radiation field. In contrast, a closure relation using non-local information,

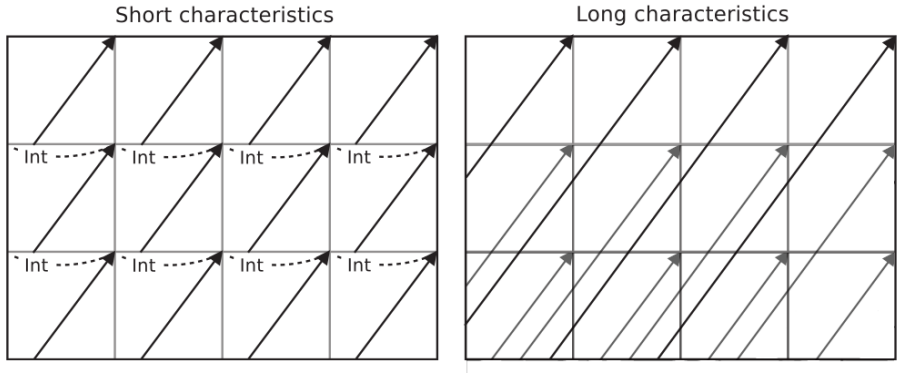


Figure 2.5: Illustration of the differences between short and long characteristics. Short charateristics requires interpolation to compute the intensity I at each grid point, while long characteristics requires a unique ray for each position. Figure modified from de Vicente et al. [2021].

will be more expensive to evaluate, as it requires an estimate of the (angle-dependent) radiation field. The moment method has mainly been used in RHD simulations, such as RAMSES-RT [Rosdahl et al., 2013] and the flux-limited diffusion module in MPI-AMRVAC [Moens et al., 2022].

Monte Carlo radiative transfer

Light can be interpreted both as a wave or as particles. Instead of computing the energy flux, one can simulate the motion of packets of photons. As the underlying physics is the same, most equations can be reused to simulate the photons. However, we have one key difference, photons are discrete. Thus they get absorbed and emitted in entirety. Photons have a chance to get absorbed and possibly re-emitted when encountering matter. This reveals the probabilistic nature of Monte Carlo radiative transfer. By simulating large quantities of photons, one can obtain an accurate statistical approximation for the radiation field. The main advantage is that Monte Carlo radiative transfer naturally accomodates photon scattering. The main disadvantage is that photons are unlikely to successfully traverse high optical depth regimes [Camps and Baes, 2018], skewing computed results. Furthermore, statistical noise is inherent to this method, which makes it harder to quantify convergence of a simulation and hinders the use of typical acceleration of convergence methods for NLTE radiative transfer, as detailed in Section 2.3. Examples of codes using this

method are SKIRT [Baes and Camps, 2015] and RADMC3D [Dullemond et al., 2012].

Emulation

More recently, neural networks gained traction in physics simulations, promising significant computational advantages. In this approach one replaces existing numerical calculations with evaluating a black-box function, which has been trained to mimic the behavior of given training data, obtained from previous simulations. As radiative transfer can be computationally expensive to simulate, generating sufficient training data can be prohibitively expensive. Once the neural network is trained, however, one can bypass the computationally expensive simulation, replacing it with evaluations of the fitted function. Emulation of radiative transfer has been applied to a wide variety of astrophysical environments, including protoplanetary discs [Kaeufer et al., 2023], AGB outflows [Su et al., 2025], stellar atmospheres [Chappell and Pereira, 2021], and the Earth atmosphere [Lagerquist et al., 2021].

PINNS

Instead of relying on training data to fit a general black-box function, as done in emulation, one can use the same machine learning framework to fit the equations describing the physics, using Physics Informed Neural Networks (PINNs) [Raissi et al., 2019]. The main advantage of this method is that it is relatively easy to implement, as we only need to be able to evaluate the residual of the discretized radiative transfer equation[13], using output from a neural network architecture. However, it is not necessarily cheaper to use PINNS to fit a solution than to use regular numerical methods [Chen et al., 2022].

Retrieval modeling

In the previous parts, we focused on how to convert a numerical simulation of an AGB star to a synthetic observation, using radiative transfer simulations. What we often want to know is what physics is happening in a faraway object, based on an observation. Thus we want to be able to convert an observation to a model of the corresponding physical environment. This is the idea behind retrieval modeling. Typically, an observer parametrizes a model, creates synthetic

[13]The residual is defined as the difference between both sides of an equation and gives an answer for how bad a guess for the solution agrees with the physics.

observations using a radiative transfer code, and tweaks the model parameters such that it fits the observation[14]. However, this procedure requires a significant amount of effort and intuition, as the observer first needs to understand how the physical environment looks like, based on the observation, before being able to define the parameterized model.

Recently, progress has been made to reduce the manual effort required for retrieval modeling. By discretizing the entire model space, one no longer needs to manually describe any features of the physical environment. If one has access to a differentiable radiative transfer model, one can apply gradient descend to find the optimal parameters, as done in [De Ceuster et al., 2024]. Both retrieval modeling and NLTE radiative transfer involve iterative computations. By simultaneously fitting an observation and doing the NLTE iterations, one can optimize retrieval modeling [Štěpán et al., 2022].

The formal solution 2.2.2

To numerically solve the radiative transfer equation, many radiative transfer codes [see e.g. Ramos and Bueno, 2006, Štěpán and Bueno, 2013, Hauschildt and Baron, 2014, Tessore et al., 2021] use a solver based on the formal solution [see e.g. Kunasz and Auer, 1988]. Starting from Eq. 2.2, we divide both side by the opacity χ, obtaining

$$\frac{\partial I(x,\nu)}{\partial \tau} = S(x,\nu) - I(x,\nu). \tag{2.32}$$

We can rewrite this equation by multiplying both sides with $e^{\tau(x,\nu)}$ and moving the intensity term to the left, such that

$$\frac{\partial \left(e^{\tau(x,\nu)} I(x,\nu)\right)}{\partial \tau} = e^{\tau(x,\nu)} \frac{\partial I(x,\nu)}{\partial \tau} + e^{\tau(x,\nu)} I(x,\nu) = e^{\tau(x,\nu)} S(x,\nu). \tag{2.33}$$

By integrating, we obtain the formal solution for the intensity I,

$$I(x,\nu) = I_{\text{bdy}}(\nu) e^{-\tau(x,\nu)} + \int_0^{\tau(x,\nu)} S(t,\nu) e^{t-\tau(x,\nu)} dt, \tag{2.34}$$

in which I_{bdy} is the boundary intensity and τ is the optical depth. We discretize this equation by defining distinct points $x_d \in 1, \ldots, N$, in which we define x_1 to be the point at which we want to compute the intensity and x_N the end (boundary) point of the computation. For convenience, we write down the intensity contribution ΔI_d for each segment $[\tau(x_d,\nu), \tau(x_{d+1},\nu)]$ of the integral

[14] This is often based on a statistical distance measure, such that it can be automated.

individually. Using a first order accurate discretization[15] for the source function S, we obtain

$$\Delta I_d(\nu) = \left(S_d \frac{e^{-\Delta\tau_d} - 1 + \Delta\tau_d}{\Delta\tau_d} + S_{d+1} \frac{1 - e^{-\Delta\tau_d} - \Delta\tau e^{-\Delta\tau_d}}{\Delta\tau_d} \right) e^{-\tau_d}, \quad (2.35)$$

in which $\tau_d = \int_{x_1}^{x_d} \chi(x,\nu)dx$ is the optical depth from x_1 to x_d, $\Delta\tau_d = \tau_{d+1} - \tau_d$ is the optical depth increment of the current segment and $S_d = S(x_d, \nu)$ is the source function evaluated at x_d. The integral for the optical depth τ can be evaluated using the trapezoidal rule, or using a semi-analytic approach as in Section 3.4.3. Finally, the intensity contribution ΔI_{bdy} given by the boundary condition is simply

$$\Delta I_{\text{bdy}} = I_{\text{bdy}}(x_N, \nu) e^{-\tau_N}, \quad (2.36)$$

in which I_{bdy} is the intensity given by the boundary condition. The total intensity is then obtained by summing over the contributions of all segments ΔI_i for $i \in 1, \ldots, N-1$ and the boundary contribution ΔI_{bdy}.

The Feautrier method 2.2.3

The Feautrier method has originally been described in [Feautrier, 1964]. For this method, one first defines the mean-intensity-like quantity u[W sr^{-1}m^{-2}Hz^{-1}] along the ray as

$$u = \frac{I(\hat{\boldsymbol{n}}, x, \nu) + I(-\hat{\boldsymbol{n}}, x, \nu)}{2}, \quad (2.37)$$

which averages the intensity in two opposite directions. Similarly, we define the flux-like quantity v[W sr^{-1}m^{-2}Hz^{-1}] along the ray

$$v = \frac{I(\hat{\boldsymbol{n}}, x, \nu) - I(-\hat{\boldsymbol{n}}, x, \nu)}{2}. \quad (2.38)$$

Using these, and the source function S[W sr^{-1} m^{-2} Hz^{-1}]

$$S(x, \nu) = \frac{\eta(x, \nu)}{\chi(x, \nu)}, \quad (2.39)$$

[15] Higher order accurate discretizations are possible, but require caution during implementation, as the spacing τ_d can be highly irregular due to Doppler shifts interacting with line profiles.

Eq. 2.2 can be rewritten in two first order equations

$$\frac{du(x,\nu)}{d\tau} = -v \tag{2.40}$$

$$\frac{dv(x,\nu)}{d\tau} = S - u \tag{2.41}$$

Combining these equations, the second order equation can be obtained,

$$u(x,\nu) - \frac{d^2u(x,\nu)}{d\tau^2} = S(x,\nu). \tag{2.42}$$

The discretization of this equation, using distinct spatial positions $x_d, d \in [1,\ldots,N]$ and a given frequency bin ν, is given by

$$-A_d u_{d-1} + B_d u_d - C_d u_{d+1} = S_d, \tag{2.43}$$

where

$$A_d = \frac{2}{\Delta\tau_{d-1}(\Delta\tau_{d-1} + \Delta\tau_d)}, \tag{2.44}$$

$$B_d = 1 + \frac{2}{\Delta\tau_{d-1}\Delta\tau_d}, \tag{2.45}$$

$$C_d = \frac{2}{\Delta\tau_d(\Delta\tau_{d-1} + \Delta\tau_d)}, \tag{2.46}$$

$S_d = S(x_d,\nu)$ and $\Delta\tau_d = \tau(x_{d+1},\nu) - \tau(x_d,\nu)$. Second-order accurate boundary conditions are given by setting the appropriate matrix coefficients, obtained by creating a Taylor expansion around $x = x_1$,

$$u_2 = u_1 + \Delta\tau_1 \left.\frac{du(x,\nu)}{d\tau}\right|_{x=x_1} + \frac{1}{2}\Delta\tau_1^2 \left.\frac{d^2u(x,\nu)}{d\tau^2}\right|_{x=x_1} \tag{2.47}$$

and replacing the derivatives using Eqs. 2.40, 2.42 and filling in the boundary condition $I(x_1,\nu) = u_1 - v_1 = I_1$

$$\left.\frac{du(x,\nu)}{d\tau}\right|_{x=x_1} = u_1 - I_1, \qquad \left.\frac{d^2u(x,\nu)}{d\tau^2}\right|_{x=x_1} = u_1 - S_1 \tag{2.48}$$

We obtain

$$u_2 = u_1\left(1 + \Delta\tau_1 + \frac{\Delta\tau_1^2}{2}\right) - \Delta\tau_1 I_1^+ - \frac{\Delta\tau_1^2}{2}S_1 \tag{2.49}$$

Dividing by $\Delta\tau_1^2/2$, and a slight rewrite later, we obtain a formula similar to the Feautrier discretization (Eq. 2.43)

$$u(x_1,\nu)\left(1+\frac{2}{\Delta\tau_1}+\frac{2}{\Delta\tau_1^2}\right)-\frac{2}{\Delta\tau_1^2}u_2=S_1+\frac{2}{\Delta\tau_1}I_1. \tag{2.50}$$

The same procedure can be repeated for the other boundary condition on the ray. The boundary conditions can be enforced by setting

$$B_1=1+\frac{2}{\Delta\tau_1}+\frac{2}{\Delta\tau_1^2} \tag{2.51}$$

$$C_1=\frac{2}{\Delta\tau_1^2} \tag{2.52}$$

$$A_N=\frac{2}{\Delta\tau_{N-1}^2} \tag{2.53}$$

$$B_N=1+\frac{2}{\Delta\tau_{N-1}}+\frac{2}{\Delta\tau_{N-1}^2} \tag{2.54}$$

and adding an extra term to the source function at the first and last point,

$$S_1\rightarrow S_1+\frac{2}{\Delta\tau_1}I_1, \tag{2.55}$$

$$S_N\rightarrow S_N+\frac{2}{\Delta\tau_{N-1}}I_{N-1}. \tag{2.56}$$

In these equations, I_1 and I_{N-1} are respectively the boundary intensity at the lower and upper end of the ray. Together, these discretized equations form a matrix-vector equation used for calculating the mean intensities u_d.

Acceleration of convergence methods 2.3

NLTE line radiative transfer is computationally expensive. Especially when applied to the large 3D models which originate from hydrodynamics simulations. In NLTE radiative transfer, we calculate both the radiation field and the energy state of the medium in a self-consistent way. We typically employ an iterative method for this, calculating successive estimates for the for the level populations n_i, until convergence is reached. However, every iteration in NLTE line radiative transfer is computationally expensive. Therefore, we want to do as few iterations as possible before model convergence is reached. Throughout the literature, several standard optimization procedures for NLTE radiative transfer have been devised for increasing the convergence rate. In this section, we explain those relevant to this thesis.

ALI 2.3.1

In optically thick regimes, NLTE line radiative transfer converges slowly [see e.g. van Zadelhoff et al., 2002] when naively iterating based on previously computed values of the source function. This process is Lambda iteration:

$$J_{ij} = \Lambda(\boldsymbol{S}), \tag{2.57}$$

in which Λ abstracts away all computations needed to calculate the intensity field based on the source function[16]. To improve upon this, you can apply an operator-splitting technique, separating the local and non-local contributions to the intensity field.

$$J_{ij} = (\Lambda - \Lambda^*)(\boldsymbol{S}) + \Lambda^*(\boldsymbol{S}) \tag{2.58}$$

in which Λ^* is the local contribution to the Λ operator. This is the basic idea behind Accelerated Lambda Iteration (ALI) (see Rybicki and Hummer [1991]). In this thesis, we only include the diagonal terms of Λ in Λ^*, as this leads to a block-diagonal matrix when solving statistical equilibrium equation, see Eq. 2.63, which is computationally advantageous. Including non-local contributions from other positions introduces extra terms coupling the different positions. When solving the radiative transfer equation, we note down how much the radiation field depends on the local source function,

$$J_{ij}(\boldsymbol{x}) = l^*_{ij}(\boldsymbol{x}) \cdot S_{ij}(\boldsymbol{x}) + \dots, \tag{2.59}$$

in which l^*_{ij} is the relative contribution of the source function to the mean line intensity. This fraction l^*_{ij} can be calculated by inspecting the numerical discretization used when computing an intensity. We explain how to obtain l^*_{ij} both for the Feautrier method and the formal solution. In the Feautrier method, when computing the mean intensity u, we obtain a matrix equation of the form

$$A\boldsymbol{u} = \boldsymbol{S} \tag{2.60}$$

in which A is a tridiagonal matrix[17], $\boldsymbol{u}$ is the vector of mean-like intensities and $\boldsymbol{S}$ is a vector of source function. From this matrix equation, we see that the source function $\boldsymbol{S}$ can be written as a linear combination of the mean intensity $\boldsymbol{u}$, using coefficients in A. Similarly, $\boldsymbol{u}$ can be written as a linear combination

[16] Note that some definitions might start from the level populations instead. I prefer the source function variation, as we can then formulate ALI as: this fraction of the intensity which is produced by the local source function.

[17] Except for the top and bottom row, which are discretized using a second order accurate boundary condition, and thus contain an extra element.

of the source function $\boldsymbol{S}$, using the coefficients A^{-1}. In practice, we only want the contributions to the mean line intensity J_{ij} at position $\boldsymbol{x}$. Therefore, we are satisfied with computing the i-th row of A^{-1}, in which i corresponds to the index corresponding to position $\boldsymbol{x}$. Thus we can solve $\boldsymbol{l}^T A = \boldsymbol{e}_i^T$ or equivalently, $A^T \boldsymbol{l} = \boldsymbol{e}_i$, in which $\boldsymbol{e}_i$ is a vector containing only zeros, except for a single 1 at index i[18]. The solution vector $\boldsymbol{l}$ will then contain the dependence of $J_{ij}(\boldsymbol{x})$ on the source functions at each discretized position. We have now obtained the contribution of the source function S to mean line intensity J_{ij} in a specific direction, at a specific frequency, which we denote with $l^*_{ij}(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu)$[19].

For the formal solution, we can directly derive l^*_{ij} from the discretization itself, as the intensity is already written as a linear combination of source functions. For example, using the first-order accurate discretization (Eq. 2.35), using a local approximated lambda operator, $l^*_{ij}(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu)$ is given by

$$l^*_{ij}(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu) = \frac{e^{-\Delta\tau} - 1 + \Delta\tau}{\Delta\tau}, \tag{2.61}$$

in which $\Delta\tau$ is the optical depth increment from the considered point to the next spatial position. This is given by the first term of Eq. 2.35.

After we have computed $l^*_{ij}(\hat{\boldsymbol{n}}, \nu)$ for all discretized frequencies and angles, all that remains is numerically integrating the obtained values for l^*_{ij}, in a similar manner to computing J_{ij} from $I(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu)$

$$l^*_{ij}(\boldsymbol{x}) = \frac{1}{4\pi} \oint \int l^*_{ij}(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu) \phi_{ij}(\boldsymbol{x}, \nu) d\nu d\Omega \tag{2.62}$$

This value can then be filled in into the statistical equilibrium equations (Eq. 2.15), obtaining

$$\begin{aligned} \frac{\partial n_i}{\partial t} =& \sum_{j<i} \left((1 - l^*_{ij}) n_i A_{ij} - (n_j B_{ji} - n_i B_{ij}) J_{ij,\text{eff}} \right) \\ & - \sum_{j>i} \left((1 - l^*_{ij}) n_j A_{ji} - (n_i B_{ij} - n_j B_{ji}) J_{ij,\text{eff}} \right) \\ & + \sum_{j=1}^{N} \left(n_i C_{ij} - n_j C_{ji} \right), \end{aligned} \tag{2.63}$$

[18] If you already happen to have computed a matrix decomposition of A when computing the mean-like intensity $\boldsymbol{u}$, you should reuse it in this step for computational efficiency.

[19] The Feautrier method computes a mean intensity. Do not forget to assign the computed value $l^*_{ij}(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu)$ to both directions $\hat{\boldsymbol{n}}$ and $-\hat{\boldsymbol{n}}$.

in which we have defined the effective mean line intensity $J_{ij,\text{eff}}(\boldsymbol{x}) = J_{ij}(\boldsymbol{x}) - l^*_{ij}(\boldsymbol{x}) \cdot S_{ij}(\boldsymbol{x})$. In this way, ALI effectively reduces the coefficients, reducing the condition number of the resulting matrix equation.

Ng acceleration 2.3.2

Computing NLTE radiative transfer is an iterative process. By observing the change of the level populations over the different iterations, we can try to predict the final converged result. In radiative transfer, this is done using Ng-acceleration [Ng, 1974]. In the mathematics literature, this method is categorized under acceleration-of-convergence techniques, and has many related variants (see e.g. Sidi [2020]), using different assumptions on the convergence behavior in order to reduce the number of required iterations. For example, Ng-acceleration assumes that the convergence behavior is linear, which is true in the neighborhood of the solution for NLTE radiative transfer.

Ng-acceleration assumes that the converged level populations $\boldsymbol{n}_\star$ can be written as linear combination of the N last iterates,

$$\boldsymbol{n}_\star = \sum_{n=1}^{N} c_n \, \boldsymbol{n}_{-n}, \tag{2.64}$$

where we used negative indices to denote the previously computed level populations. To ensure that the total level populations remain conserved in the prediction, we require the coefficients c_n to sum to one, i.e. $\boldsymbol{c}^{\mathrm{T}}\mathbf{1} = 1$, where $\mathbf{1}$ is a vector of ones. The goal is now to find the coefficients, c_n, such that $\boldsymbol{n}_\star$ is the "optimal prediction" in some sense for the fixed-point of the iterative process that we are interested in. We abstract the entire iteration NLTE step by defining the operator Λ, which given initial level populations, computes the level populations in the next iterate. At the fixed point $\boldsymbol{n}_*$,

$$\boldsymbol{n}_* = \Lambda[\boldsymbol{n}_*]. \tag{2.65}$$

In non-LTE line transfer, this fixed point corresponds to the level populations that are self-consistent with the radiation field, given that the radiation field we calculate from them gives the fixed point level populations back when applying the statistical equilibrium equations. Using this formalism, we can define the optimal coefficients, c_n, as the solution of the following constraint minimisation problem,

$$\boldsymbol{c} = \underset{\boldsymbol{c}:\, \boldsymbol{c}^{\mathrm{T}}\mathbf{1}=1}{\operatorname{argmin}} \left\{ \left\| \boldsymbol{n}_* - \Lambda[\boldsymbol{n}_*] \right\|^2 \right\} \tag{2.66}$$

In words, we want the fixed-point equation (2.65) to be satisfied as closely as possible, given the constraint $\boldsymbol{c}^{\mathrm{T}}\mathbf{1} = 1$. In order to solve this problem, we need to assume that the Λ-operator is linear. Although, in general, this assumption will not hold, it will hold approximately near the fixed-point, which is the region we are interested in anyway. Assuming linearity of the Λ-operator and substituting equation (2.64), yields,

$$\boldsymbol{c} = \underset{\boldsymbol{c}:\, \boldsymbol{c}^{\mathrm{T}}\mathbf{1}=1}{\operatorname{argmin}} \left\{ \left\| \sum_{n=1}^{N} c_n \, \boldsymbol{n}_{-n} - \sum_{n=1}^{N} c_n \, \Lambda[\boldsymbol{n}_{-n}] \right\|^2 \right\}, \tag{2.67}$$

$$= \underset{\boldsymbol{c}:\, \boldsymbol{c}^{\mathrm{T}}\mathbf{1}=1}{\operatorname{argmin}} \left\{ \left\| \sum_{n=1}^{N} c_n \big(\boldsymbol{n}_{-n} - \boldsymbol{n}_{-n+1}\big) \right\|^2 \right\}, \tag{2.68}$$

where, in the second line, we used that $\boldsymbol{n}_{i+1} = \Lambda[\boldsymbol{n}_i]$. Defining the residuals, $\boldsymbol{r}_n \equiv \boldsymbol{n}_{-n} - \boldsymbol{n}_{-n+1}$, and accumulating them in a matrix,

$$R^{\mathrm{T}} \equiv \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \dots & \boldsymbol{r}_{N-1} \end{bmatrix} \tag{2.69}$$

the constraint minimisation problem can be written as,

$$\boldsymbol{c} = \underset{\boldsymbol{c}:\, \boldsymbol{c}^{\mathrm{T}}\mathbf{1}=1}{\operatorname{argmin}} \left\{ \boldsymbol{c}^{\mathrm{T}} \, RR^{\mathrm{T}} \, \boldsymbol{c} \right\}, \tag{2.70}$$

$$= \underset{\boldsymbol{c},\, \lambda}{\operatorname{argmin}} \left\{ \boldsymbol{c}^{\mathrm{T}} \, RR^{\mathrm{T}} \, \boldsymbol{c} \; + \; \lambda \left(\boldsymbol{c}^{\mathrm{T}}\mathbf{1} - 1 \right) \right\}, \tag{2.71}$$

where, in the second line, we accounted for the constraint by introducing a Lagrange multiplier, λ. At the minimum, both the gradient with respect to $\boldsymbol{c}$ and the gradient with respect to λ should vanish, yielding equations,

$$0 = 2\boldsymbol{c}^{\mathrm{T}} RR^{\mathrm{T}} \; + \; \lambda \mathbf{1}^{\mathrm{T}} \tag{2.72}$$

$$0 = \boldsymbol{c}^{\mathrm{T}}\mathbf{1} - 1 \tag{2.73}$$

Solving for λ, yields,

$$\lambda = \frac{-2}{\mathbf{1}^{\mathrm{T}} \big(RR^{\mathrm{T}}\big)^{-1} \mathbf{1}}, \tag{2.74}$$

while solving for $\boldsymbol{c}$ and substituting the result for λ yields,

$$\boldsymbol{c} \; = \; \frac{\big(RR^{\mathrm{T}}\big)^{-1}\mathbf{1}}{\mathbf{1}^{T} \big(RR^{\mathrm{T}}\big)^{-1} \mathbf{1}}. \tag{2.75}$$

These are the coefficients that make equation (2.64) optimally predict the fixed-point of the iterative process defined by Λ, in the sense of equation (2.66), assuming linearity.

Magritte 2.4

Simulating radiative transfer allows us to compare observations with simulations. Starting from an observation of a distant astronomical object, one can create a model of the environment, for which a synthetic observation can be made. The model for which the synthetic observation agrees most with the actual observation, can then be used to infer the physical conditions of the object. On the other hand, one can start from hydrodynamics models, creating synthetic observations of these. From this, we can deduce which features we might be able to observe.

In both applications, simulating radiative transfer is crucial. In this thesis, we focus on MAGRITTE[20][Ceulemans et al., 2024, De Ceuster, 2022, De Ceuster et al., 2020, 2019], an open-source 3D NLTE line radiative transfer library. The code has mainly been used to simulate the environment of AGB stars. Examples include modeling the distribution of NaCl around IK Tauri Coenegrachts et al. [2023], and modeling the molecule-rich disk around L2 Puppis Van de Sande et al. [2024].

We clarify the computational details of MAGRITTE in Section 2.4.1. Afterwards, we explain how to use the code in Section 2.4.2. Finally, we detail any further improvements in Section 2.4.3.

Computational details 2.4.1

The code MAGRITTE is able to solve the radiative transfer equation using both the first order accurate formal solution discretization (see Section 2.2.2) and the Feautrier solver (see Section 2.2.3), and uses long characteristics. We can apply the code to post-process any 3D hydrodynamics simulation, regardless of whether it originates from adaptive mesh refinement (AMR) or smoothed particle hydrodynamics (SPH). This is achieved by implementing a flexible spatial discretization, which uses a point cloud with neighbor information. For this discretization, we require the locations for each individual point and the corresponding indices of all neighboring points. This information is available in any grid-based simulation and can obtained from particle-based simulations after applying a Delaunay triangulation[21] to the point cloud.

In order to evaluate the radiation field $I(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu)$, we discretize the path traveled by light (in a given direction) using ray-tracing [as described in De Ceuster

[20] Available at `https://github.com/Magritte-code/Magritte`

[21] We choose to employ a Delaunay triangulation as it avoids generating triangles with very small angles, which would impact the ray tracing scheme.

et al., 2020]. We briefly summarize the ray-tracing algorithm (see Fig. 2.6) here. Starting from a origin point in the point cloud, we find the next point in a given direction by looking at all neighboring points, choosing the neighbor whose distance is closest to the ray, defined by the straight line drawn through the origin point, and lies in the forward direction, i.e.

$$\hat{\boldsymbol{n}} \cdot (\boldsymbol{x}_{\text{neighbor}} - \boldsymbol{x}_{\text{current}}) > 0, \tag{2.76}$$

in which $\hat{\boldsymbol{n}}$ is the ray direction and $\boldsymbol{x}_{\text{neighbor}}$ and $\boldsymbol{x}_{\text{origin}}$ are respectively the position of the considered neighboring point and current point in the ray-tracing algorithm. Then this neighbor gets added to the list of discretized positions used for calculating the intensity. Next, we find the neighbors of this point, using exactly the same algorithm. The ray-tracing algorithm ends when we reach the boundary of the model. When calculating the intensities, we calculate the emissivities and opacities at the discrete positions on the ray using nearest neighbors interpolation. When benchmarking MAGRITTE and SKIRT [Baes and Camps, 2015], which does interpolate these properties, we obtain the same results [Vermeulen et al., 2025]. Therefore, we do not deem it necessary to implement a better interpolation scheme for the properties at the ends of each ray segment in MAGRITTE. However, we should clarify that we do add some intermediate interpolation points within the segments, when it is required in the calculation (see Section 3.4.5).

For the angular discretization of the intensity, we rely on the HEALPIX discretization [Górski et al., 2005] of the solid angle. It discretizes the entire solid angle into 12 regions with equal area, and can be subdivided recursively as many times as necessary. In MAGRITTE, this scheme is used to obtain an angular discretization for calculating the mean line intensity J_{ij} (see Eq. 4.20). Note that we also need to specify how we discretize the frequency space for calculating J_{ij}. For this, we rely on the Gauss-Hermite quadrature in order to properly sample the intensity in each region of the line profile function ϕ_{ij} [for more details, see De Ceuster et al., 2019].

In MAGRITTE, we compute self-consistent NLTE line radiative transfer by iteratively computing the radiation field and the level populations n_i. To improve the convergence behavior of this iterative scheme, we have implemented both ALI (see Section 2.3.1) and Ng-acceleration (see Section 2.3.2).

To further improve computational efficiency, we apply model grid reduction, as point cloud distributions fit for hydrodynamics simulations might prove unnecessarily detailed when computing radiation transport [De Ceuster et al., 2020]. Starting from the original point cloud, we generate a new point cloud, containing less points compared to the original model. This increases the computational performance of the radiative transfer calculation, at the cost of a

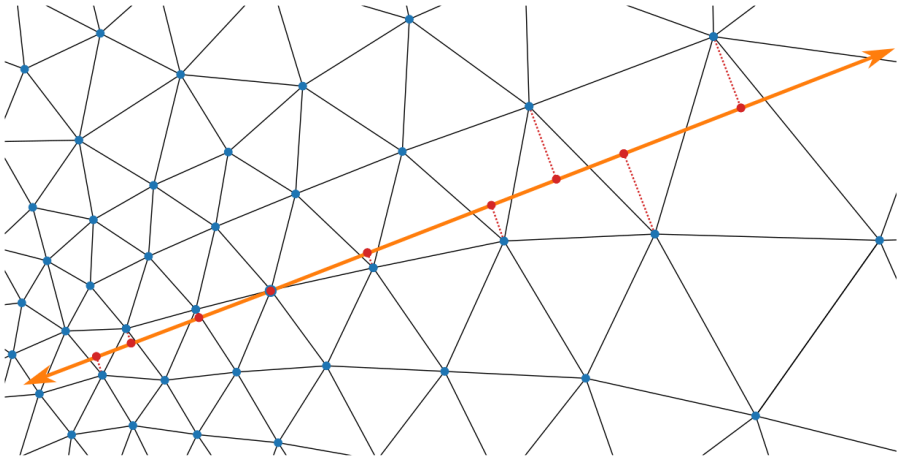


Figure 2.6: Illustration of the ray tracing algorithm used in MAGRITTE. The ray starts at a given point and keeps adding the neighbor closest to the ray to the path discretization until the model boundary is reached. Figure taken from De Ceuster et al. [2020].

reduction in accuracy. Finally, we explore further computational improvements made throughout this thesis in Section 2.4.3.

User interface 2.4.2

MAGRITTE has publicly available documentation[22], which includes installation instructions, jupyter notebooks with examples and full API definitions. For performance reasons and memory control, the code base was written in C++, but it has a python interface, made using pybind11[23], which exposes the C++ functions and data structures to python. For ease of use, the internal geometry is based on point clouds, thus it can accept both particle-based and grid-based hydrodynamics models as input[24]. In Listing 2.1, we present an example of how to create a MAGRITTE model. Then, in Listing 2.2, we use this MAGRITTE model to create synthetic channel maps, similar to Fig. 2.7.

[22] Available at `https://magritte.readthedocs.io/en/stable/`

[23] Available at `https://github.com/pybind/pybind11`

[24] Analytical models are also possible as input, but need to be converted to one of both mentioned input options.

Listing 2.1: Example PYTHON script for creating a MAGRITTE model. The hardcoded settings denote typical parameters used for a 3D MAGRITTE model.

```
import magritte.setup as setup # Model setup
import magritte.core as magritte # Core functionality
import numpy as np

model = magritte.Model ()# Create model object

model.parameters.set_model_name(model_file) # Magritte model file
model.parameters.set_dimension(3) # This is a 3D model
model.parameters.set_npoints(npoints) # Number of points
model.parameters.set_nrays(48) # Number of rays
model.parameters.set_nspecs(3) # Number of species
model.parameters.set_nlspecs(1) # Number of line species
model.parameters.set_nquads(15) # Number of quadrature points

model.geometry.points.position.set(position)
model.geometry.points.velocity.set(velocity)

model.geometry.points.neighbors.set(nbs)
model.geometry.points.n_neighbors.set(n_nbs)

model.chemistry.species.abundance = np.array((nCO, nH2, zeros)).T
model.chemistry.species.symbol = ['CO', 'H2', 'e-']

model.thermodynamics.temperature.gas.set(tmp)
model.thermodynamics.turbulence.vturb2.set(trb)

model.parameters.set_nboundary(boundary.shape[0])
model.geometry.boundary.boundary2point.set(boundary)

setup.set_uniform_rays(model)
setup.set_boundary_condition_CMB(model)
setup.set_linedata_from_LAMDA_file(model, lamda_file)
setup.set_quadrature(model)

model.write()
```

Listing 2.2: Example PYTHON script for creating a synthetic observation from a MAGRITTE model.

```
import magritte.core   as magritte  # Core functionality
import magritte.plot   as plot      # Plotting
import magritte.tools  as tools     # Save fits
import matplotlib.pyplot as plt
```

```
5
6  # load the saved model
7  model = magritte.Model(model_file)
8
9  # Initialize model setup
10 model.compute_spectral_discretisation ()
11 model.compute_inverse_line_widths ()
12 model.compute_LTE_level_populations ()
13
14 # Iterate level populations until statistical equilibrium
15 model.compute_level_populations_sparse (True, 20)
16
17 # Setup details for synthetic observations: specify frequency range
18 # For this example, we image around the 2-1 line
19 fcen = model.lines.lineProducingSpecies[0].linedata.frequency[1]
20 maxvel = 25000 #[m/s]
21
22 # Compute doppler shift corresponding to this velocity
23 dd   = maxvel / magritte.CC
24 fmin = fcen - fcen*dd
25 fmax = fcen + fcen*dd
26
27 # Create a set of images of the model using the specified
       (uniformly spaced) frequency discretization
28 model.compute_spectral_discretization(fmin, fmax, n_image_freqs =
       121)
29 model.compute_image_new(0.0, 0.0, 1.0, 512, 512)
30 #imaging in the direction (nx,ny,nz) = (0,0,1), using a resolution
       of 512x512 for the image.
31
32 #combine the images into a channel map
33 image_frequency_indices_to_use = [30+4*i for i in range(16)]
34 channel_map_shape = (4,4)
35 plot.image_channel(model, image_frequency_indices_to_use,
       channel_map_shape, zoom = 1.8, xticks=[-50,0,50],
       yticks=[-50,0,50], npix_x=512, npix_y=512)
36 plt.show()
```

Computational improvements 2.4.3

The goal for this thesis is to be able to efficiently compute NLTE line radiative transfer in the AGB environment. As stated before, 3D NLTE line radiative transfer is computationally expensive. Thus we require new optimizations, both to the exact implementation and the numerical algorithms. In Chapter 3,

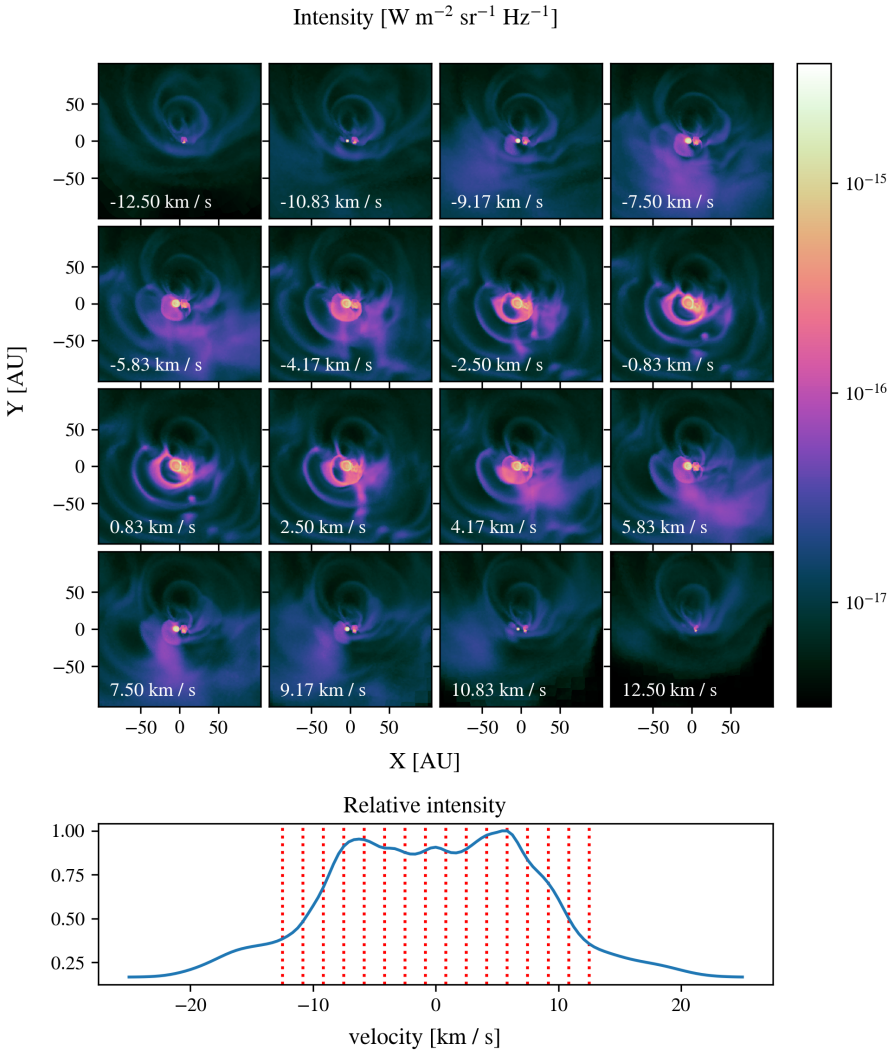


Figure 2.7: Channel map for the intensity, made using MAGRITTE by post-processing a hydrodynamics model from Malfait et al. [2021]. The different boxes correspond to material moving towards or away from the observer at the specified velocity. On the bottom, we find the line profile, which intensity per channel map integrated. Figure taken from Ceulemans et al. [2024]

we implement various improvements to the MAGRITTE code base and several new optimizations for NLTE line radiative transfer, including a variation on Ng-acceleration (see Section 3.4.2) and a semi-analytic method to calculate optical depths when encountering large velocity gradient regions (in Section 3.4.3). Then, in Chapter 4, we explore a new discretization for the comoving variation of the radiative transfer equation (Eq. 2.19), suppressing numerical instabilities, while also allowing a for non-monotonic velocity field. Furthermore, we implement an adaptive angular discretization to suppress the so-called ray-effect which originates from using a limited angular discretization when using discrete ordinates. Afterwards, we explore using GPU parallellization to speed up line radiative transfer. In Chapter 5, we port MAGRITTE to PYTORCH, a machine learning library. We translate the CPU-based algorithms from MAGRITTE to GPU. Furthermore, we explain how one can utilize the machine learning framework for uncertainty quantification. Finally, we will be able to compute the CO line cooling in Chapter 6. However, NLTE line radiative transfer is too computationally expensive to include on the fly during hydrodynamics simulations. Therefore, we propose a CO cooling rate prescription for the AGB environment, based on radiative transfer post-processing of AGB hydrodynamics simulations. Furthermore, as these simulations might not yet include all physics, we study the influence of perturbations on the radiative transfer model input to the cooling rates.

CHAPTER 3
Computational improvements for Magritte

This chapter is an updated version of:

Magritte, a modern software library for spectral line radiative transfer

Ceulemans, T., De Ceuster, F., Decin, L., Yates, J., 2024.

Astronomy and Computing Volume 49, 100889.
`https://doi.org/10.1016/j.ascom.2024.100889`

This paper was published before re-implementing interpolation, which improves the accuracy of the results, but introduces an extra computational cost. The exact timings in this chapter are therefore not representative for the latest version of MAGRITTE, but this should not affect the scaling behavior for any optimization presented in this chapter.

CRediT authorship contribution statement
T. Ceulemans: Writing – original draft, Visualization, Validation, Software. F. De Ceuster: Writing – review & editing, Supervision, Software, Funding acquisition. L. Decin: Writing – review & editing, Supervision, Funding acquisition. J. Yates: Writing – review & editing.

Abstract

Spectral line observations are an indispensable tool to remotely probe the physical and chemical conditions throughout the universe. Modelling their behaviour is a computational challenge that requires dedicated software. In this chapter, we present the first long-term stable release of MAGRITTE, an open-source software library for line radiative transfer. First, we establish its necessity with two applications. Then, we introduce the overall design strategy. Finally, we present several key improvements over previous versions: (1) an improved re-meshing algorithm to efficiently coarsen the spatial discretisation of a model; (2) a variation on Ng-acceleration, a popular acceleration-of-convergence method for non-LTE line transfer; (3) a semi-analytic approximation for line optical depths in the presence of large velocity gradients; (4) a computationally efficient evaluation of the total emissivity and opacity; and, (5) an interpolation scheme for accurately evaluting the optical depth opacity.

Introduction 3.1

Electromagnetic radiation contains a wealth of information, in particular, spectral lines. These sharp features in frequency space originate from transitions between the quantized energy levels of electronic, rotational, or vibrational states of the atoms and molecules in the medium. Their specific frequencies uniquely characterise the chemical species that produces them and thus can be used to infer the chemical composition of the medium. Spectral lines also encode the kinematic structure of the environment. Their narrow extent in frequency space makes them very sensitive to Doppler shifts caused by macroscopic velocity gradients in the medium, while their widths are governed by microscopic thermal and turbulent motions. Furthermore, the relative intensities of lines can be used to infer the thermodynamic properties of the environment. As such, spectral line are excellent tracers of both the physical and chemical properties of the medium from which they originate, making them indispensable for astrophysics research. In order to leverage all this information, over the years, several software models have been developed to simulate line radiative transfer, such as MCFOST [Pinte et al., 2006], LIME [Brinch and Hogerheijde, 2010], SKIRT [Baes and Camps, 2015, Matsumoto et al., 2023], and also MAGRITTE [De Ceuster et al., 2019, 2020, De Ceuster et al., 2022, Ceulemans et al., 2024].

In this chapter, we present the first stable release of MAGRITTE, a modern software library for spectral line radiative transfer. Previous papers about

Magritte have focused on the specific algorithms for ray-tracing, the self-consistent solution of the radiation field with the state of the medium, and benchmarks against other codes [De Ceuster et al., 2019], and also on reduction methods for the model discretisation and their effect on the computational efficiency [De Ceuster et al., 2020]. In this chapter, we focus on the now-consolidated code design and maintenance strategy for Magritte[1] while also presenting several further algorithmic improvements over previous versions [De Ceuster et al., 2022].

This chapter is organized as follows. In Section 3.2, we demonstrate the utility of Magritte with two important applications. In Section 3.3, we describe the code structure and elaborate on our software development and maintenance strategy. We present five key improvements in Magritte in Section 3.4, and we discuss current limitations and future work in Section 3.5.

Statement of need 3.2

The era of spatial and spectral high-resolution imaging, such as with the Atacama Large (sub)Millimetre Array (ALMA) or the Square Kilometre Array (SKA), has pushed the need for versatile radiative transfer simulators that can handle high-resolution models to simulate all complex phenomena that are observed [see e.g. Andrews et al., 2018, Decin et al., 2020, Öberg et al., 2021]. Magritte is a radiative transfer simulator that was primarily designed with applications in mind of modelling stellar environments. Since its conception, it has been used, for instance, to model the unusual NaCl distribution around asymptotic giant branch (AGB) star IK Tauri [Coenegrachts et al., 2023], to model the molecule-rich disk around AGB star L_2 Puppis [Van de Sande et al., 2024], and to create synthetic observations of smoothed-particle hydrodynamics (SPH) models of companion-perturbed stellar winds [Malfait et al., 2024b]. Furthermore, Esseldeurs et al. [2023] implemented the ray-tracer of Magritte in the SPH code Phantom [Price et al., 2018] to couple radiation and hydrodynamics. Below, we demonstrate how Magritte can be used to model the CO line emission of an analytic model of a protoplanetary disk and an intricate SPH model of a companion-perturbed stellar wind.

Analytic protoplanetary disk model 3.2.1

Figure 3.1 shows synthetic observations created with Magritte of the CO $J = 1 - 0$ line emission emanating from a proto-planetary disk model, viewed

[1] Publicly available at `github.com/Magritte-code/Magritte`.

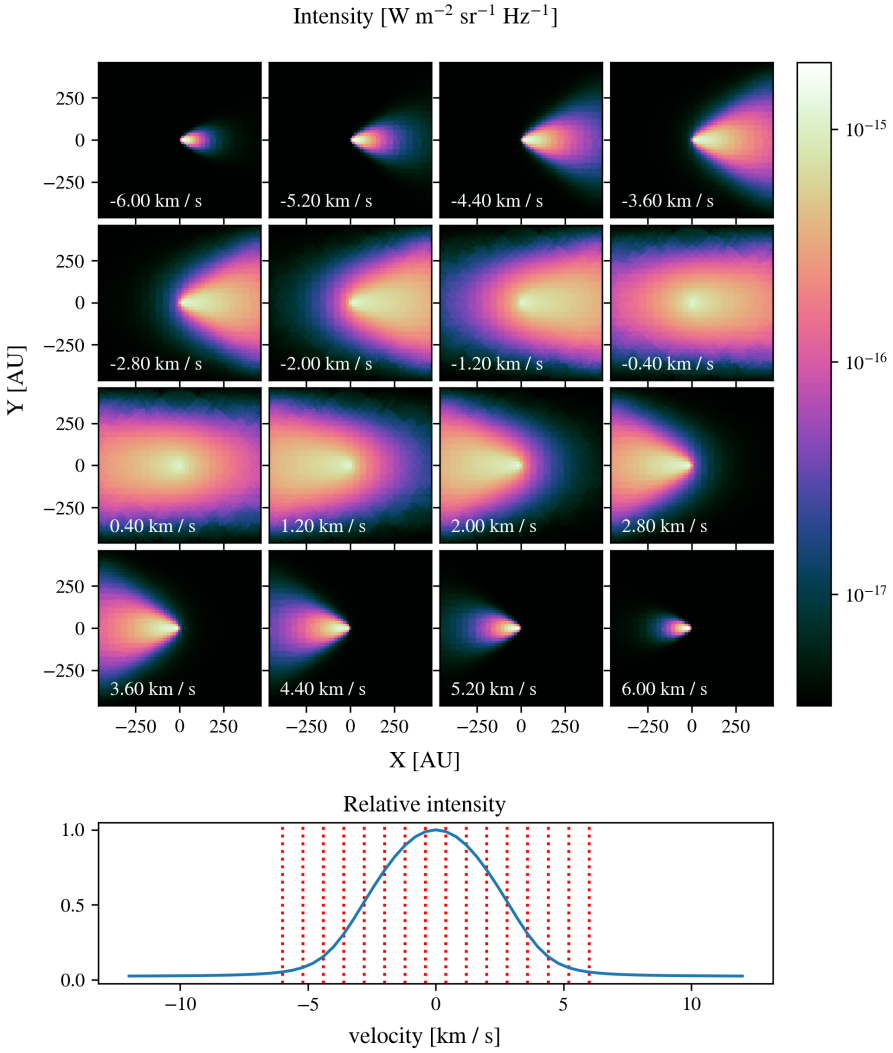


Figure 3.1: Synthetic observations of the CO $J = 1 - 0$ line of an analytic protoplanetary disk model. *(Top.)* Specific intensity in each frequency/velocity bin. *(Bottom.)* Normalized spatially integrated intensity.

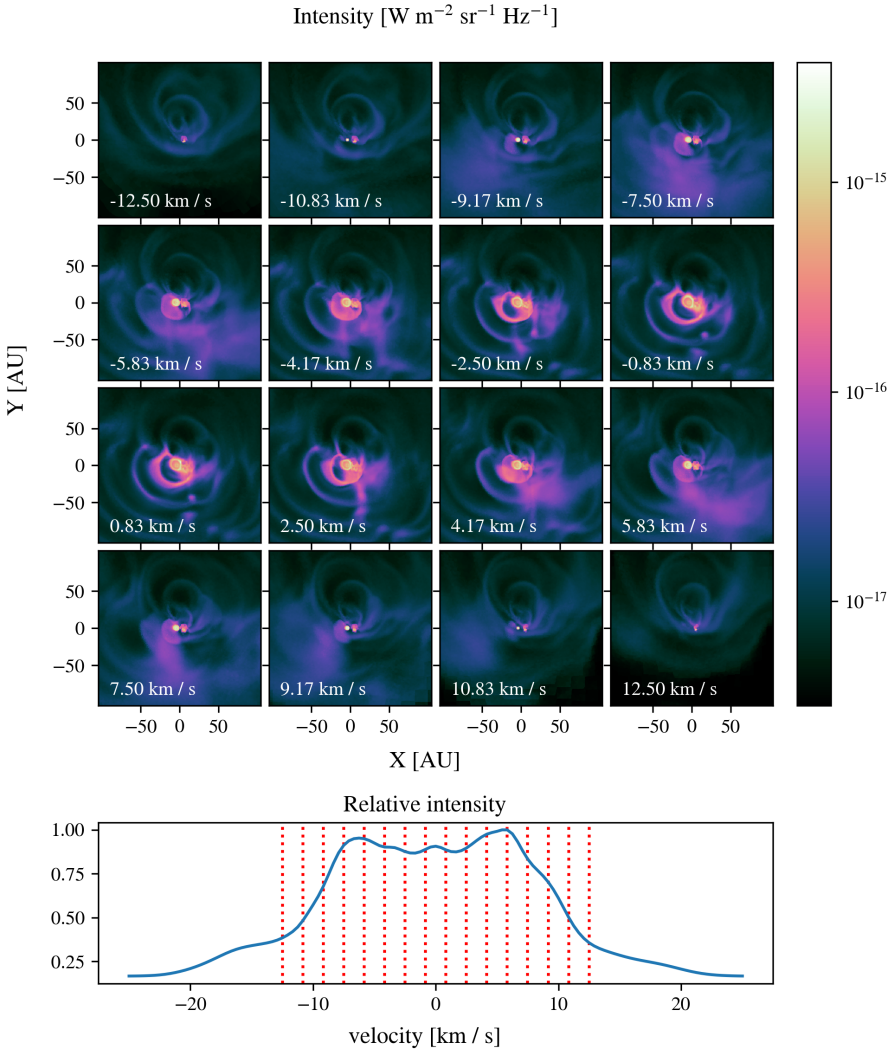


Figure 3.2: Synthetic observations of the CO $J = 2-1$ line of an SPH simulation of a companion-perturbed stellar wind model. *(Top.)* Specific intensity in each frequency/velocity bin. *(Bottom.)* Normalized spatially integrated intensity.

edge-on. A detailed description of the model can be found in [De Ceuster et al., 2019][2]. The MAGRITTE model used a discretisation of about 100k points, 48 directions, and 7 frequency bins per line. The radiative properties of the medium[3] were determined self-consistently with the radiation field, without assuming local thermodynamic equilibrium (i.e. non-LTE). The corresponding channel maps are shown in Figure 3.1. These show the intensity in different frequency bins, which, through the Doppler shift, can be related to the emission of material moving with a specific velocity along the line of sight. When scanning through the different channels from top-left to bottom-right, we can see the emission shift from right to left with respect to the centre of each image. This is a clear signature of the rotation of the disk. The detailed structure of the line emission of the disk can further be used to infer physical and chemical properties, such as the velocity and temperature distribution [see e.g. Van de Sande et al., 2024].

Numerical companion-perturbed stellar wind model 3.2.2

Figure 3.2 shows synthetic observations created with MAGRITTE of the CO $J = 2 - 1$ line emission emanating from an SPH simulation of a companion-perturbed stellar wind. The line-of-sight is orthogonal to the orbital plane of the companion. A star in the centre is losing mass in an initially spherically-symmetric wind, stirred up by a companion on an eccentric orbit. Figure 3.3 depicts a slice through the centre of the model, showing the intricate wind structure in terms of the CO number density.[4] A detailed description of the SPH model can be found in Malfait et al. [2021]. The SPH model, as processed by MAGRITTE, contains about 1M points, with 48 directions and 7 frequency bins per line to discretise the radiation field. Using MAGRITTE, first, non-LTE line transfer is performed to determine the level populations of the CO rotational levels, and then the images are created. Given the complexity of the model, the interpretation of the synthetic observations is less straightforward as with the previous disk model. The channel maps in Figure 3.2 clearly show the arcs that are stirred up by the companion in the stellar wind. In the higher-velocity

[2]De Ceuster et al. [2019] also showed synthetic observations of this model. However, due to boundary effects, the integrated line profile was inaccurate. This has now been resolved. The computational improvements presented in this chapter, moreover, allow us now to create much higher resolution images.

[3]This means the emissivities/opacities computed using the populations of the 41 lowest rotational energy levels of CO. As for every model in this chapter, the line data were obtained from the LAMDA database, which can be found at `home.strw.leidenuniv.nl/~moldata/`.

[4]To obtain the CO number density from the SPH model (that only contains the total mass density), we attribute all the mass density to H_2 and then assume a constant ratio of 10^{-4} between the CO and H_2 number densities.

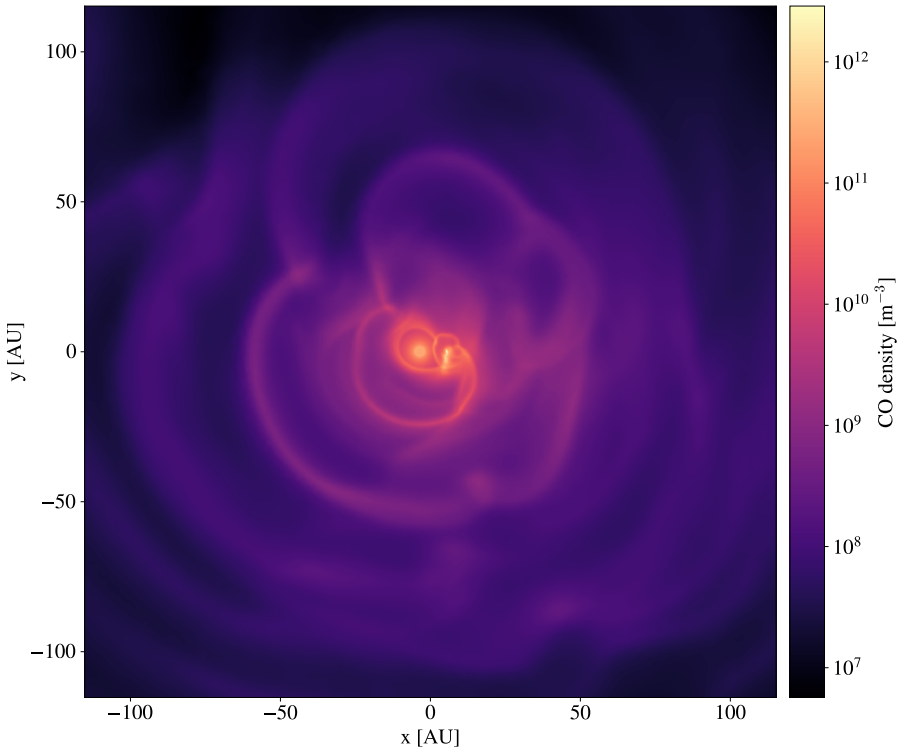


Figure 3.3: Slice through the centre of the companion-perturbed stellar wind model showing the CO number density.

images we can even discern the mass-losing star and its companion through the hot and fast-moving material that enshrouds them. The velocity structure is quite symmetric, due to the initially spherically-symmetric velocity field of the wind and the fact that the line-of-sight is orthogonal to the orbital plane.

Magritte design 3.3

The design of Magritte is mainly driven by the need for a modular and performant general-purpose radiation transport simulator that can easily be applied to grid-based as well as particle-based model data. The resulting design choices for the geometric data structures, the model discretisation, and the ray-tracing algorithm have extensively been discussed in De Ceuster et al. [2019,

2020] and De Ceuster [2022]. Here, we focus on the code structure, and the development and maintenance strategy. The application / programmer interface (API) has already been explained in Section 2.4.2.

Code structure 3.3.1

The core of MAGRITTE is written in C++, but it has a convenient user-interface in PYTHON that is linked to the C++ core using PYBIND11.[5] The MAGRITTE core is written in an object-oriented fashion, organising model data in a physically relevant class structure. The main class structure of a MAGRITTE model is given in Listing 3.1. The PARAMETERS class stores the general model parameters, such as a name for the model, the number of points, directions, and frequency bins, as well as algorithmic parameters, such as various threshold values. Since all classes require this data, each model subclass has access to the PARAMETERS class through a C++ smart pointer (`std::shared_ptr`). The contents of the other classes should be evident from their names and is further illustrated in the exposition of the API.

Listing 3.1: Model class structure in MAGRITTE.

```
Model.Parameters

Model.Geometry
Model.Geometry.Boundary
Model.Geometry.Points
Model.Geometry.Rays

Model.Lines
Model.Lines.LineProducingSpecies

Model.Thermodynamics
Model.Thermodynamics.Temperature
Model.Thermodynamics.Turbulence

Model.Radiation
Model.Radiation.Frequencies

Model.Chemistry
Model.Chemistry.Species
```

[5] Publicly available at `github.com/pybind/pybind11`.

Code development & maintenance strategy 3.3.2

MAGRITTE is a collaborative project that has multiple developers and several contributors. To facilitate the collaborative development process, we adopted the following strategy.

Version control

In MAGRITTE, we use GIT for version control. The repository is hosted online on GITHUB. Different projects and features are developed in different branches. The latest stable version, the one that is recommended for scientific use, can be found on the STABLE branch.

To facilitate communication regarding updates, an automated versioning scheme has been implemented. This means that whenever the STABLE branch is updated, the version number of the source code will increase automatically. This provides users with a consistent and human-readable way to determine code versions, in addition to the less human-readable git hashes.

Automated testing

In order to swiftly catch the unavoidable errors that will be introduced by continued development on an existing code base, we introduced an automated testing framework in MAGRITTE. We use PYTEST for defining the test cases and automate them in our online repository using GITHUB ACTIONS. We currently do not have unit tests but only regression tests. The tests compare results obtained with MAGRITTE against analytical results when possible, and, otherwise, with previously-obtained and validated numerical results. The analytical tests that we now automated have previously been described in De Ceuster et al. [2019]. To test the non-LTE solver, we use the benchmarks described in van Zadelhoff et al. [2002]. We also include one "real-world" scientific application in our test suite, namely, the imaging of the SPH model presented in Section 3.2.2.

We support MAGRITTE both on Linux and MacOS and thus also test on both operating systems.[6] We noticed that minor differences can be found between the online numerical test results on the two operating systems, mainly because the MacOS version is executed on ARM M1 Apple silicon, on which the C++ LONG DOUBLE type has only 64 bits, while it has 80 bits on the X86 hardware

[6] Windows users are encouraged to use MAGRITTE through the Windows Subsystem for Linux (WSL).

on which the Linux tests are executed. Despite the numerical difference, both solutions safely reside within the assumed error margins.

New features & Improvements 3.4

Since the last publication about MAGRITTE [De Ceuster et al., 2022], some new features and several improvements have been introduced. We highlight the most important ones below.

Recursive re-meshing 3.4.1

As in many numerical methods, also in radiation transport, discretisation plays a key role in the trade-off between accuracy and computational speed. The computational cost of a ray-tracing radiative transfer solver roughly scales as $N^{4/3}$, with, N, the number of points in the spatial discretisation, while it scales linearly with respect to other parameters, such as the directional and frequency discretisation. Therefore, it is crucial to be able to optimally choose the spatial discretisation. De Ceuster et al. [2020] proposed a simple algorithm to tailor the spatial discretisation of a hydrodynamics model to radiative transfer simulations. The algorithm coarsened the spatial discretisation, based on a criterion on the gradient of a so-called tracer function, i.e. the function we want to properly sample with the (reduced) discretisation. This was implemented in MAGRITTE, using the mesh-generation functionality provided by GMSH[7] [Geuzaine and Remacle, 2009] and SCIPY [Virtanen et al., 2020]. Although effective, generating new discretisations was computationally expensive [it was e.g. a limiting factor in the work by Coenegrachts et al., 2023], and the implementation was rather cumbersome. To alleviate this, we have implemented a much faster and simpler recursive re-meshing algorithm that we describe below.

Following De Ceuster et al. [2020], we assume the density, ρ, of a model to be an appropriate tracer function, i.e. we search for a coarser spatial discretisation that still properly samples the density. Note, however, that we can replace this function by any other (combination of) model function(s), if needed. In accordance with the internal data structures in MAGRITTE, we assume the density to be given on a point cloud. The recursive re-meshing algorithm then goes as follows. Consider a rectangular box containing all data points and compute the variation of the tracer function within that box. We define the variation within a box as

$$2\left(\frac{\rho_{\max}-\rho_{\min}}{\rho_{\max}+\rho_{\min}}\right), \tag{3.1}$$

[7] Publicly available at: `gmsh.info/`.

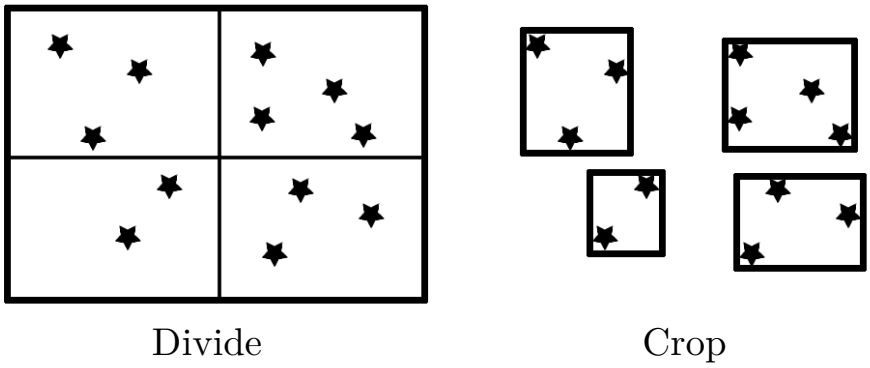


Figure 3.4: Illustration of the "divide" and "crop" steps in the recursive re-meshing algorithm. The stars denote the data points of the point cloud.

where $\rho_{\max}$ and $\rho_{\min}$ are respectively the maximum and minimum value of the tracer function attained within the box. If this variation exceeds a certain threshold value, $r_{\max}$, then we deem the variation too large and subdivide the rectangular box in 4 (in 2D) or 8 (in 3D) equal rectangular boxes. Then, if possible, we crop each rectangular box to its minimal size still containing all points belonging to that box. These "divide" and "crop" steps are illustrated in Figure 3.4. This procedure is then recursively repeated for all boxes in which the variation exceeds the threshold, until a maximum level of recursion, l, is reached. Once the recursion halts, we find the resulting reduced discretisation by representing each box by a single point at its centre. To ensure a proper definition of the boundary, after recursive re-meshing, we add an evenly-spaced set of boundary points on the surface of the outermost box.

The recursive re-meshing method has two parameters that determine the resulting discretisation, the maximum variation within a box, $r_{\max}$, and the maximum recursion level l. The former dictates the smoothness of the tracer function, while the latter limits the overall recursion level, thereby limiting the smallest resolvable length scale. Increasing $r_{\max}$ will result in a smoother representation of the tracer function, while increasing l will result in better-resolved small-scale details. We have implemented this recursive re-meshing algorithm efficiently in Python in Magritte, using standard numpy array operations [Harris et al., 2020][8].

[8]The implementation can be found in the function remesh_point_cloud in the file magritte/mesher.py in the Magritte code base.

Table 3.1: Comparing the number of points, time it takes to create a reduction, and time it takes to perfom a radiative transfer simulation.

	N_{points}	$t_{\text{reduction}}$ [s]	t_{RT} [s]
Original	$1.1 \cdot 10^6$	–	$5.2 \cdot 10^3$
Reduced old	$9.2 \cdot 10^4$	$2.2 \cdot 10^2$	$1.9 \cdot 10^2$
Reduced new	$8.5 \cdot 10^4$	$6.0 \cdot 10^0$	$2.3 \cdot 10^2$

We can now demonstrate the utility of this new method by applying it to the SPH model of a companion-perturbed stellar wind, introduced in Section 3.2.2. We chose the parameter values, $r_{\max} = 0.4$ and $l = 12$, to obtain similar resulting discretisations as the old re-meshing algorithm from De Ceuster et al. [2020], so we can compare the two. The results are summarised in Table 3.1. Both methods reduce the number of points by about one order of magnitude. However, we can see that the new method is about 50 times faster. We note that, even though the new method results in slightly fewer points, the time it takes to perform a radiative transfer simulation on it is slightly longer. This can be understood, since the new method creates a slightly more regular discretisation, resulting on average in slightly more points per ray. Nevertheless, radiative transfer simulations are still an order of magnitude faster compared to the original model. As a side note, we find that the GMSH reduction takes a similar amount of time as the corresponding radiative transfer simulation. We think that the required time for this remeshing method is proportional to original model size, given that we first create a mesh from the original point cloud, and give that as input to GMSH.

Figure 3.5 shows the three different discretisations, the corresponding synthetic observations of the CO ($J = 2 - 1$) line, and the relative differences of these with respect to the original model. Qualitatively, both reduced discretisations trace the underlying tracer function equally well and both yield equally accurate synthetic observations. To quantitatively asses the accuracy of the reduced discretisations, following De Ceuster et al. [2020], in Figure 3.6, we plotted the cumulative distribution of relative differences between the radiation fields computed on the reduced discretisations and the original model. Since for non-LTE line transfer the direction- and frequency-averaged mean intensity is important, we have plotted the relative differences for the mean intensity as well as for the resulting intensity on the images. For both quantities, the relative differences are below 10% for 80% of the points in the model or pixels in the image. The new method can thus produce equally good reduced discretisations at a fraction of the computational overhead.

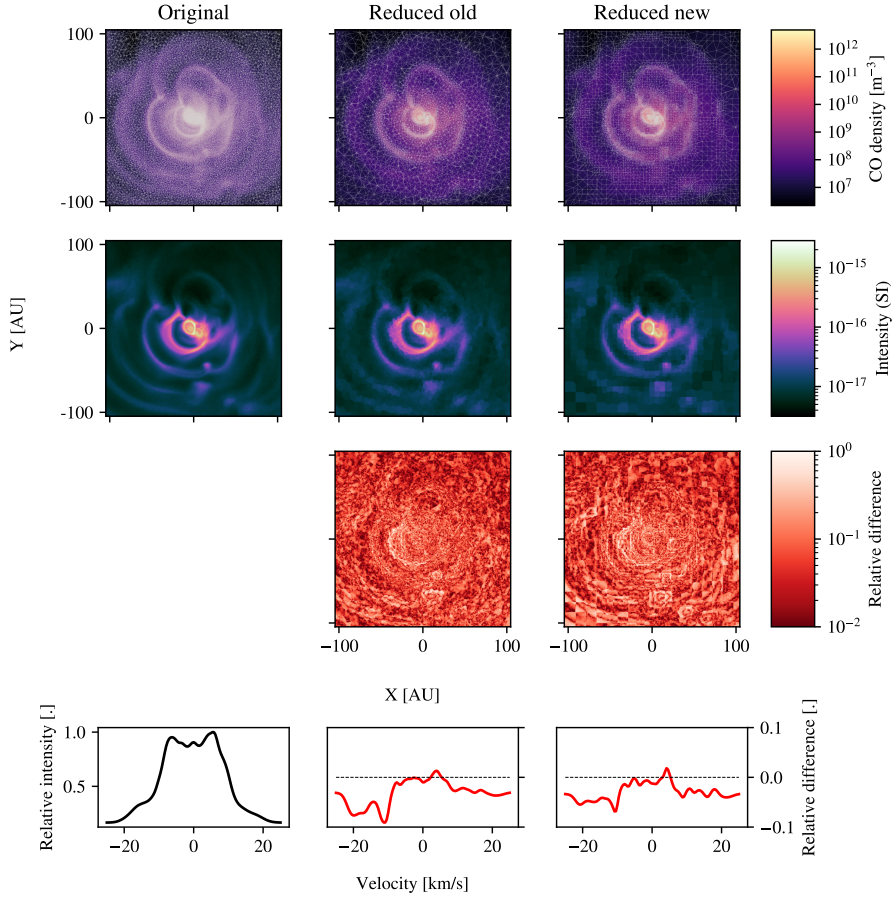


Figure 3.5: *(Top row.)* Slice through the centre of the companion-perturbed stellar wind model showing the CO number density (cfr. Figure 3.3), but now with a Delaunay tetrahedralisation of the spatial discretisation superimposed. *(Second row.)* Synthetic observations of the CO $J = 2 - 1$ line, showing only $v = 0$ km/s, created with MAGRITTE, using the different spatial discretisations. *(Third row.)* The relative difference in the synthetic observations, compared to the original model. *(Bottom row.)* Integrated line profile obtained from the original model, and the relative differences for the reduced discretisations compared to the the original model.

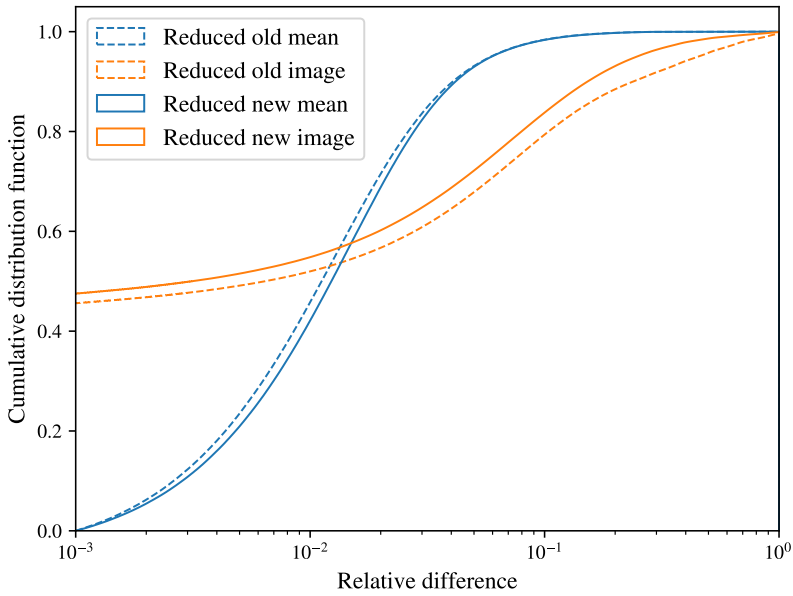


Figure 3.6: Cumulative distributions of the relative differences between results obtained on the reduced discretisations and the original model.

Adaptive Ng-acceleration 3.4.2

In line radiative transfer, the emissivity and opacity of the medium depend on the radiation field in a non-trivial way. To resolve this interdependence, the state of the medium and the radiation field are often alternately solved in an iterative way. The convergence of this iterative process can be notoriously slow, and, as a result, various techniques have been devised to accelerate it [see e.g. Olson et al., 1986, Rybicki and Hummer, 1991]. One of these techniques is Ng-acceleration [Ng, 1974, Olson et al., 1986]. This is a general acceleration-of-convergence technique and is known under various other names and has various related variants in other application domains [see e.g. Sidi, 2020]. It aims to predict the fixed-point of the iterative process by a linear combination of the N_{steps} last iterates. In the classical formulation, the value of N_{steps} is a fixed predefined hyper-parameter, and it was also implemented like this in MAGRITTE [De Ceuster et al., 2019].

We noted, however, that this value can adaptively be chosen to obtain even better convergence. We call this "adaptive Ng-acceleration", and we determine N_{steps} as follows. After each regular iteration, we compute the Ng-prediction, using the last N regular iterates.[9] Then, we decide whether or not to use this prediction as the next iteration step. The criterion depends on the convergence of the running Ng-predictions versus the convergence of the regular iterates. We only use the Ng-prediction as the next iterate if the maximum relative change in the last two running Ng-predictions is smaller than the maximum relative change in the last two regular iterates. Stated differently, we only use the Ng-prediction as the next iterate if the Ng-predictions converge faster than the regular iterates. In this way, we keep on using more previous iterates in the Ng-prediction, i.e. N keeps on growing, until we eventually use the Ng-prediction. If the criterion remains unsatisfied for many iterations, of course, we cannot keep on storing the previous iterates indefinitely. Therefore, we also introduce a maximum, N_{max}, after which we use the Ng-prediction as new iterate anyway.

This adaptive version of Ng-acceleration requires a way to conveniently compute the Ng-prediction for any number N of previous iterations. We present only the resulting equations here and provide a full derivation in 2.3.2. The Ng-prediction for the fixed-point of an iterative process is a linear combination of the N last iterates,

$$\boldsymbol{n}_* = \sum_{n=1}^{N} c_n \, \boldsymbol{n}_{-n}, \tag{3.2}$$

[9] To be precise, we only include iterates after the last Ng-prediction, if a previous Ng-prediction was already made.

where we used negative indices to denote elements, starting from the last one counting backwards (like in PYTHON). To ensure normalisation of the prediction, which is important e.g. for level populations, we require the coefficients, c_n, to sum to one. Defining the residuals, $\boldsymbol{r}_n \equiv \boldsymbol{n}_{-n} - \boldsymbol{n}_{-n+1}$, and accumulating them in a matrix,

$$R^{\mathrm{T}} \equiv \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \dots & \boldsymbol{r}_{N-1} \end{bmatrix} \tag{3.3}$$

the resulting coefficient vector, $\boldsymbol{c}$, for the Ng-prediction reads,

$$\boldsymbol{c} = \frac{\left(RR^{\mathrm{T}}\right)^{-1}\mathbf{1}}{\mathbf{1}^T\left(RR^{\mathrm{T}}\right)^{-1}\mathbf{1}}, \tag{3.4}$$

in which $\mathbf{1}$ denotes a $(N-1)$-dimensional vector of ones. Note that the matrix that needs to be inverted is $(N-1 \times N-1)$-dimensional and does not depend on the dimension of the vector iterates, $\boldsymbol{n}$. Since we expect to require only a relatively small number of iterates, we expect the matrix inversion to be cheap.

We can demonstrate the performance of our adaptive Ng-acceleration scheme on the non-LTE benchmark problems by van Zadelhoff et al. [2002]. We consider what they call "problem 1a/b", which describes non-LTE line transfer in a spherically-symmetric steady-state model for a fictitious two-level species. The a/b refers to the density of the fictitious species, resulting in a low/high optical depth medium. Hence, we expect convergence for problem 1a, with lower optical depth, to be faster than for 1b. Figure 3.7 shows the convergence behaviour of classical and adaptive Ng-acceleration on this benchmark problem, for several choices of N_{steps}, in the classical, and N_{max}, in the adaptive algorithm. Adaptive Ng-acceleration consistently shows faster convergence. Also when N_{max} is chosen relatively large (e.g. 16 or 32), the algorithm timely injects the Ng-prediction for optimal convergence. As a result, for large enough N_{max}, it does not affect the convergence behaviour anymore. The choice of N_{max} should therefore only be informed by the amount of memory available.

Semi-analytical approximation for line optical depths 3.4.3

Optical depth plays a crucial role in radiative transfer, as it defines the relevant length-scales. In a spatial discretisation, the optical depth between two neighbouring points, $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$, is defined in terms of the opacity, χ, as,

$$\Delta\tau = \Delta x \int_0^1 \mathrm{d}\lambda\ \chi(\lambda), \tag{3.5}$$

where, $\lambda \in [0,1]$, is a parameter interpolating between $\boldsymbol{x}_0$ at $\lambda = 0$, and $\boldsymbol{x}_1$ at $\lambda = 1$, and $\Delta x \equiv ||\boldsymbol{x}_1 - \boldsymbol{x}_0||$, is the Euclidean distance between the two points.

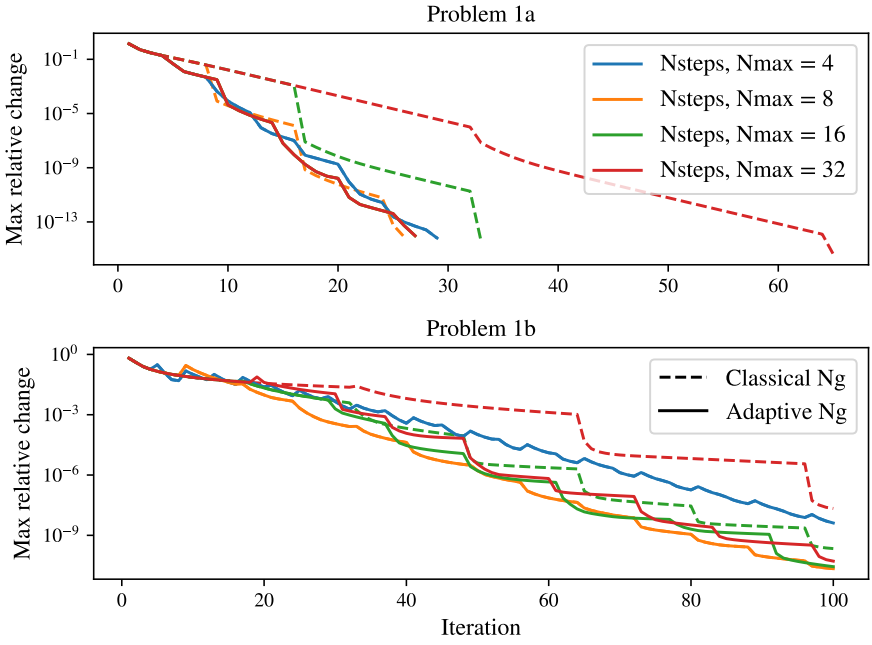


Figure 3.7: Convergence behavior of the classical and adaptive Ng-acceleration method applied to the van Zadelhoff et al. [2002] problem 1a/b, using different values for N_{steps} and N_{max}. Both legends are for both plots. Ng-acceleration occurs at the sudden changes in max relative change.

In previous versions of MAGRITTE [De Ceuster et al., 2019], the integral in the optical depth equation (3.5) was approximated using the trapezoidal rule,

$$\Delta\tau = \Delta x \left(\frac{\chi_0 + \chi_1}{2} \right), \tag{3.6}$$

which implicitly assumes that the opacity can be well-approximated by a linear function between the two points. The subscripts indicate where the function is evaluated, i.e. $\chi_i \equiv \chi(\boldsymbol{x}_i)$. As long as the opacity is well-sampled by the spatial discretisation, this is a valid assumption. However, this becomes difficult for line opacities when large velocity gradients are involved. The line opacity of a transition $i \leftrightarrow j$, can be written as,

$$\chi = \chi_{ij}\phi_{ij}(\nu), \tag{3.7}$$

where the first factor only depends on the position in the medium and the second factor is the line profile function that carries all frequency dependence. In MAGRITTE, we assume the line profile to be dominated by Doppler shifts caused by thermal and turbulent motions in the medium. This results in a Gaussian line profile,

$$\phi_{ij}(\nu) = \frac{1}{\delta\nu_{ij}\sqrt{\pi}} \exp\left[-\left(\frac{\nu - \nu_{ij}}{\delta\nu_{ij}} \right)^2 \right], \tag{3.8}$$

where ν_{ij} is the line centre, and the line width is defined as,

$$\delta\nu_{ij} = \frac{\nu_{ij}}{c} \sqrt{\frac{2k_\mathrm{B}T}{m_\mathrm{spec}} + v_\mathrm{turb}^2}, \tag{3.9}$$

with k_B Boltzmann's constant, T the local gas temperature, m_spec the molecular mass of the gas, and v_turb the turbulent velocity. The line profile is typically very narrow in frequency space and, therefore, very sensitive to velocity gradients in the medium due to Doppler shifts. For the optical depth increment (3.5), computed at a particular frequency, ν, this implies,

$$\Delta\tau(\nu) = \Delta x \int_0^1 \mathrm{d}\lambda \; \chi_{ij}(\lambda)\, \phi_{ij}\left(\nu + \lambda\Delta\nu\right), \tag{3.10}$$

where $\Delta\nu$ is the Doppler shift, caused by the velocity difference, $\Delta\mathbf{v}$, between $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$, projected along a direction $\hat{\boldsymbol{n}}$,

$$\Delta\nu = \frac{\hat{\boldsymbol{n}} \cdot \Delta\mathbf{v}}{c}\, \nu. \tag{3.11}$$

The problem is that the integrant in (3.10) is not necessarily well-approximated by a linear function, as illustrated in Figure 3.8. In previous versions of

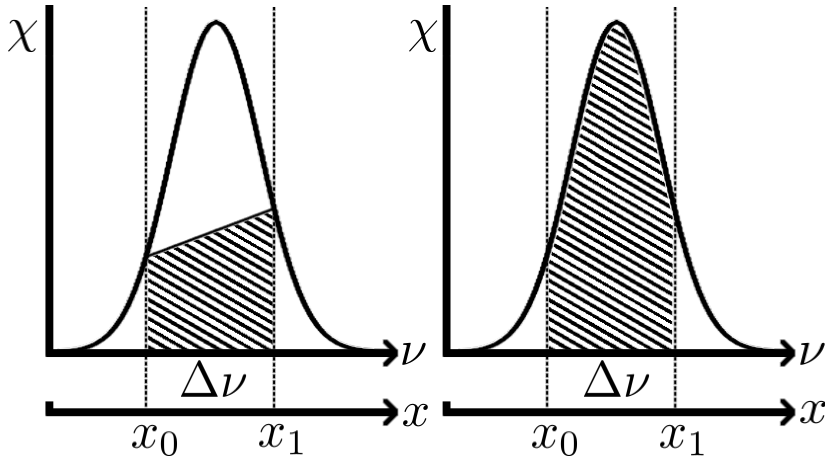


Figure 3.8: Illustration of the effect of a velocity gradient on line optical depth. *(Left.)* The resulting integral assuming a linear integrant. *(Right.)* The resulting integral by analytic integration of the Gaussian factor.

Magritte, this was resolved by subdividing the interval between neighbouring points if the Doppler shift was too large, such that on each subdivision the integrant was approximately linear [De Ceuster et al., 2020]. Although effective, when large velocity gradients are present, many subdivisions will be necessary, causing significant computational overhead. Therefore, in the current version of Magritte, we provide a semi-analytical approximation for the non-linear part of the integrant. We do this by approximating χ_{ij} by its average between the points, and analytically integrating the remaining Gaussian factor,

$$\Delta\tau(\nu) = \Delta x \left(\frac{\chi_{ij,0} + \chi_{ij,1}}{2}\right) \int_0^1 \mathrm{d}\lambda \; \phi_{ij}\big(\nu + \lambda\Delta\nu\big), \tag{3.12}$$

$$= \Delta x \left(\frac{\chi_{ij,0} + \chi_{ij,1}}{2}\right) \left(\frac{\mathrm{Erf}\left(\frac{\nu_{ij}+\Delta\nu-\nu}{\delta\nu_{ij}}\right) - \mathrm{Erf}\left(\frac{\nu_{ij}-\nu}{\delta\nu_{ij}}\right)}{2\Delta\nu}\right). \tag{3.13}$$

In principle, the line width is also position-dependent but we assume that it can be well-approximated using the average local gas temperature and turbulent velocity between the two points. Equation (3.13) allows us to accurately account for Doppler shifts of the line profile at a small and fixed computational cost.

Figure 3.9 shows the relative error and computational cost for the line optical depth computation using the trapezoidal rule, the trapezoidal rule with subdivision, and the semi-analytical method. With the semi-analytic

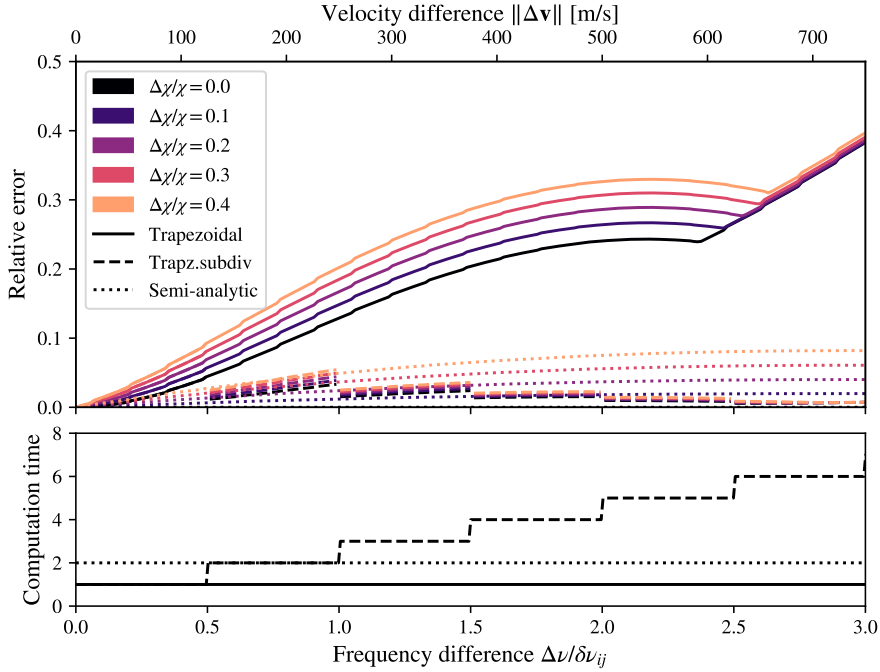


Figure 3.9: Relative error and computational cost of computing line optical depths, comparing the trapezoidal rule, the trapezoidal rule with subdivision, and the semi-analytic method.

method, we achieve comparable accuracy as with the trapezoidal rule with subdivision, but at a lower computational cost, especially for large velocity gradients. The semi-analytic method is about twice as slow as the trapezoidal rule (without subdivision). Therefore, in MAGRITTE, for small Doppler shifts, i.e. $\Delta\nu/\delta\nu_{ij} < 0.35$, we still use the trapezoidal rule, while for larger Doppler shifts, we revert to the semi-analytical method to compute line optical depths.

Efficiently computing line opacities/emissivities 3.4.4

In MAGRITTE, we are able to compute radiative transport with overlapping lines. For computational efficiency, we do not want to take into account all lines in every computation. This computational consideration is also made in other line radiative transfer codes, see e.g. FASTWIND [Puls et al., 2005]. In MAGRITTE, when computing opacities and emissivities for specific frequencies,

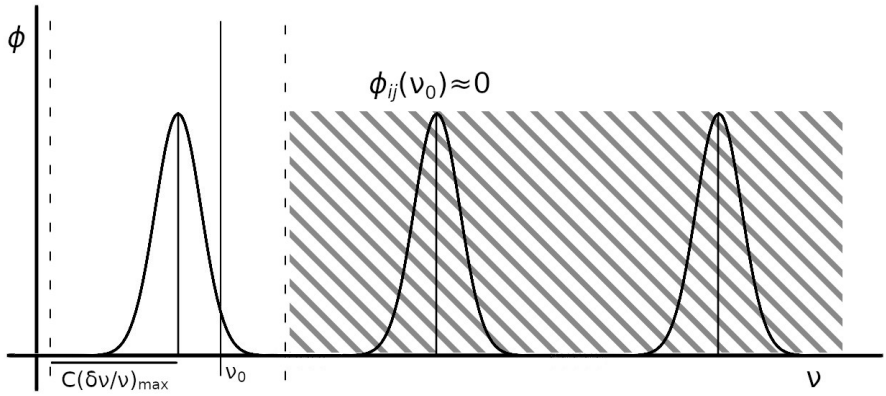


Figure 3.10: When computing the opacity/emissivity for a specific frequency ν_0, far lines (in the shaded region) can be ignored, given the locality of the influence of the line profiles ϕ_{ij}. Each line should only be taken into account if ν_0 lies within the bounds given by Eq. 3.15.

only nearby lines in frequency space are taken into account. Since the influence of a line on these values is proportional to the evaluation of the line profile function ϕ_{ij}, which has a short influence in frequency space, faraway lines do not significantly contribute to the opacities and emissivities (as illustrated in Fig. 3.10). In this section, we will describe the implementation of this optimization in MAGRITTE.

To determine the lines that significantly contribute to the opacity/emissivity calculation at the comoving frame frequency ν, we employ a heuristic approach. We precompute a maximal relative line width $\left(\frac{\delta\nu}{\nu}\right)_{\text{max}}$ at each point, defined as:

$$\left(\frac{\delta\nu}{\nu}\right)_{\text{max}}(x) = \max_{ij\in\text{lines}} \frac{\delta\nu_{ij}}{\nu}. \tag{3.14}$$

By using this value, we establish lower and upper bounds to identify lines whose frequency lies outside these limits, making their contribution negligible:

$$\left[\nu\left(1 - C\left(\frac{\delta\nu}{\nu}\right)_{\text{max}}\right), \nu\left(1 + C\left(\frac{\delta\nu}{\nu}\right)_{\text{max}}\right)\right]. \tag{3.15}$$

Here, C represents a chosen constant, set to 10 as the default value in MAGRITTE. Using this value for the constant C, we guarantee that any evaluation of the

Table 3.2: Computation time required for achieving full convergence in the Van Zadelhoff 2 models, comparing the calculation using all lines with only using nearby lines (see Section 3.4.4). As the performance improvement differs significantly between 1D and 3D models, a single 3D version of the Van Zadelhoff 2a model has been included in this comparison.

Model	All lines [s]	Nearby lines [s]
1D Van Zadelhoff 2a	$2.5 \cdot 10$	$1.5 \cdot 10$
1D Van Zadelhoff 2b	$7.3 \cdot 10$	$4.3 \cdot 10$
3D Van Zadelhoff 2a	$2.8 \cdot 10^3$	$8.8 \cdot 10^2$

exponential in the profile function is less than 10^{-40} when the line frequency ν_{ij} lies outside these bounds. Any evaluation involving a multiplication with the line profile function, i.e. when computing single line emissivities and opacities according to Eqs. 2.11 and 2.12, will therefore be deemed negligible if the line frequency ν_{ij} falls outside this range. By sorting the line frequencies in advance, standard search algorithms can be employed to determine which lines lie within these bounds.

To show the impact of this optimization on the computation time, we will apply it to the Van Zadelhoff problem 2 benchmark models [van Zadelhoff et al., 2002], which represent a collapsing cloud consisting of HCO^+. The benchmarks are divided into two variants: an optically thin (referred to as 'a') and an optically thick (referred to as 'b') model. We should note that the HCO^+ lines do not overlap in these models. We therefore expect a significant performance improvement by applying this optimization. Even though these models have a relatively modest number of lines, specifically $N_{\text{lines}} = 20$, this optimization results in noteworthy improvements in computation time (see Table 3.2). We remark that the magnitude of performance improvement differs significantly between the 1D and 3D models. This difference may be attributed to variations in model size, where smaller 1D models experience relatively more overhead from other aspects of the Λ-iteration process.

An analogous improvement can be applied to the optical depth computation in Section 3.4.3, involving the line profile function integral. The bounds are slightly extended to account for the Doppler shift between successive points. We use the comoving frame frequencies at both points, denoted as ν_0 and ν_1, to compute the bounds. The extended bounds are given by:

$$[\min(\nu_0, \nu_1) - C\delta\nu_{\max}, \max(\nu_0, \nu_1) + C\delta\nu_{\max}]\,. \tag{3.16}$$

Here, $\delta\nu_{\max}$ is the maximum line width $\delta\nu$ over both points, which ensures the accuracy of the computation.

Interpolation 3.4.5

In previous versions of MAGRITTE, interpolation was used when the Doppler shift was too high. However, in Section 3.4.4, we improved the evaluation of the optical depth in high velocity gradient regimes, and therefore decided to disable this feature.
Some time later, we discovered that some NLTE models generated nonphysically high intensities in a ring on the outside of the model. After inspection, we found that the culprit regions all were near masering: due to the incoming radiation, the local opacity was almost zero, but still positive. This caused numerical issues when calculating the intensity generated in a single ray segment (assuming optically thin, zero velocity field):

$$S_0 \Delta\tau \simeq \frac{\eta_0}{\chi_0} \Delta x \left(\chi_0 + \chi_1\right) \tag{3.17}$$

in which $*_i$ represents the value at position i and the distance for the segment between point 0 to point 1 is Δx. In case of a significant difference[10] between the opacity at nearby points, this can lead to a severe overestimation [as has previously been noticed by Pauldrach et al., 2001, Hoffmann et al., 2014]. In the references, this has been solved by slightly adapting the numerical scheme for solving radiative transfer. However, their mitigation method directly impacts the numerical discretization of the equations and is therefore not algorithm independent. We have opted to instead add interpolation points when necessary. The amount is extra interpolation points is given by

$$\left\lfloor \left| \ln\left(\frac{\chi_{ij,0}}{\rho_0} \Big/ \frac{\chi_{ij,1}}{\rho_1} \right) \right| \Big/ \ln C \right\rfloor , \tag{3.18}$$

in which ρ_* is the species number density at position $*$ and C is the maximal multiplication factor allowed between each point, which we take to be 1.4 by default. During interpolation, we interpolate the emissivities and opacities logarithmically and the line width $\delta\nu_{ij}$ linearly. Note that this criterion has been defined on a line-by-line basis; as different lines might encounter (almost) masering regimes at different positions[11], the interpolation needs to be done seperately for the different lines, as we can otherwise waste up to a factor N_{lines} computation time.

[10] In our testing, we found differences of several orders of magnitude. We note that this high difference might originate from not yet including any dust opacity.

[11] The density, temperature and intensity field affect all energy states differently. Therefore the corresponding energy transitions will also be in a different state.

Current limitations & future work 3.5

MAGRITTE currently only supports line radiative transfer without any continuum emission or absorption within the medium.[12] Although this might be a safe assumption when modelling low-J lines in stellar atmospheres, in the future we would like to be able to include dust emission and absorption, which would be important for line transfer in higher-J lines [see e.g. Matsumoto et al., 2023]. This will be included in a future version.

MAGRITTE also cannot model radiation scattering. Efficiently modelling scattering would probably require a significant revision of the entire modelling strategy. Therefore, we do not plan to include it, at least not in the near future.

In the initial design of MAGRITTE, considerable efforts were made to be able to leverage distributed computing systems [see e.g. De Ceuster et al., 2020]. We noticed, however, that for our target applications, rather than to simulate a single large model, it is more important to be able to efficiently simulate large grids of medium-sized models [see e.g. Coenegrachts et al., 2023, Van de Sande et al., 2024]. These medium-sized models typically fit on a single compute node. Moreover, our discretisation reduction techniques help us fit also larger models on a single node [see e.g. De Ceuster et al., 2020, and Section 3.4.1]. Therefore, we currently focus on single-node performance, and have, at least temporarily, discontinued support for simulating on distributed systems. Still, most of the required infrastructure is still in the code base, so this could be restored in the future.

Conclusion 3.6

We have presented the first long-term stable release of MAGRITTE, an open-source software library for line radiative transfer. We have demonstrated its use with two applications creating synthetic spectral line observations, one for an analytic model of a protoplanetary disk and one for a numerical SPH model of a complex companion-perturbed stellar wind. Further, we have introduced the overall design strategy for MAGRITTE, including the class structure, and the code development and maintenance strategy, including automated testing. Finally, we presented three key improvements over previous versions. First, we have presented a simple and fast recursive re-meshing algorithm to efficiently create spatial discretisations of a model that can be tailored for radiative transfer simulations. We have demonstrated that our new algorithm can provide similar

[12] The boundary conditions can, of course, contain continuum radiation, and are, in fact, usually continuum sources (e.g. cosmic microwave background).

quality spatial discretisations as before, but it is 50 times faster. Second, we have presented an adaptive variation on the popular Ng-acceleration method to accelerate convergence in non-LTE line transfer simulations. We have shown that an optimal value for the only hyper-parameter in the classical algorithm (N_{steps}) can be determined on-the-fly to consistently obtain optimal convergence. Third, we have derived a semi-analytical approximation for the line optical depth, which can accurately handle even large velocity gradients at a small and fixed computational cost. This obviates the need for computationally more expensive sub-stepping methods. Fourth, we have implemented a more efficient evaluation of the total emissivity and opacity, which significantly improves computational performance. Finally, we have implement an interpolation scheme, based on the variation in line opacity, which improves the numerical accuracy in case of rapid source function variations. In summary, MAGRITTE is now a mature project with a significant application domain in astrophysical line radiation transport.

Acknowledgments

We thank Jolien Malfait for kindly providing the SPH model of the companion-perturbed stellar wind. We thank the anonymous reviewers for their helpful comments. TC is a PhD Fellow of the Research Council - Flanders (FWO), grant 1166722N. FDC is a Postdoctoral Research Fellow of the Research Foundation - Flanders (FWO), grant 1253223N. LD acknowledges support from the KU Leuven C1 MAESTRO grant C16/17/007, the FWO research grant G099720N and the KU Leuven C1 BRAVE grant C16/23/009.

Data Availability

MAGRITTE is an open-source software library publicly available at `github.com/Magritte-code/Magritte`. The benchmarks presented in this chapter are part of the source code. A fork of the source code, using MAGRITTE version 0.7.2, including JUPYTER notebooks and PYTHON scripts used for the simulations and creating the plots in this paper, can be found at `github.com/Magritte-code/Magritte_Paper_Astronomy_Computing`.

CHAPTER 4

Comoving Solver

This chapter is a slightly modified version of:

A numerically stable comoving frame solver for line radiative transfer

Ceulemans, T., De Ceuster, F., Decin, L., 2025

Journal of Quantitative Spectroscopy and Radiative Transfer Volume 343, 109470.
https://doi.org/10.1016/j.jqsrt.2025.109470

CRediT authorship contribution statement

T. Ceulemans: Writing – original draft, Visualization, Validation, Software. F. De Ceuster: Writing – review & editing, Supervision, Funding acquisition. L. Decin: Writing – review & editing, Supervision, Funding acquisition.

Abstract

Radiative transfer is essential in astronomy, both for interpreting observations and simulating various astrophysical phenomena. However, self-consistent line radiative transfer is computationally expensive, especially in 3D. To reduce the computational cost when utilizing a discrete angular discretization, we use a comoving frame interpretation of the radiative transfer equation. The main innovation of this paper lies in the novel stabilization method for the resulting numerical discretization. The stabilization method is able to reduce spurious oscillatory behavior in the computed intensities, at the expense of extra boundary conditions which need to be enforced. We also implement an adaptive angular discretization for the ray-tracing implementation, in order to efficiently and accurately calculate the radiation field. Finally, we apply this new numerical method to compute NLTE line radiative transfer on a hydrodynamics model, showcasing its potential improvement in computation efficiency.

Introduction 4.1

In the astronomy literature, various algorithms already exist for computing intensities in (line) radiative transfer, including discretizing the formal solution [as in Auer, 2003], using the Feautrier method [Feautrier, 1964], and moment method [see e.g. Auer and Mihalas, 1970]). Self-consistent non-local thermodynamical equilibrium (NLTE) line radiative transfer comprises of a set of non-linear equations, in which the radiation field and the properties of the medium depend on each other. To solve this problem, one can either resort to an iterative approach, such as the accelerated lambda iteration of Rybicki and Hummer [1991] or linearizing the equations [as done in Auer and Mihalas, 1969]. In this paper, we focus on the former approach. In this method, the slowest calculation step consists of computing the mean line intensity, which requires us to compute the radiation field at frequencies around the line center in the comoving frame at each position. To compute the intensity in a single direction, we require the computed intensity in the same direction from the upwind point. Therefore, theoretically, we can optimize our computations by reusing previously computed intensity information, instead of restarting the intensity computation from the model boundary. However, in case of a non-constant velocity field, we need to take into account the Doppler shift, which causes the required frequency ranges to be misaligned at the different positions.

To address this issue, one can rewrite the radiative transfer equation to the comoving frame following the approach outlined in Hauschildt [1992], or Castor [2004], Chapter 6. However, these initial approaches were restricted to using monotonic velocity fields. In Baron and Hauschildt [2004], Hauschildt and Baron [2004], the authors proposed a new solution method, which also allows for a non-monotonic velocity field. Nonetheless, the solution method may encounter numerical instabilities [see e.g. Sampoorna and Nagendra, 2016, Hauschildt and Baron, 2004] in case of larger Doppler shifts. In this paper, we will derive a similar algorithm, and provide a novel, and intuitive method to obtain a numerically stable numerical discretization. Afterwards, we will evaluate the accuracy and computational efficiency of the new solver by computing the radiation field on an AGB hydrodynamics simulation [taken from Malfait et al., 2024b] and comparing it to the results using a reference Feautrier solver [Feautrier, 1964]. For accurate results, we require an appropriate angular discretization for computing the intensity field. We have therefore implemented an adaptive angular discretization scheme, which generates a different set of directions for each model position.

This paper is structured as follows: Section 4.2 contains the derivation of the numerical discretization. Afterwards, we will stabilize the new solver in Section 4.3 using an intuitive method and provide a simple illustration why this stabilization is necessary. Afterwards, we describe our implementation of an adaptive angular discretization scheme in Section 4.4. Finally, we apply this new method to a 3D hydrodynamics model in Section 4.5, and analyze the resulting speedup and accuracy.

Comoving frame solver 4.2

Solving the radiative transfer equation gives us the difference in intensity between successive positions. Thus, to calculate the intensity in a specific direction using ray-tracing, the intensity at the previous position in that direction is required. For computational efficiency when computing the radiation field, it would be efficient to use the computed intensity at the previous position both for calculating the mean intensity at that position and to compute the intensity at the next position. However non-zero Doppler shifts change the frequency of the line center in the observer frame. To address this issue, we start from the comoving frame equation of radiative transfer for a non-relativistic steady flow [see Eq. 19.156 in Hubený and Mihalas, 2014]

$$\frac{\partial I(x,\nu)}{\partial x} - \frac{\nu}{c}\left(\frac{\partial \mathrm{v}}{\partial x}\right)\frac{\partial I(x,\nu)}{\partial \nu} = \eta(x,\nu) - \chi(x,\nu)I(x,\nu) \tag{4.1}$$

in which $I[\mathrm{W\ sr^{-1}\ m^{-2}\ Hz^{-1}}]$ denotes the specific monochromatic intensity, $x[\mathrm{m}]$ is the position, $\mathrm{v}[\mathrm{m\ s^{-1}}]$ is the velocity of the medium, $\nu[\mathrm{Hz}]$ is the frequency in the comoving frame, $\eta[\mathrm{W\ sr^{-1}m^{-3}\ Hz^{-1}}]$ is the total emissivity and $\chi[\mathrm{m^{-1}}]$ is the total opacity. We then adapt the equation by moving the $\frac{\partial I}{\partial \nu}$ term to the other side, and to improve the numerical stability of the resulting discretization, we add the term $\frac{\partial I}{\partial \nu}\frac{\partial \nu_{\mathrm{extra}}}{\partial x}$ to both sides. This allows for some freedom in the resulting numerical discretization, as the choice of $\nu_{\mathrm{extra}}(x)$ changes the comoving frame frequency along the path. Note that we have not yet defined $\nu_{\mathrm{extra}}(x)$. For stability reasons (as explained in Section 4.3), this term should be chosen to connect the (yet to be discretized) frequencies at the successive spatial points such that the total frequency change in the observer frame is minimized. The resulting equation becomes:

$$\frac{\partial I}{\partial x} + \frac{\partial I}{\partial \nu}\frac{\partial \nu_{\mathrm{extra}}}{\partial x} = \eta(x,\nu) - \chi(x,\nu)I + \frac{\partial I}{\partial \nu}\left[\frac{\nu}{c}\left(\frac{\partial \mathrm{v}}{\partial x}\right) + \frac{\partial \nu_{\mathrm{extra}}}{\partial x}\right]. \tag{4.2}$$

Assuming a non-relativistic Doppler shift, the term between square brackets is the total frequency change in the observer frame $\frac{\partial \nu_{\mathrm{obs}}}{\partial x}$, to which both the comoving frame and the extra frequency term ν_{extra} contribute.

By defining a path s such that[1]

$$\frac{dx}{ds} = 1 \tag{4.3}$$

$$\frac{d\nu}{ds} = \frac{\partial \nu_{\mathrm{extra}}}{\partial x}, \tag{4.4}$$

we can apply a change of variables to the left hand side of Eq. 4.2 in order to obtain

$$\frac{dI}{ds} = \eta(x,\nu) - \chi(x,\nu)I(x,\nu) + \frac{\partial I}{\partial \nu}\frac{\partial \nu_{\mathrm{obs}}}{\partial x}. \tag{4.5}$$

In this, s is the path length (see Figure 4.1). Given that s and x are equivalent (except for an additive constant, see Eq. 4.3), we will replace s with x in any future equation.

By defining the optical depth increment $\Delta\tau_i$ along this path with comoving frame frequency $\nu_i(x)$[2]

$$\Delta\tau_i = \int_{x_j}^{x_{j+1}} \chi(s,\nu_i(s))ds, \tag{4.6}$$

[1]The path defined by the set of equations $x(s) = s, \nu(s) = \nu_{\mathrm{CMF}} + \nu_{\mathrm{extra}}(x)$, in which ν_{CMF} is a constant, satisfies both equations 4.3 and 4.4.

[2]For consistency in notation, we already introduce the frequency index i, and the spatial index j.

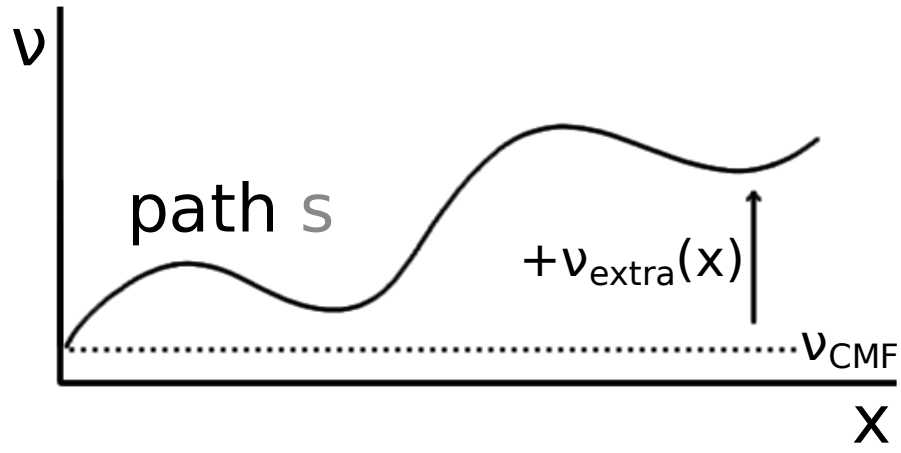


Figure 4.1: Illustration of how the frequency and position change over the path s, which is used for solving Eq. 4.5. By adding the term $\frac{\partial I}{\partial \nu}\frac{\partial \nu_{\rm extra}}{\partial x}$ to both sides of Eq. 4.2, the comoving frame frequency ν is different from a constant value $\nu_{\rm CMF}$ which would be used in Eq. 4.1.

we solve equation 4.5 in a similar manner to the formal solution [see e.g., Kunasz and Auer, 1988]

$$I_{j+1,i} = I_{j,i}e^{-\Delta\tau_i} + e^{-\Delta\tau_i}\int_0^{\Delta\tau_i} e^{\tau}\left(S + \frac{d\nu_{\rm obs,i}}{d\tau}\frac{\partial I}{\partial \nu}\right)d\tau, \tag{4.7}$$

in which we denote the discretized intensity $I(x_j, \nu_i(x_j))$ as $I_{j,i}$ and we define the source function S[W sr^{-1} m^{-2} Hz^{-1}] to be

$$S(x,\nu) = \frac{\eta(x,\nu)}{\chi(x,\nu)}. \tag{4.8}$$

Given the similarity in its definition with the observer frame optical depth, we will call τ_i the effective optical depth.

By discretizing this equation in a similar way to the first order accurate formal solution-type solvers, we end up with the following discretization

$$I_{j+1,i} = I_{j,i}e^{-\Delta\tau_i} + \text{Source} + \text{Shift} \tag{4.9}$$

in which the source term is given by

$$\text{Source} = S_{j,i}\left(\frac{1-e^{-\Delta\tau_i}-\Delta\tau_i e^{-\Delta\tau_i}}{\Delta\tau_i}\right) + S_{j+1,i}\left(\frac{\Delta\tau_i - 1 + e^{-\Delta\tau_i}}{\Delta\tau_i}\right), \quad (4.10)$$

in which $S_{j,i} = S(x_j, \nu_i(x_j))$ is the source function at a specific location and frequency. The shift term is given by, [3]

$$\begin{aligned}\text{Shift} =& \frac{\Delta\nu_{\text{obs},i}}{\Delta\tau_i}\frac{\partial I}{\partial\nu}|_{x_j,\nu_i(x_j)}\left(\frac{1-e^{-\Delta\tau_i}-\Delta\tau_i e^{-\Delta\tau_i}}{\Delta\tau_i}\right) \\ &+ \frac{\Delta\nu_{\text{obs},i}}{\Delta\tau_i}\frac{\partial I}{\partial\nu}|_{x_{j+1},\nu_i(x_{j+1})}\left(\frac{\Delta\tau_i - 1 + e^{-\Delta\tau_i}}{\Delta\tau_i}\right)\end{aligned} \quad (4.11)$$

in which $\Delta\nu_{\text{obs},i}$ denotes the frequency difference in the observer frame and the subscript denotes at which point to evaluate the source function and derivative. We denote with $\nu_{\text{obs},i}(x)$ the required frequencies in the observer frame used at each position to discretize this equation. Therefore, $\Delta\nu_{\text{obs},i} = \nu_{\text{obs},i}(x_{j+1}) - \nu_{\text{obs},i}(x_j)$.

To complete the discretization, we utilize a second order accurate forward discretization for the derivative of the intensity with respect to the frequency $\frac{\partial I}{\partial\nu}$

$$\frac{\partial I}{\partial\nu}|_{x=x_j,\nu=\nu_i(x_j)} \simeq \sum_{k=0}^{2} c_{ijk} I(x_j, \nu_{i+k}(x_j)) \quad (4.12)$$

for which the coefficients c_{ijk} are derived in Appendix 4.B and in which $\nu_i(x_j)$ are the discretized frequencies in the comoving frame at each position.

To efficiently solve the coupled set of Equations 4.9 for a single spatial increment, we will impose one assumption. We will demand that the stencil is chosen such that all frequency differences $\Delta\nu_{\text{obs},i}$ in the observer frame have the same sign, which is given by sign of the Doppler shift. After applying appropriate boundary conditions, the resulting set of equations will become a matrix equation with an upper or lower diagonal matrix on the left hand side, similar in shape to the left hand side of Eq. 4.15. We solve this set of equations using Gaussian elimination in $O(N_{\text{frequencies}})$ time.

[3] We have replaced the term $\frac{d\nu_{\text{obs}}}{d\tau}$ in equation 4.7 with the constant $\frac{\Delta\nu_{\text{obs}}}{\Delta\tau}$. This is an approximation, as we typically assume the frequency to change linearly with the distance x in order to compute the optical depth increment $\Delta\tau$.

How to practically compute the effective optical depth 4.2.1

In Eq. 4.6, we have defined the optical depth increments $\Delta\tau_i$ using a non-constant frequency path $\nu_i(x)$. We now explain how to evaluate this expression, as this formula looks similar to the observer frame optical depth, but is slightly different.

Starting from any computation method for the optical depth, when filling in any quantity at a given position x, we also need to substitute the corresponding frequency $\nu(x)$ at that position. For example, in case of the trapezoidal rule, the optical depth increment $\Delta\tau_i$ becomes

$$\Delta\tau_i = \Delta x \frac{\chi(x_j, \nu_i(x_j)) + \chi(x_{j+1}, \nu_i(x_{j+1}))}{2}, \tag{4.13}$$

in which Δx is the distance increment.

Instabilities 4.2.2

In two papers [Sampoorna and Nagendra, 2016, Hauschildt and Baron, 2004], numerical instabilities have been observed when applying the method of Baron and Hauschildt [2004] to models with large velocity gradients, but the origin of the instabilities was not discussed in these papers. We argue that the numerical instabilities are caused by an extrapolation in the frequency discretization, based on the stability criterion we derive in Section 4.2.3. The criterion states that numerical stability becomes worse as the Doppler shift increases relative to the spacing between adjacent frequencies in the frequency discretization [see also Section 6.6 in Castor, 2004].

To illustrate the instabilities which can occur, we have created a simple 1D single ray model with a single line transition $u \to l$ using a Gaussian line profile ϕ_{ul}

$$\phi_{ul}(x, \nu) = \frac{1}{\delta\nu_{ul}(x)\sqrt{\pi}} e^{-\frac{(\nu-\nu_{ul})^2}{\delta\nu_{ul}(x)^2}} \quad , \tag{4.14}$$

in which $\delta\nu_{ul}$ is the line width and ν_{ul} is the line center. The setup is described in Table 4.1. To properly showcase the possible numerical instabilities, we have defined this model with high optical depths and moderate Doppler shift. As we have defined the model such that the source function $S = 1.0$ is constant, and the initial intensity is 0, we expect from the formal solution that the intensity I in the entire frequency range is bounded from above by 1.0. However, when applying the comoving solver, if the path s is following the Doppler shift in

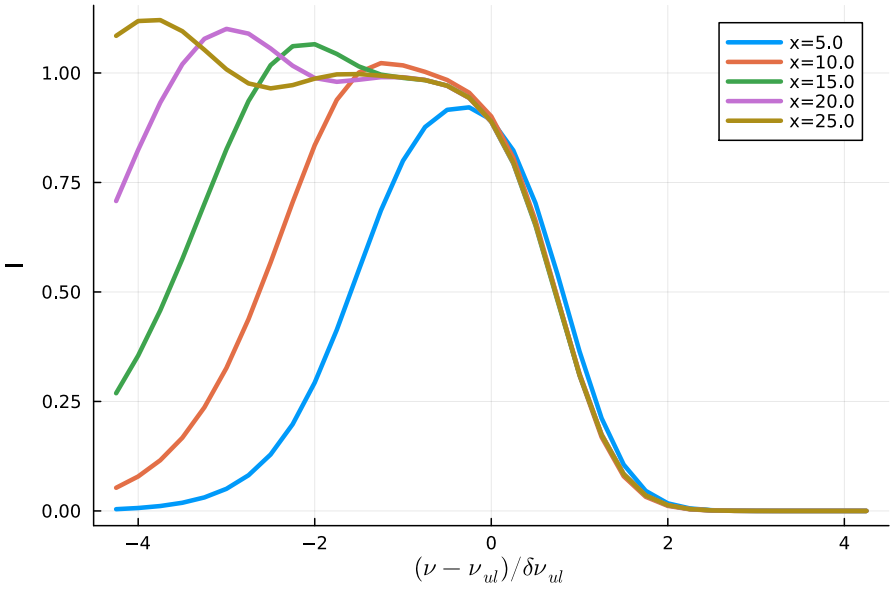


Figure 4.2: Illustration of the numerical behavior of the comoving solver, when following the Doppler shift in order to determine the discretization. All intensity profiles at the different positions are plotted in the comoving frame, centered around the line center ν_{ul}. Numerical oscillations are present in the computed intensity.

Table 4.1: Parameters for the 1D model with a constant velocity gradient. To evaluate the intensities, we use a uniform frequency discretization with spacing $\Delta\nu_s$. Between successive points at distance Δx, the Doppler shift causes a constant observer frame frequency difference $\Delta v_{\rm obs}$. The boundary intensity is set to 0. All units are re-scaled in dimensionless units such that the line width $\delta\nu_{ul} = 1.0$ and the line optical depth $\int_0^1 \chi/\phi_{ul}\ dx$ over a unit distance is 1.0.

χ/ϕ_{ul}	η/ϕ_{ul}	S	$\Delta\nu_s$	Δx	$\Delta v_{\rm obs}$
1.0	1.0	1.0	$\delta\nu_{ul}/4$	5.0	$1.1\delta\nu_{ul}$

frequency space, we observe oscillations (see Figure 4.2), pushing the computed intensities above the value of the source function, which is nonphysical.

Derivation of the stability of the comoving solver 4.2.3

Given the numerical instabilities described in the previous section, we derive the stability criterion for our discretization in this section. Therefore, we apply perturbation analysis to the discretized equations for the position interval $[x_0, x_1]$. The resulting set of equations can be written as a matrix equation, for which we will compute the eigenvalues. We have assumed the frequency differences $\Delta\nu_i$ to be positive (a similar derivation can be made for negative $\Delta\nu_i$). To complete the set of equations, boundary conditions have been posed on $I(x_1, \nu_{n-1}(x_1)), I(x_1, \nu_n(x_1))$.

$$\begin{bmatrix} 1+A_1 & B_1 & C_1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 1+A_{n-2} & B_{n-2} & C_{n-2} \\ 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta I_{1,1} \\ \vdots \\ \delta I_{1,n-2} \\ \delta I_{1,n-1} \\ \delta I_{1,n} \end{bmatrix} =$$

$$\begin{bmatrix} e^{-\Delta\tau_1}+A'_1 & B'_1 & C'_1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & e^{-\Delta\tau_{n-2}}+A'_{n-2} & B'_{n-2} & C'_{n-2} \\ 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta I_{0,1} \\ \vdots \\ \delta I_{0,n-2} \\ \delta I_{0,n-1} \\ \delta I_{0,n} \end{bmatrix}, \tag{4.15}$$

in which we define $\delta I_{j,i}$ as the perturbation on $I(x_j, \nu_i(x_j))$. In this equation,

$$A_i = -c_{i10}\frac{\Delta\nu_{\text{obs},i}}{\Delta\tau_i}\left(\frac{\Delta\tau_i - 1 + e^{-\Delta\tau_i}}{\Delta\tau_i}\right)$$

$$B_i = -c_{i11}\frac{\Delta\nu_{\text{obs},i}}{\Delta\tau_i}\left(\frac{\Delta\tau_i - 1 + e^{-\Delta\tau_i}}{\Delta\tau_i}\right)$$

$$C_i = -c_{i12}\frac{\Delta\nu_{\text{obs},i}}{\Delta\tau_i}\left(\frac{\Delta\tau_i - 1 + e^{-\Delta\tau_i}}{\Delta\tau_i}\right)$$

On the right-hand side, we have similarly defined

$$A'_i = c_{i00}\frac{\Delta\nu_{\mathrm{obs},i}}{\Delta\tau_i}\left(\frac{1-e^{-\Delta\tau_i}-\Delta\tau_i e^{-\Delta\tau_i}}{\Delta\tau_i}\right)$$

$$B'_i = c_{i01}\frac{\Delta\nu_{\mathrm{obs},i}}{\Delta\tau_i}\left(\frac{1-e^{-\Delta\tau_i}-\Delta\tau_i e^{-\Delta\tau_i}}{\Delta\tau_i}\right)$$

$$C'_i = c_{i02}\frac{\Delta\nu_{\mathrm{obs},i}}{\Delta\tau_i}\left(\frac{1-e^{-\Delta\tau_i}-\Delta\tau_i e^{-\Delta\tau_i}}{\Delta\tau_i}\right)$$

The eigenvalues of this system of equations determine the stability of this discretization. To find them, one can exploit the properties of upper diagonal matrices: the inverse of an upper diagonal matrix also being an upper diagonal matrix and the multiplication of two upper diagonal matrices still being an upper diagonal matrix. Thus, by inverting the matrix on the left-hand side (multiplying it together with the one on the right-hand side), another upper diagonal matrix is obtained. For an upper diagonal matrix, one can simply find the eigenvalues λ_i by looking at the diagonal. The resulting diagonal elements are given by:

$$\lambda_i = \frac{e^{-\Delta\tau_i}+A'_i}{1+A_i}. \tag{4.16}$$

As equation (4.16) is too dense, we will first define the constants $d_i = c_{i10}\Delta\nu_{\mathrm{obs},i}$, $d'_i = c_{i00}\Delta\nu_{\mathrm{obs},i}$. These quantities are related to the ratio of frequency difference $\Delta\nu_{\mathrm{obs},i}$ versus the width of the frequency discretization. Then by rewriting, one finds the stability condition to be

$$\lambda_i = \frac{\Delta\tau_i^2 e^{-\Delta\tau_i} + d'_i\left(1-e^{-\Delta\tau_i}-\Delta\tau_i e^{-\Delta\tau_i}\right)}{\Delta\tau_i^2 - d_i\left(\Delta\tau_i - 1 + e^{-\Delta\tau_i}\right)}. \tag{4.17}$$

The result is that this discretization is almost unconditionally stable[4], meaning that $|\lambda_i| \leq 1$, given that coefficients $c_{i00}, c_{i10} < 0$ if all frequencies are ordered (see Appendix 4.B). Assuming $c_{i00} = c_{i10}$, and $\Delta\tau_i > 0$, both terms of the numerator are smaller (in absolute value) than the corresponding (strictly positive) terms in the denominator, and thus $|\lambda_i| < 1$. However, oscillatory behavior ($-1 \lessapprox \lambda_i < 0$) can exist if the parameters d_i, d'_i are large. This corresponds to extrapolating the frequency derivative terms.

[4]Due to the possible difference between c_{i10} and c_{i00}, no guarantee exists for stability. This corresponds with the contraction/expansion of the frequency discretization. The existence of maser regimes, for which $\Delta\tau < 0$, is also ignored, as the frequency derivative will not be sampled well in that regime.

Frequency matching 4.3

In the preceding section, we have derived the comoving solver and its properties regarding numerical stability. Given the potential for numerical instabilities to produce nonphysical results, we require a method to mitigate these numerical effects without significantly affecting computational efficiency. In this section, we describe a novel, yet intuitive method for stabilizing numerical discretizations of transport equations in order to reduce numerical oscillations in the computed result.

A novel way to sidestep numerical instabilities 4.3.1

Upon examining the stability criterion (Eq. 4.17), two apparent methods emerge to improve numerical stability: either insert extra points in-between for interpolating the velocity difference, therefore reducing the Doppler shifts between successive spatial positions, or coarsen the frequency discretization. However, neither solution is appropriate for our purposes: in the former case, the computation time increases, contradicting the objective of this paper, while in the latter case, the accuracy decreases when computing the mean line intensity. In order to reach our goal of numerical stability, we must explore alternative approaches.

The main stability issue arises from the extrapolation of the frequency differences, due to the change in frequency being dictated by the Doppler shift, which can be greater than the spacing of the frequency discretizations. Therefore, the stability issues could be resolved by minimizing the frequency changes. This can be achieved by altering how to connect the frequency discretizations of successive spatial positions. The most straightforward approach involves mapping all frequency indices one-to-one from low to high frequency. However, mapping all frequency indices is not required. Instead, we can do the mapping in such a way which ensures the frequency differences to be minimized. This solves our numerical stability issues at the cost of extra boundary conditions (see Figure 4.3 and Sect. 4.3.2). To illustrate the impact of this method, we apply it to the simple 1D ray model described in Table 4.1. By mapping the frequency discretization such that the observer frame frequency differences $\Delta\nu_{\mathrm{obs},i}$ are minimal (see Eq. 4.18), we obtain the results in Figure 4.4. No numerical oscillations occur, and the computed intensities remain below the value of the constant source function S. We prove that this method makes the numerical discretization always stable in Appendix 4.C (except for maser-like conditions).

For practically implementing this method, we recommend using the observer frame to evaluate the frequency difference, as the objective is to minimize

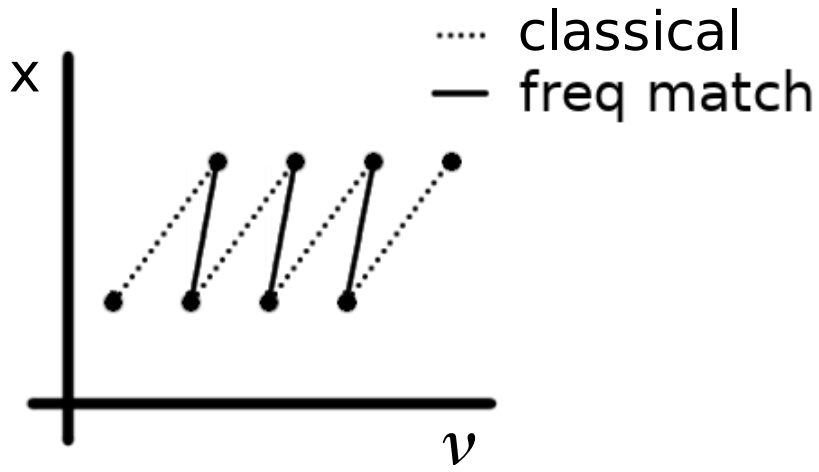


Figure 4.3: Illustration of the different ways to match frequencies in the comoving solver. Intuitively, one would connect the frequencies by following the doppler shift (classical). By minimizing the frequency difference in the observer frame instead (freq match), numerical instabilities can be suppressed, at the cost of extra boundary conditions, as less frequency points are connected.

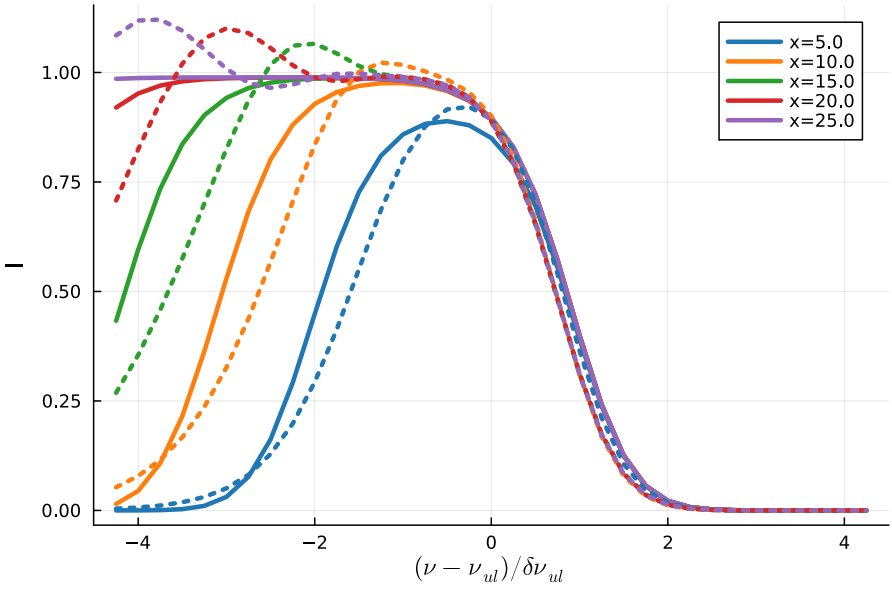


Figure 4.4: Illustration of the numerical behavior of the comoving solver, when matching the frequencies in order to determine the discretization. No numerical oscillations can be seen in this plot, in contrast to when letting the frequencies follow the Doppler shift. For a better comparison with Figure 4.2, we have added the data of that figure as dotted lines in this figure.

$|\Delta\nu_{\mathrm{obs},i}|$ when constructing the stencil. Assuming frequency discretizations $\nu_{\mathrm{obs},i}(x_j), \nu_{\mathrm{obs},i}(x_{j+1})$, determine the stencil for all $\nu_{\mathrm{obs},i}(x_{j+1})$ such that

$$|\Delta\nu_{\mathrm{obs},i}| = \left|\nu_{\mathrm{obs},i_{j+1}}(x_{j+1}) - \nu_{\mathrm{obs},i_j}(x_j)\right| \tag{4.18}$$

is minimized, in which i_j, i_{j+1} are the frequency indices to connnect. For ease of solving the comoving formulation of the radiative transfer equation (Eq. 4.5), make sure that the sign of $\nu_{\mathrm{obs},i_{j+1}}(x_{j+1}) - \nu_{\mathrm{obs},i_j}(x_j)$ is positive for all $\nu_{\mathrm{obs},i}(x_{j+1})$ in case of a positive Doppler shift ($\Delta \mathrm{v} > 0$) and vice versa. In this way, we either obtain an upper or a lower diagonal matrix for the system of equations to solve when discretizing the equation using Eq. 4.9. In Section 4.3.2, we will explain how to implement the boundary conditions.

As a side note, when taking the limit of an infinitely dense frequency discretization, resulting in $|\Delta\nu_{\mathrm{obs},i}| \to 0$, the discretized equations will converge to the observer frame formulation, given that the Shift term in Eq. 4.9 is linearly dependent on $\Delta\nu_{\mathrm{obs},i}$.

Dealing with boundary conditions 4.3.2

Matching the frequencies introduces extra boundary conditions, due to not mapping the entire frequency discretization. Given the complexity of implementing these boundary conditions, we provide a brief overview of our approach in this section. A more comprehensive explanation can be found in Appendix 4.A.

In general hydrodynamics models, the velocity profile is usually non-monotonic. In this paper, we assume the intensities to only change due to the influence of molecular lines, meaning that the intensities at frequencies sufficiently far enough from the line centers, remain constant. The comoving method discards information at the frequencies it is moving away from. Consequently, during any later computation, it might re-encounter frequencies for which it has discarded the previously computed intensities, as illustrated in Fig. 4.5. To address this issue, we have implemented two different approaches. A first approach involves temporarily storing the computed intensities and reusing them when re-encountering the corresponding frequencies, which requires extensive bookkeeping. Alternatively, for a simpler, more approximate implementation, one can disregard the previously computed intensities entirely, always filling in the boundary conditions using black-body intensities. However, this method is only strictly valid if the non-monotonicity of the velocity field is minor compared to the total width of the frequency discretization. While we expect the former approach to be more accurate, it is also more time-consuming than the latter

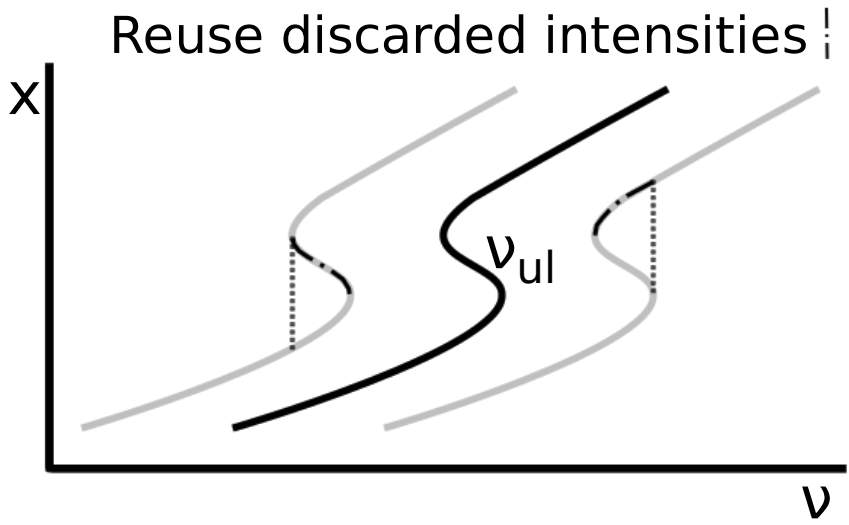


Figure 4.5: Illustration of why extra boundary conditions are required for the comoving solver. The line center ν_{ul}, together with the bounds of the frequency discretization (in grey) move according to the Doppler shift. As we shift the discretization, the computed intensities at frequencies outside of the current frequency discretization will be discarded. Inside the dash-dotted regions, we require previously computed intensity information as we re-encounter these frequencies.

approach.

A benchmark with non-monotonic velocity field 4.3.3

To show the impact of the different treatments of the frequency boundary condition (see Section 4.3.2) on the accuracy, we devise a minimal benchmark with a non-monotonic velocity field to compare the computed mean line intensities J_{ul} (see Eq. 4.20) of the different solvers. This benchmark involves a 1D model of a single ray, where we compute intensities for the CO v= 0 J=1 − 0 line[5]. We will compare the mean line intensity computed using the comoving solver against the value computed using an established solver, the Feautrier

[5]The line data is obtained from the LAMDA database, available at `https://home.strw.leidenuniv.nl/~moldata/`.

solver [Feautrier, 1964], as implemented in MAGRITTE [see e.g. Ceulemans et al., 2024, De Ceuster et al., 2022]. We will test both implementations of the frequency boundary conditions. To properly assess the capability of the solver in handling non-monotonic velocity fields, we opt for a sinusoidal velocity prescription (see Fig. 4.6), with its amplitude several times the width of the frequency discretization around each line. Since this is a benchmark for the non-monotonic behavior of the velocity field, we keep all other parameters constant. The velocity profile is given by a sinusoidal function with period of 60 model points and amplitude $2.5 \cdot 10^3$ m/s. The model assumes LTE, a Gaussian line profile (see Eq. 4.14) and corresponding line width $\delta\nu_{ul}$ computed according to

$$\delta\nu_{ul}(x) = \frac{\nu_{ul}}{c}\sqrt{\frac{2k_bT}{m_{\text{spec}}} + v_{\text{turb}}^2}. \tag{4.19}$$

In this equation, $k_b = 1.38 \cdot 10^{-23}$J K^{-1} denotes Boltzmann's constant, T[K] represents the local gas temperature, m_{spec}[kg] is the gas species mass and v_{turb}[m s^{-1}] is the turbulent velocity.

We use a Gauss-Hermite quadrature, using 25 frequencies for the frequency discretization. The spatial domain is linearly discretized with 150 points, using a spacing of $1.0 \cdot 10^{11}$ m between them. Finally, the gas temperature is 500 K and the CO number density is given by $1.0 \cdot 10^8$ m^{-3}. This benchmark represents a simplified version of a slice of the spiral outflow originating from the binary interaction between an AGB star and a companion, which represents our application in Section 4.5.

In this benchmark, a correct treatment of the frequency boundary is required (compare Fig. 4.6 with Fig. 4.5). We will compare the results of the different solvers using the mean line intensity

$$J_{ul}(\boldsymbol{x}) = \frac{1}{4\pi}\oint_\Omega \int_0^{+\infty} I(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu)\phi_{ul}(\nu)\mathrm{d}\Omega\mathrm{d}\nu. \tag{4.20}$$

As expected, the approximate version of the comoving solver performs poorly (see Fig. 4.7). The computed mean line intensity is significantly lower than the reference intensity because this implementation does not store the previously computed intensities. The version with correct boundary conditions obtains similar results to the reference Feautrier solver, with a mean absolute relative difference in mean line intensity J_{ul} of 2% for the CO v=0 J=1-0 line. However, the comoving solver is more dissipative than the Feautrier solver. This benchmark served a proof of concept, to show that we can apply the comoving solver to non-monotonic velocity field. We refer the reader to Section 4.5 for an

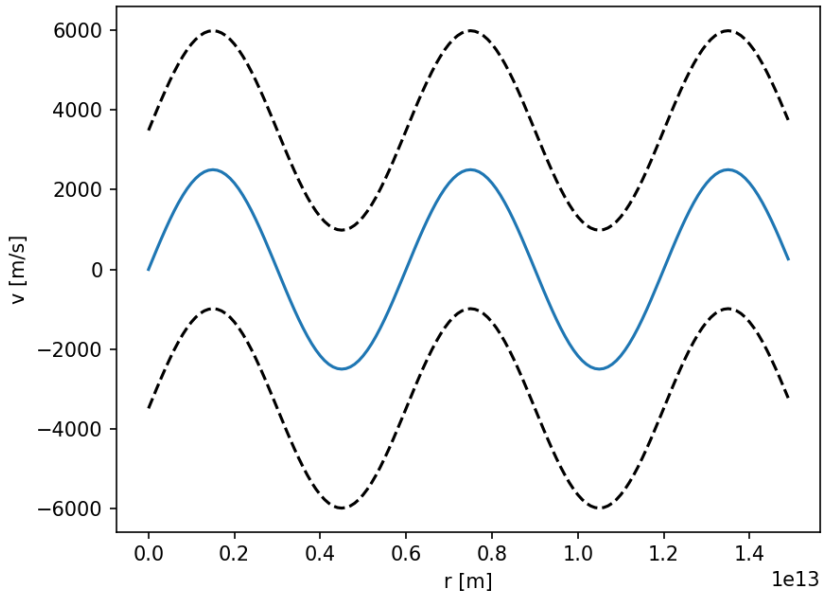


Figure 4.6: The non-monotonic velocity profile of the comoving solver boundary condition benchmark. Both the velocity at each position and corresponding bounds of the frequency discretization (converted to velocity using the Doppler shift) are present. For most of the domain, the boundary conditions we evaluate require previously computed intensities (see Fig. 4.5).

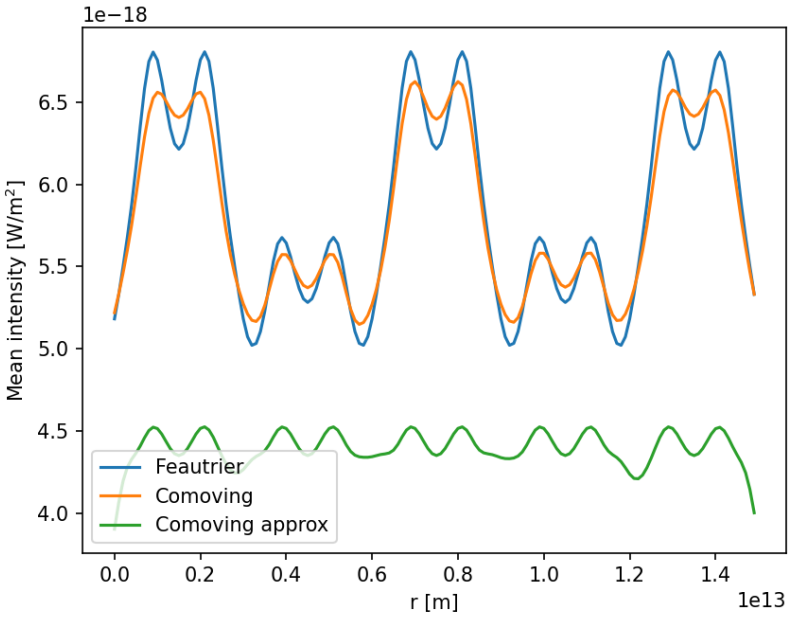


Figure 4.7: The computed mean line intensity J_{ul} for the CO J=1-0 v=0 line, for the 1D benchmark model with a sinusoidal velocity field. The computed results from the comoving solver with correct frequency boundary conditions agree with the reference values obtained from the Feautrier solver. The comoving solver with approximate boundary conditions fails the benchmark, given that the velocity field of the model is significantly non-monotonic.

application of this solver to a 3D hydrodynamics model. Please note that the result of this benchmark does not imply that one must always use fully correct frequency boundary conditions. In situations where the non-monotonicity of the velocity field is insignificant, using approximate frequency boundary conditions, which require less computation time, may suffice.

Adaptive angular discretization 4.4

When computing angle-averaged mean intensities in a model using ray-tracing, the chosen angular discretization impacts the results. The simplest option is to

use a uniform angular discretization at each position in the model. However, when the incoming intensity varies significantly depending on the angle, this approach can result in numerical artifacts, as smaller angular features cannot be resolved with a coarse discretization. For example, when applying radiative transfer on the hydrodynamics model in Section 4.5, the computed mean intensity field shows rays originating from the dense model center, due to the higher intensity originating from this small, dense region (see Fig. 4.8).

To solve this numerical issue, we will not naively increase the density of the angular discretization, as this will significantly increase the overall computational cost. Instead, we will determine for each position a good set of directions to use. For every point, we estimate the density of the direction vectors towards the other points in the model on a fine discretization of the entire solid angle. The regions of the solid angle with the highest density require a fine angular resolution, while the other regions can be approximated using coarse angular resolution.

Adaptive angular discretization algorithm 4.4.1

In this section, we will explain the algorithm [inspired by De Ceuster et al., 2020] that we use to generate the adaptive angular discretization, and how to apply this to the comoving solver. The algorithm allows us to achive a proper angular resolution for small-scale features at any given point by adaptively assigning an angular resolution depending on the density of model points around a given direction. This allows us to efficiently resolve the star or the companion object at positions far away from these objects, by using a relatively low amount of directions for evaluating the radiative transfer equation.

We initialize the algorithm by specifying the discretization level of the HEALPIX sphere. In this paper, we have chosen the 5th discretization level, which contains to $12 \cdot 2^{2 \cdot 5} = 12288$ pixels. Since the computation time for the mean intensity scales linearly with the number of angles, we want to minimize the number of directions. Every individual point requires a different angular discretization, as the regions of interest lie in a different position relatively to the position of the point. For computation time scaling purposes[6], we will first select a randomly chosen set of points used to infer which model regions should be sampled more densely (totaling 10000 points in this paper). To shorten notation, we define the "pixel count" to be the number of sample points for which their relative position to the current point has a direction contained in the pixel of the HEALPIX

[6]Figuring out at each position in which pixel all other points lie, costs $O(N^2_{\text{points}})$ computation time. A randomly chosen subset will be used as a proxy for the location of all points.

[Górski et al., 2005] discretization of the sphere. We give the algorithm for generating the adaptive angular discretization for a single position in Listing 4.1.

Listing 4.1: Adaptive direction discretization algorithm. We denote the pixel count with $P(a, n)$, using angle index a, discretization level $n \in 0, \ldots, N$, with N being the finest level. The HEALPIX angular discretization recursively subdivides, such that $P(a, N-1) = \sum_{i=0}^{3} P(4a+i, N)$. Finally, M is the maximum amount of pixels to subdivide per discretization level (2 in this paper).

1. Create empty list $\mathcal{L}$ of discretized angles
2. Compute pixel count $P(a, N)$
3. Compute $P(a, N-1) = \sum_{i=0}^{3} P(4a+i, N)$
4. Subdivide M pixels with highest $P(a, N-1)$, adding 4M pixels to $\mathcal{L}$
5. Compute $P(a, N-2) = \sum_{i=0}^{3} P(4a+i, N-1)$
6. Identify pixels at level $N-2$ which contain pixels in $\mathcal{L}$, denote count with $C \leq M$
7. Add the corresponding subpixels which do not yet overlap with any pixels in $\mathcal{L}$, to $\mathcal{L}$
8. Find other M-C pixels with highest $P(a, N-2)$, adding 4(M-C) pixels to $\mathcal{L}$
9. Repeat steps 5 to 8 for each coarser level until either the entire solid angle is subdivided or the coarsest level is reached. Add remaining pixels at level 0 to $\mathcal{L}$, such that $\mathcal{L}$ covers the entire solid angle.
10. Return $\mathcal{L}$

Note that as MAGRITTE requires point-wise symmetric directions (needed for the Feautrier solver), we apply this algorithm to only half of the solid angle. To achieve this, we have mapped the entire sphere onto half a sphere by considering the maximal count of the point-wise symmetric pixels over both halves.

The algorithm above describes the angular discretization used in the Feautrier solver. For the comoving solver, we need to slightly modify the ray-tracing algorithms. As a first implementation, we will only reuse intensities if the direction of the ray is a part of the angular discretization of another point on the ray. Do note that this results in a reduced efficiency of the comoving solver when compared to using a uniform angular discretization, as less information will be reused, and therefore radiative transfer will need to be computed on more rays. As future work, an interpolation scheme might be implemented in order to evaluate the mean intensity using intensities from directions which do not exactly match the angular discretization at a specific point. However, this is out of scope for this paper.

Table 4.2: Phantom model “v10e00" model parameters

AGB stellar mass M_p	$1.5 M_\odot$
AGB stellar temperature T_p	3000 K
Companion mass M_s	$1 M_\odot$
Initial wind speed v_{ini}	10 km/s
Semi-major axis a	6 AU

Application 4.5

In the previous sections, we considered 1D radiative transfer. For quantifying the speed and accuracy of the new solver, we have implemented the comoving solver in Magritte [Ceulemans et al., 2024, De Ceuster et al., 2022, De Ceuster et al., 2020, 2019] and we will compare it against the well-established Feautrier solver [Feautrier, 1964] by computing NLTE line radiative transfer in a 3D Phantom [Price et al., 2018] model. As Magritte is a ray-based 3D radiative transfer code, it computes the radiation field by solving many 1D radiative transfer problems[7]. The comoving solver can therefore replace the Feautrier solver in the intensity calculation step.

The model “v10e00” taken from Malfait et al. [2024b] is a smoothed particle hydrodynamics simulation of an AGB star with a companion star. The model parameters can be found in Table 4.2. In this model, we assume a CO/H_2 abundance ratio of 10^{-4} and will be using the CO v=0 J=0 to J=40 v=0 levels available from the LAMDA database [Schöier et al., 2005]. Velocities, temperatures and H_2 densities are taken from the hydrodynamics model. We have implemented two versions of the comoving solver, each using a different treatment of the boundary conditions, as explained in Section 4.3.2. The version with approximate boundary condition will be denoted by ‘comoving approx’. Since the timings and accuracy results between these two approaches vary, we will test both on the model to compare them versus the Feautrier solver. All simulations in this section were run on a server with dual AMD EPYC 7643 socket, utilizing all 192 threads. We compare the results of the different solvers using the mean line intensity J_{ul} (see Eq. 4.20).

Uniform angular discretization 4.5.1

In this section, we compare the solvers using a uniform angular discretization, based on the second discretization level of the Healpix [Górski et al., 2005],

[7] For the radiation field, we require the intensity at every position, discretized direction and discretized frequency in the comoving frame.

Table 4.3: Time required to compute NLTE level populations (5 iterations) on the PHANTOM model, containing 885501 points and using a uniform angular discretization. To illustrate the maximum possible obtainable speedup (assuming all algorithms taking the same amount of time per computed ray), we also add the number of rays used to compute the intensity field per solver. This table compares the Feautrier solver, the comoving solver with correct boundary conditions, and the comoving solver with approximate boundary conditions.

Method	Time[s]	Rays [.]
Feautrier	$9.7 \cdot 10^3$	$2.1 \cdot 10^7$
Comoving	$4.8 \cdot 10^3$	$9.3 \cdot 10^5$
Comoving approx	$1.9 \cdot 10^3$	$9.3 \cdot 10^5$

containing 48 discrete ray directions.

On Fig. 4.8, we notice that the relative differences between the intensities obtained using the comoving and the Feautrier solvers seem to be quite high on some straight rays which originate from the dense model center. These numerical artifacts have already been explained in Section 4.4. Due to the presence of the aforementioned numerical artifacts, the obtained results might not be fully representative, but for completeness, we still include the timings and accuracy results in this section.

We find that the comoving solver with accurate boundary conditions has relative differences in NLTE intensities less than 1% for 89% of model points, when compared to the Feautrier solver (see Fig. 4.9). The comoving solver with approximate boundary conditions has relative differences in mean intensities below 1% for 75% of the model points. The timings show (see Table 4.3) that both comoving solvers provide significant speedups when compared to the Feautrier solver. The reuse of computed intensities in the comoving solvers results in significantly less rays being used for computing the intensity field, when compared to the Feautrier solver.

Adaptive angular discretization 4.5.2

In order to be able to compare the comoving solver of this paper and the reference Feautrier method without the presence of numerical artifacts, we implement an adaptive angular discretization, as explained in Section 4.4. As seen in Fig. 4.10, both versions of the comoving solver perform well in the outside regions of the model. But, the comoving solver version with more correct treatment of the boundary, is more accurate in the center of the model, where the velocity field is non-monotonic (see Fig. 4.11). We remark that we have not

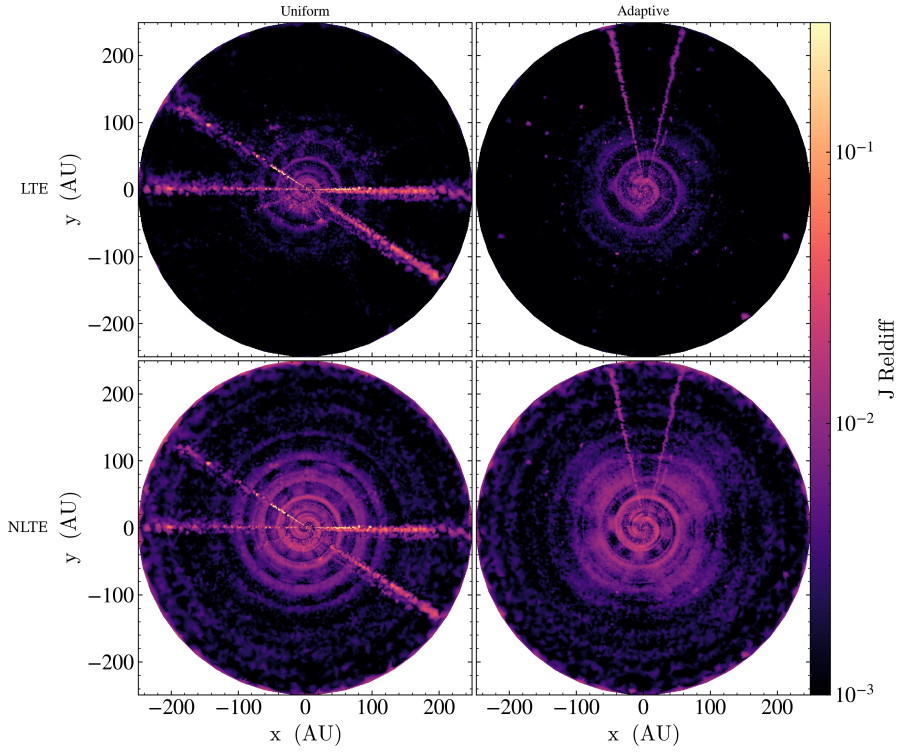


Figure 4.8: Slice plot of the relative differences between the computed mean line intensity J_{ul} for the CO v=0 J=2-1 line using the Feautrier solver and the comoving solver. Left: using the uniform angular discretization. Right: using the adaptive angular discretization. We plot both the intensity in case of LTE and the self-consistent intensity after convergence (NLTE). On the left, we see high relative differences on straight rays emerging from the model center, which is an effect of the limited angular discretization.

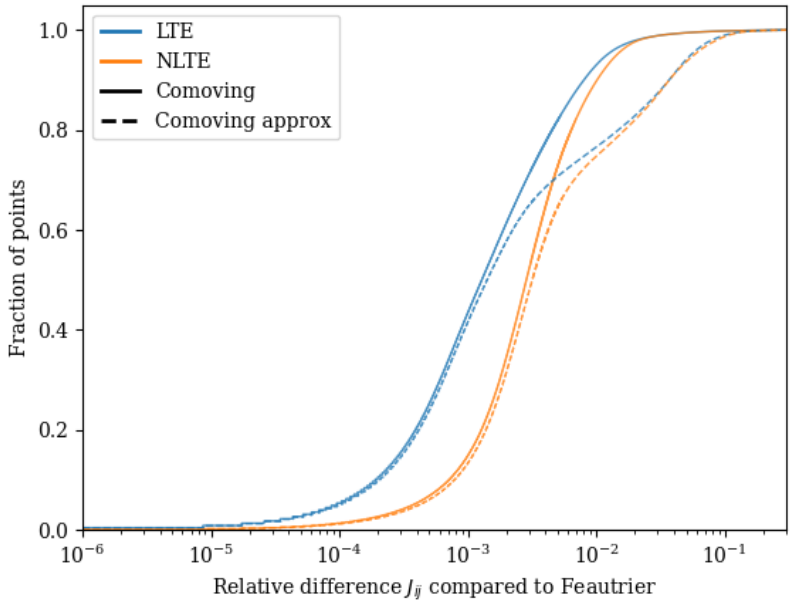


Figure 4.9: Cumulative difference between the computed mean line intensity J_{ul} for the CO J=2-1 v=0 line using the Feautrier solver and the comoving solvers, when using a uniform angular discretization. Both the differences in case of a single iteration after setting the level populations n_i to LTE (in blue) and the difference after letting the level populations converge (in orange). The figure shows the results for the correct boundary conditions described in Section 4.3.2 as solid lines and the results when using the approximate boundary conditions as dashed lines.

been able to fully eliminate the numerical artifacts seen in Section 4.5.1. We think this to be caused due to the rays used in the comoving solver not passing exactly through each model point, which slightly changes the viewing angle towards the dense regions. The numerical artifacts are not occurring in the approximate version of the comoving solver because the intensity information from the dense center gets lost due to Doppler shifts. By further increasing the angular resolution (as done in Appendix 4.D), we can suppress the numerical artifacts. In Fig. 4.12, we see that the relative differences of the mean line intensity J_{ul} at NLTE, compared to the value computed using the Feautrier solver, lie below 1% for 94% of the points in the model in case of the comoving solver, and below 1% for 78% of the points in the model in case of the comoving solver with approximate boundary conditions. We find that the adaptive angular discretization has a significant impact on the computational performance, due to decreased reuse of the computed intensities $I(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu)$ on rays, as different positions now require a different set of directions. As shown in Table 4.4, we need relatively more rays to compute the entire intensity field, compared to the uniform angular discretization case (about 10% of the rays used by the Feautrier solver compared to 5% when using a uniform angular discretization). This results in an insignificant speedup for the comoving solver with correct boundary conditions, and a significantly lower speedup for the comoving solver with approximate boundary conditions, compared to Section 4.5.1.

In this model, which has a spherically symmetric outflow perturbed by a companion, it appears that the more correct boundary conditions are unnecessary for the outer regions of the model. In these areas, the velocity field can be approximated by a monotonic profile for most directions of the entire solid angle. The only region where the more accurate boundary conditions are required, is the very center of the model. In all other regions of the model, disregarding the non-monotonic behavior of the velocity field may result in insignificant errors, potentially allowing us to save computation time (compare both versions of the comoving solver in Table 4.4).

Limitations 4.6

All tests above were done using Doppler shifts between successive positions significantly smaller than the width of the frequency discretization (of the individual lines). The stabilization method breaks down when the Doppler shift becomes similar size of large than the width of the frequency discretization, as then all computed data will be evaluated using boundary conditions. So for larger Doppler shifts, this method may fail; however, this criterion is always evaluated relative to the width of the frequency discretization. For example,

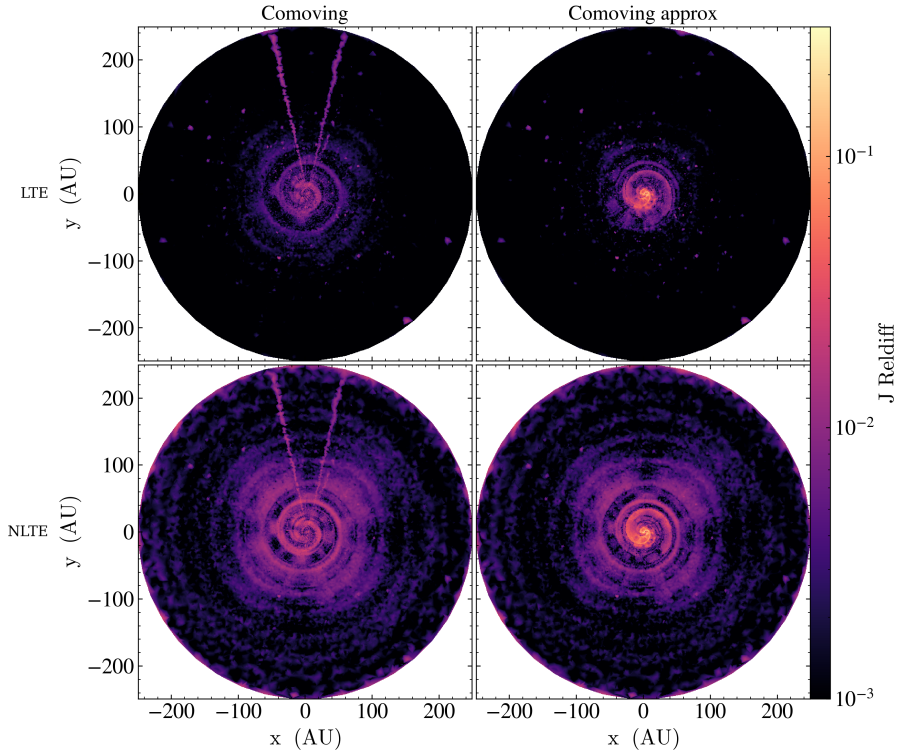


Figure 4.10: Slice plots of the relative differences between the computed mean line intensity J_{ul} for the CO J=2-1 v=0 line using the Feautrier solver and both comoving solvers. The relative differences in mean line intensity have been calculated both after a single iteration starting from LTE (LTE) and after convergence (NLTE).

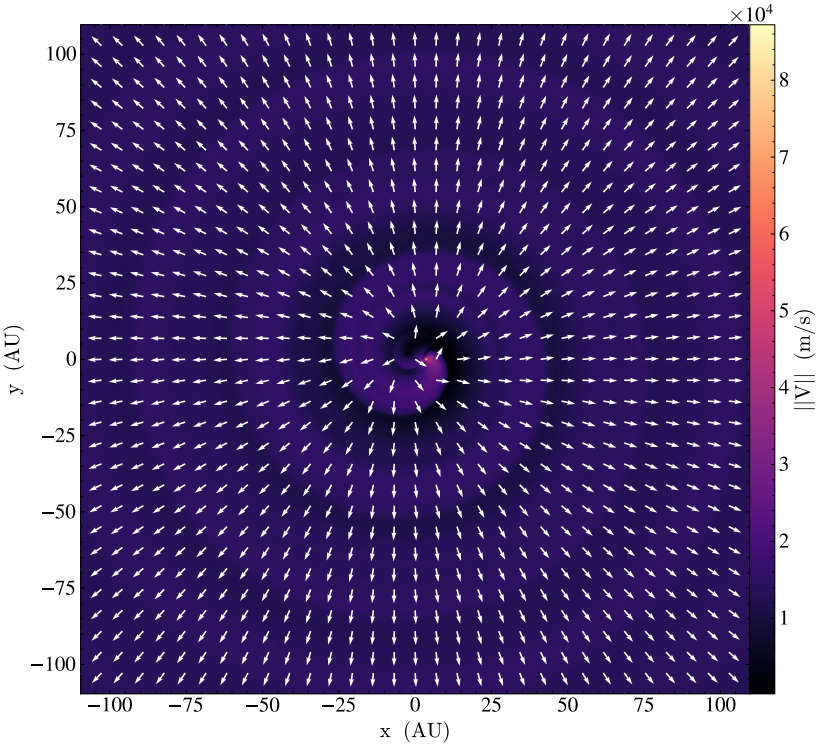


Figure 4.11: Zoomed in slice plot of the velocity field of the PHANTOM hydrodynamics model. The arrows denote the direction of the velocity field.

Table 4.4: Time required to compute NLTE level populations (5 iterations) on the PHANTOM model, containing 885501 points, using an adaptive angular discretization. To illustrate the maximum possible obtainable speedup, we also add the number of rays used to compute the intensity field per solver. Do note that this theoretical speedup assumes that all computation methods require the same computation time per ray. This table compares the Feautrier solver, the comoving solver with correct boundary conditions, and the comoving solver with approximate boundary conditions.

Method	Time[s]	Rays[.]
Feautrier	$1.9 \cdot 10^4$	$3.2 \cdot 10^7$
Comoving	$1.8 \cdot 10^4$	$3.1 \cdot 10^6$
Comoving approx	$7.2 \cdot 10^3$	$3.1 \cdot 10^6$

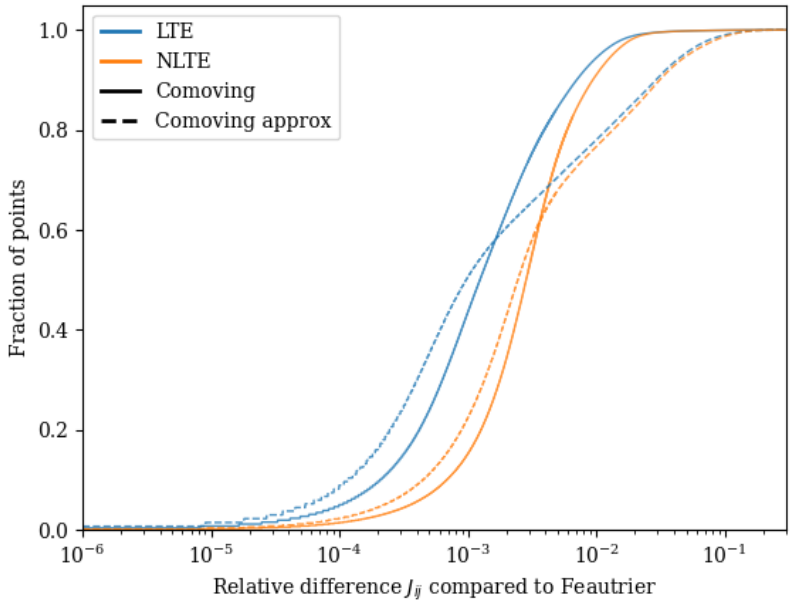


Figure 4.12: Cumulative difference between the computed mean line intensity J_{ul} for the CO J=2-1 v=0 line using the Feautrier solver and the comoving solvers with an adaptive angular discretization. Both the differences in case of a single iteration after setting the level populations n_i to LTE (in blue) and the difference after letting the level populations converge (in orange). The figure shows the results for the correct boundary conditions described in Appendix 4.A as solid lines and the results when using the approximate boundary conditions as dashed lines.

Wolf-Rayet stars may exhibit very strong winds, with terminal velocities on the order of 1000 km/s [see Fig. 9 of Poniatowski et al., 2021], but the corresponding radiative transfer models are typically not evaluated line-by-line because of the resulting Doppler shifts. Thus, as long as the encompassing frequency distribution is significantly wider than the maximum Doppler shift per position increment, this method is still applicable.

Conclusion 4.7

In this paper, we have derived a computationally efficient, numerically stable method for calculating the radiation field in line radiative transfer by enabling reuse of previously computed intensities. Compared to the comoving frame solution methods found in the literature [see e.g. Baron and Hauschildt, 2004, Hauschildt and Baron, 2004, Castor, 2004], our new stabilization method ensures numerical stability, no matter whether the size of the Doppler shifts is smaller or larger than the utilized frequency discretization. We derive the computation method in Section 4.2, where we have discretized a version of the 1D radiative transfer equation which allows for a change in frequency, similar to [Baron and Hauschildt, 2004, Hauschildt and Baron, 2004]. We prove that the numerical stability of the method mostly depends on the ratio of the Doppler shift versus the spacing of the frequency discretization (see Section 4.2.3). To improve the numerical stability of the discretization, we introduce a technique we call frequency matching, detailed in Section 4.3. In this technique, we ensure that the frequency paths $\nu_i(x)$ used for discretizing the equations cover the shortest possible distance in frequency space. Through a simple 1D plane-parallel example (see Sections 4.2.2, 4.3.1), we show that this modification can significantly reduce the oscillatory behavior of the resulting solver. This frequency matching procedure introduces additional boundary conditions in our numerical discretization, which we treat in Section 4.3.2. The importance of using an appropriate implementation for the boundary condition is demonstrated in Section 4.3.3. In the first implementation, the previously computed intensities are stored and interpolated to obtain an accurate estimate for intensities at later points, while the second implementation always assumes a black-body intensity. Afterwards, we apply this computation method to an actual 3D hydrodynamics model in Section 4.5 in order to assess both its accuracy and speedup compared to the Feautrier solver [Feautrier, 1964], using two different treatments for the boundary condition. In our application, we noticed numerical artifacts appearing in the results for all solvers, due to the uniform angular discretization we used to compute the mean intensity. Therefore, we devised a method to adaptively sample the angular discretization of the solid angle to eliminate these numerical effects (see Section 4.4). It should be noted, however, that this new

angular discretization results in reducing the computational efficiency for the comoving solvers by a factor 2, due to the more limited reuse of the computed intensity. The comoving solver with more accurate boundary conditions is more precise than the other implementation, especially in the inner, more turbulent regions of the model, but is also significantly slower than the more approximate implementation. The former achieves speedup of a factor 2 compared to the Feautrier solver when using a uniform angular discretization, but this speedup becomes negligible when using an adaptive angular discretization. Similarly, the latter implementation obtains a speedup of a factor 5, which drops to a factor 2.5 when an adaptive angular discretization is used.

Acknowledgments

T. Ceulemans has been supported by the Research Foundation - Flanders (FWO) grant 1166722N and 1166724N. F. De Ceuster is a postdoctoral research fellow of the Research Foundation - Flanders (FWO), grant 1253223N. Leen Decin acknowledges support from the KU Leuven C1 MAESTRO grant C16/23/009 and from the KU Leuven methusalem SOUL grant METH/24/012. The authors express their thanks to J. Malfait for providing the PHANTOM model used in this paper. The authors thank the anonymous referee for their comments, which significantly helped in improving the clarity of this manuscript.

Data Availability

MAGRITTE is an open-source software library available at `https://github.com/Magritte-code/Magritte`. The code used to generate the results and figures of this paper can be found in the fork at `https://github.com/Magritte-code/Magritte_Paper_Comoving`.

Boundary condition implementation 4.A

For the comoving solver, we require extra information on outermost frequencies of the frequency discretization. These are boundary conditions. If only line radiative transfer is considered (and no continuum sources, such as dust), the intensities at frequencies far away from the line centers, can be considered constant. One possibilty is to use the initial intensities, computed at the start

of the ray, as boundary conditions. To simplify the explanation later on, we call these initial boundary conditions. At other places, we might require an intensity at a frequency we have encountered before, but have doppler shifted away from (see Fig. 4.5). This can happen because of a non-monotonic velocity field. In the next part, these are called returning boundary conditions.

As implementing both frequency matching (as defined in Section 4.3) and appropriate boundary conditions together in a numerical code is complicated[8], we describe in this appendix how we implemented this in MAGRITTE[9] Ceulemans et al. [2024], De Ceuster et al. [2022], De Ceuster et al. [2020, 2019].

The philosophy behind the implementation 4.A.1

When computing the intensity using the comoving solver, one does not know in advance which intensities (at a specific frequency and position) will be required by a returning boundary condition. To prevent the excessive use of if-clauses and to simplify the code a bit, we will first map all data needed (source functions $S_{j,i}$, optical depth increments $\Delta\tau$, frequency derivative coefficients c_{ijk}, and position index[10] to start from) to do the computations on a single ray, before computing any intensity. It turns out that the general discretization of the co-moving frame formal solution (see Eq. 4.9) is flexible enough for us to implement boundary conditions using the same computations. We will therefore exploit the formulation to make computation consistent.

For an initial boundary condition, one can put the frequency derivative coefficients c_{ijk} to 0, the optical depth $\Delta\tau_i$ very large (e.g. $\Delta\tau_i = 50$) and the source functions $S_{j,i}, S_{j+1,i}$ equal to the boundary intensity I_{bdy}. In this way, the computed intensity will be approximately equal to the boundary intensity I_{bdy}:

$$I_{j+1,i} = I_{j,i} e^{-\Delta\tau_i} + I_{\text{bdy}} \left(1 - e^{-\Delta\tau_i}\right) \simeq I_{\text{bdy}}. \tag{4.21}$$

In case of a boundary condition for which we already have computed intensities at frequencies around the requested frequency, we can interpolate these previous results by noticing that the required frequency lies between two frequencies

[8]The authors are not aware of any other literature describing a numerical method with similar implementation challenges. Therefore we consider this algorithm to be a first implementation attempt, which can be further optimized.

[9]Available at `https://github.com/Magritte-code/Magritte`

[10]The position index is required for the practical implementation of boundary conditions. If we happen to Doppler shift away from and back to a specific frequency during the computations on a single ray, we need to interpolate based on intensities computed from the previous position.

$\nu_i < \nu_{i+1}$ from which we shifted away on previous points on the ray. Then we linearly interpolate those previous intensities in order to obtain a boundary intensity. This can be done by pointing the computation data towards using the intensities at those frequencies at the corresponding previous points, and setting the frequency derivative coefficients c_{ijk} for the explicit part of the computations to $[-2/(\nu_{i+1}(x_j)-\nu_i(x_j)), 2/(\nu_{i+1}(x_j)-\nu_i(x_j)), 0]$, for the implicit part to 0 and setting the optical depth $\Delta\tau_i$ to almost zero[11], e.g. $\Delta\tau_i = 10^{-10}$. The source functions $S_{j,i}, S_{j+1,i}$ will be put to 0. The resulting equation will interpolate the intensity:

$$I_{j+1,i} = I_{j,i}e^{-\Delta\tau_i} + \frac{\Delta\nu_{\text{obs},i}}{\Delta\tau_i}\frac{2I_{j,i+1} - 2I_{j,i}}{\nu_{i+1}(x_j) - \nu_i(x_j)}\left(\frac{1 - e^{-\Delta\tau_i} - \Delta\tau_i e^{-\Delta\tau_i}}{\Delta\tau_i}\right)$$

$$\simeq I_{j,i} + \frac{\Delta\nu_{\text{obs},i}}{\nu_{i+1}(x_j) - \nu_i(x_j)}\left(I_{j,i+1} - I_{j,i}\right) \tag{4.22}$$

Determining the frequency boundary points 4.A.2

Now we still need to decide when to use boundary conditions, and to which type they belong. Assume that we want to compute intensities starting from the start of a ray, until we reach the end. For determining the boundary conditions, we will start from the last point of the ray instead. First, we note down the frequency range which is spanned by each frequency quadrature of all lines, excluding the boundary points required for the discretization in frequency space and store these boundary frequencies in a map[12], using frequency as key and spatial position index as value. Second, we compare the currently stored map of boundary points against the currently spanned frequency ranges, if any boundary frequencies are contained, then (at intensity calculation time) these are returning boundary conditions and should be interpolated using the encompassing frequencies at the current spatial position. If so, they can be removed from the map of still to match boundary frequencies. This procedure is then repeated at all previous spatial positions of the ray, from end to start. Afterwards, if any boundary points remain in the map, they are then treated as initial boundary conditions.

[11] Using $\Delta\tau_i = 0$ is not recommended, due to dividing by zero in both the source and shift term of Eq. 4.9 Note that in the limit $\Delta\tau_i \to 0$, both terms converge to a limit value, thus it is possible to evaluate these terms using Taylor expansions when $\Delta\tau_i$ close to 0.

[12] The map type should allow for duplicate keys, as it can happen that the required frequencies at different spatial positions may overlap (e.g. in case of no velocity difference between successive points). In C++, we use the std::multimap for this.

4.B Discretization of a second order accurate frequency derivative

For a second order accurate frequency derivative in ν, we need to derive the quadrature coefficients. These are found by solving the following system of equations

$$\begin{cases} \sum_{k\in\{0,1,2\}} c_{ijk} = 0 \\ \sum_{k\in\{0,1,2\}} c_{ijk}(\nu_{i+k}(x_j) - \nu_i(x_j)) = 1 \\ \sum_{k\in\{0,1,2\}} c_{ijk}(\nu_{i+k}(x_j) - \nu_i(x_j))^2 = 0 \end{cases} \tag{4.23}$$

Solving the third equation, we get

$$c_{ij1} = -c_{ij2}\frac{(\nu_{i+2}(x_j) - \nu_i(x_j))^2}{(\nu_{i+1}(x_j) - \nu_i(x_j))^2}.$$

This, we fill in into the second equation

$$c_{ij2}\left((\nu_{i+2}(x_j) - \nu_i(x_j)) - \frac{(\nu_{i+2}(x_j) - \nu_i(x_j))^2}{\nu_{i+1}(x_j) - \nu_i(x_j)}\right) = 1$$

and we find:

$$c_{ij2} = \frac{(\nu_{i+1}(x_j) - \nu_i(x_j))}{(\nu_{i+2}(x_j) - \nu_i(x_j))(\nu_{i+1}(x_j) - \nu_i(x_j)) - (\nu_{i+2}(x_j) - \nu_i(x_j))^2}$$

Thus c_{ij1} is given by

$$c_{ij1} = \frac{(\nu_{i+2}(x_j) - \nu_i(x_j))}{(\nu_{i+2}(x_j) - \nu_i(x_j))(\nu_{i+1}(x_j) - \nu_i(x_j)) - (\nu_{i+1}(x_j) - \nu_i(x_j))^2}$$

The explicit form[13] of c_{ij0} is given by

$$c_{ij0} = \frac{1}{\nu_{i+2}(x_j) - \nu_{i+1}(x_j)}\left(\frac{\nu_{i+1}(x_j) - \nu_i(x_j)}{\nu_{i+2}(x_j) - \nu_i(x_j)} - \frac{\nu_{i+2}(x_j) - \nu_i(x_j)}{\nu_{i+1}(x_j) - \nu_i(x_j)}\right)$$

and is always negative if $\nu_i(x_j) < \nu_{i+1}(x_j) < \nu_{i+2}(x_j)$.

4.C Proof of numerical stability when using frequency matching

Starting from the stability criterion (Eq. 4.17), we investigate whether the computed result could still exhibit oscillatory behavior in the worst conditions

[13]The explicit form is only useful for derivations; use $c_{ij0} = -c_{ij1} - c_{ij2}$ for any practical implementation.

possible. For this proof, we assume that $\Delta\nu_{\mathrm{obs},i} \geq 0$, but an analogous proof can be made in case of negative Doppler shifts. We start by giving a lower bound for d_i' (remember that c_{ij0} is negative), given that the frequency matching guarantees that $0 \leq \Delta\nu_{\mathrm{obs},i} \leq \nu_{i+1}(x_0) - \nu_i(x_0)$. Therefore

$$d_i' = c_{i00}\Delta\nu_{\mathrm{obs},i} \geq c_{i00}\left(\nu_{i+1} - \nu_i\right) \tag{4.24}$$

$$= \frac{\nu_{i+1} - \nu_i}{\nu_{i+2} - \nu_{i+1}}\left(\frac{\nu_{i+1} - \nu_i}{\nu_{i+2} - \nu_i} - \frac{\nu_{i+2} - \nu_i}{\nu_{i+1} - \nu_i}\right) \tag{4.25}$$

$$= \frac{\left(\nu_{i+1} - \nu_i\right)^2 - \left(\nu_{i+2} - \nu_i\right)^2}{\left(\nu_{i+2} - \nu_{i+1}\right)\left(\nu_{i+2} - \nu_i\right)} \tag{4.26}$$

$$= \frac{\left(\nu_{i+1} - \nu_{i+2}\right)\left(\nu_{i+2} + \nu_{i+1} - 2\nu_i\right)}{\left(\nu_{i+2} - \nu_{i+1}\right)\left(\nu_{i+2} - \nu_i\right)} = -\frac{\left(\nu_{i+2} + \nu_{i+1} - 2\nu_i\right)}{\left(\nu_{i+2} - \nu_i\right)} \tag{4.27}$$

$$\geq -2, \tag{4.28}$$

in which the last step uses that $\nu_{i+1} < \nu_{i+2}$. We have furthermore removed the position dependence of the frequencies on x_0 to simplify notation.

Secondly, we ignore the second term of the denominator, given that we are looking for oscillatory behavior, in which case the numerator is negative, creating another lower bound to simplify finding the minimum. The value of d_i can be (close to) zero, in case of a very wide frequency discretization at the position x_1, compared to corresponding width at position x_0.

Putting both lower bounds together, a lower bound for the stability criterion λ_i is given by

$$\lambda_i \geqslant \frac{\Delta\tau_i^2 e^{-\Delta\tau_i} - 2\left(1 - e^{-\Delta\tau_i} - \Delta\tau_i e^{-\Delta\tau_i}\right)}{\Delta\tau_i^2} \tag{4.29}$$

This function has minimum value of $\lambda_i \simeq -0.17$ at $\Delta\tau_i \simeq 1.45$ (see Fig. 4.13). Therefore, any oscillatory behavior will be quickly damped.

Angular resolution scaling test 4.D

In the main text, we restricted ourselves to using a limited amount of directions to compute the radiation field. By increasing the angular resolution, one can obtain more accurate results, but the accompanying increase in computation time might not be worth the trade-off. In this appendix, we rerun the same model as in Section 4.5, but with a higher resolution adaptive angular discretization,

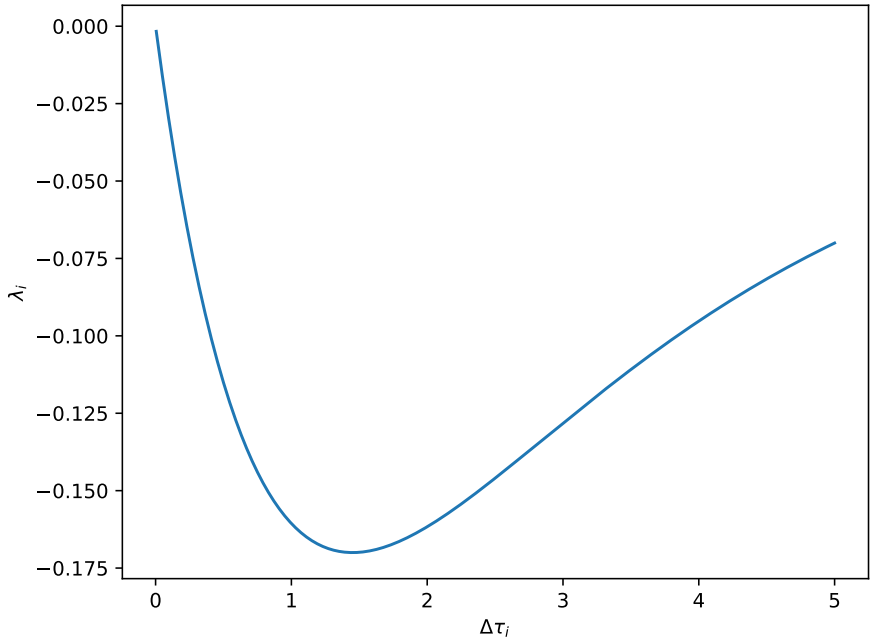


Figure 4.13: Stability criterion of the discretized comoving equation when using frequency matching, in the worst case scenario.

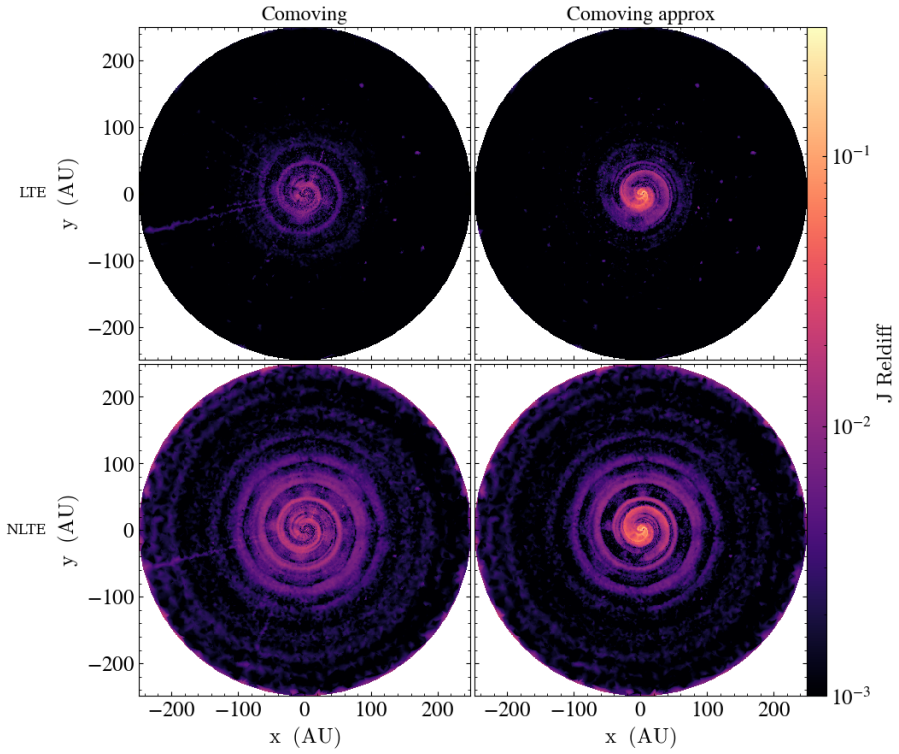


Figure 4.14: Slice plots of the relative differences between the computed mean line intensity J_{ul} for the CO J=2-1 v=0 line using the Feautrier solver and both comoving solvers, with a more dense angular discretization. The relative differences in mean line intensity have been calculated both after a single iteration starting from LTE (LTE) and after convergence (NLTE).

using parameters N= 6 and M= 4 (as defined in Listing 4.1) for a total of 156 directions.

Compared to Fig. 4.10, we in Fig. 4.14 that the ray-effect appear less pronounced for the comoving solver. Comparing the computed intensities, we find similar results to Fig. 4.12 in Fig. 4.15. The relative differences in mean line intensity J_{ul} in NLTE compared to the Feautrier solver lie below 1% for 93% and 76% of the points for comoving solver and comoving solver with approximate boundary conditions, respectively. The results differ by only a few percent compared to those in Section 4.5.2. For completeness, we add the timings in Table 4.5.

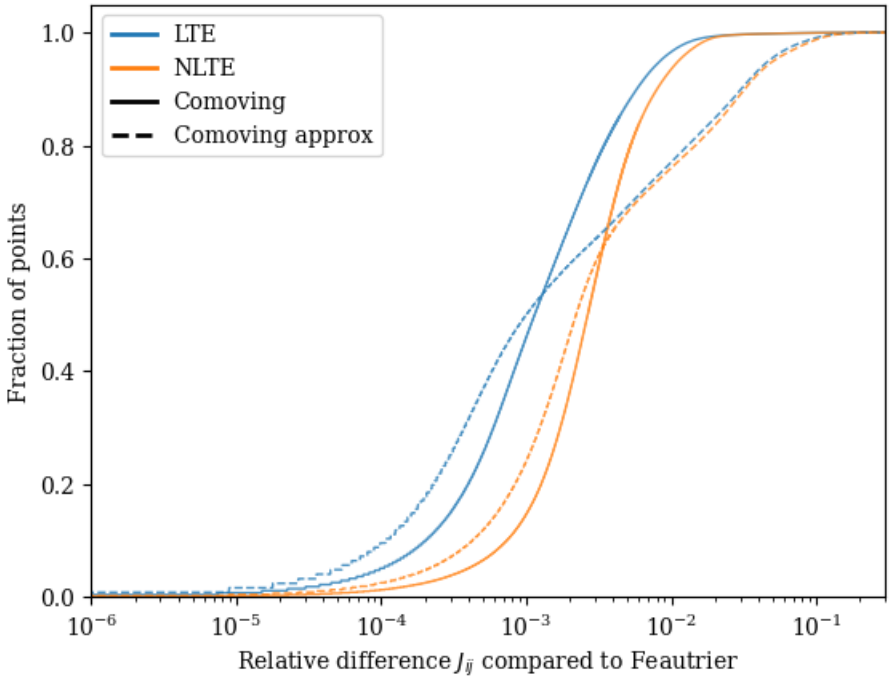


Figure 4.15: Cumulative difference between the computed mean line intensity J_{ul} for the CO J=2-1 v=0 line using the Feautrier solver and the comoving solvers with an adaptive angular discretization with increased angular resolution. Both the differences in case of a single iteration after setting the level populations n_i to LTE (in blue) and the difference after letting the level populations converge (in orange). The figure shows the results for the correct boundary conditions described in Appendix 4.A as solid lines and the results when using the approximate boundary conditions as dashed lines.

Table 4.5: Time required to compute NLTE level populations (5 iterations) on the PHANTOM model, containing 885501 points, using an adaptive angular discretization with increased angular resolution.

Algorithm	Time [s]
Feautrier	$4.1 \cdot 10^4$
Comoving	$3.8 \cdot 10^4$
Comoving approx	$1.7 \cdot 10^4$

CHAPTER 5
Magrittetorch

This chapter has not yet previously been published.

The common theme throughout this thesis is to see whether we can efficiently compute line radiative transfer on large 3D hydrodynamics models. This in order to create a framework to be able to compare observations to the synthetic observations made from simulations. Model retrieval is based on finding the optimal parameters of a model, such that the synthetic observation agrees with the astrophysical observation. Thus, we ideally have access to the following: First, an efficient way to calculate line radiative transfer, and second, a way to predict how the model output (e.g. a synthetic observation) changes depending on model input.

In this chapter, we aim to use the flexibility of machine learning libraries to achieve both goals outlined above for simulating radiative transfer. Note that we will not try to emulate radiative transfer, even though in current research, machine learning has been used to emulate various aspects of physics simulations. Examples include hydrodynamics [see e.g. Ferrer-Sánchez et al., 2024], chemistry [see e.g. Maes et al., 2024] and radiative transfer [see e.g. Chappell and Pereira, 2021]. Even though emulation provides a computationally efficient alternative to solving differential equations, it requires fitting a complicated black box function. From the resulting fit, we can no longer understand how it obtains results in any specific parameter regimes.

Even though we will not be using emulation in this thesis to accelerate radiative transfer computations, we are interested in using the machine learning libraries, as they satisfy the previously mentioned requirements. The first condition is satisfied by the machine learning libraries being optimized for computing on various hardware, including GPUs. The second one is satisfied by remembering that machine learning libraries can computing gradients on arbitrarily complex

functions. Therefore we can compute derivatives on any numerical solvers we build using these libraries. By porting our existing radiative transfer code Magritte to the machine learning library Pytorch, we can satisfy both requirements.

In this chapter, we explain the goals and design of Magrittetorch, a port of Magritte to Pytorch. In Section 5.1, we start by detailing the code design. We then explain how we ported certain CPU-parallelizable algorithms to GPU in Section 5.2, using the ray tracing algorithm and emissivity/opacity evaluation as examples. We also detail how we implement GPU memory management in Section 5.2.3, given that GPU memory is limited. Afterwards, we show how to use Magrittetorch for uncertainty quantification in Section 5.2.4. Finally, we compare the computational performance of Magrittetorch with Magritte in Section 5.3.

Code Design 5.1

The code base consists of 2 major parts: the part interacting with the user, and the computational algorithms. For the user part, we have opted for an object-oriented design, to allow for consistency checking and providing different input/output options. In contrast, the main computation algorithms are written using functional programming, as the data input must be flexible to accommodate data of differing sizes and data being stored on different compute devices.

Listing 5.1: Typical example of model creation in Magrittetorch.

```
import magrittetorch.model.model as magritte
from magrittetorch.model.geometry.geometry import GeometryType
from magrittetorch.model.geometry.boundary import BoundaryType
from astropy import units, constants # Unit conversions
from astropy.units import Quantity # For function arguments

model = magritte.Model(model_file) # Create model object, with
    default save location at 'model_file'
model.geometry.geometryType.set(GeometryType.General3D) # This is a
    3D model
model.geometry.boundary.boundaryType.set(BoundaryType.Sphere3D) #
    With a spherical boundary

# In order to make unit conversions trivial, we use astropy
    quantities as input
model.geometry.points.position.set_astropy(position) # Set point
    positions
```

```
model.geometry.points.velocity.set_astropy(velocity) # Set point velocities
model.chemistry.species.abundance.set_astropy(np.stack([nCO, nH2], axis=1))# Set species number densities
model.chemistry.species.symbol.set(np.array(['CO', 'H2'], dtype='S')) #Set species symbols; should correspond to the LAMDA file format
#Note: the dtype='S' is necessary to correctly save and read the species symbols to/from the hdf5 file

model.thermodynamics.temperature.gas.set_astropy(tmp) # Set gas temperature
model.thermodynamics.turbulence.vturb.set_astropy(trb) # Set turbulence velocity

model = setup.set_Delaunay_neighbor_lists(model) # Automatically computes and sets neighbors for each point, using a Delaunay triangulation
# For unitless quantities, we can also directly set the torch tensors
boundary_indices = torch.arange(nb_boundary, dtype = torch.int64)
model.geometry.boundary.boundary2point.set(boundary_indices) # Set which points are boundary points
# Conveniently, the remeshing function puts the boundary points in front of the positions array
model = setup.set_boundary_condition_CMB (model) # Set CMB as boundary condition
model = setup.set_uniform_rays(model, 12*4) # Number of rays for NLTE raytracing; has be of the form 12*2**n

model = setup.set_linedata_from_LAMDA_file(model, lamda_file) # Consider all transitions
model = setup.set_quadrature(model, 7) # Set number of quadrature points

model.write() #saving the model at 'model_file'
```

We have not changed much from the user interface (see Listing 5.1), compared to the original design of MAGRITTE (see Section 2.4.2). Improvements are however always possible, especially pertaining to model creation. In MAGRITTE, some friction exists when creating models, as we first need to manually define the global parameters[1] for error checking and must manually convert the units of the input data. Finally, the MAGRITTE models first need to be saved and

[1]Global model quantities such as number of points, number of lines and so on. These generally define dimensions of some data structure, thus can be used for consistency checks.

loaded in order to complete model setup. All of this can be optimized: The global parameters used for consistency checking, can be inferred from the input data itself, thus the user does not need to be queried. Finally, we can use the unit conversion package of Astropy [Astropy Collaboration et al., 2022] to automatically convert units of the required input data.

GPU Algorithms 5.2

In MAGRITTE, our computation algorithms are optimized for CPU. The individual cores of a CPU can apply different operations on different data. Therefore, parallelization over different CPU cores is typically done by distributing the workload over many threads, each performing operations on limited data. Newer hardware architectures, including GPUs, instead provide the ability to perform the same operation on a large data array. Therefore, in order to efficiently compute on other hardware architectures, we must modify our computation algorithms. In this section, we explore a few algorithms for radiative transfer for which this modification is nontrivial. We start with converting the ray-tracing algorithm in Section 5.2.1. Then, in Section 5.2.2, we explain how to efficiently evaluate the opacities/emissivities, as implemented in MAGRITTE (see Section 3.4.4). Afterwards, we explore memory management in Section 5.2.3, as the entire model might not fit on GPU memory. Finally, we explain how MAGRITTETORCH can be used for uncertainty quantification in Section 5.2.4.

Ray-tracing 5.2.1

In computer graphics, GPUs are used efficiently for ray-tracing. Therefore we might assume that also our ray-tracing algorithms might translate well from CPU to GPU. However, our requirements are significantly different: we have a different internal geometry and expect different outputs from the algorithm[2]. This leads to a major assumption not being fulfilled for GPU computing: our data are not homogeneous in shape, as different rays contain a different number of points.

We start by explaining how to convert the CPU-only ray-tracing algorithm to GPU. For comparison, we will briefly summarize the CPU ray-tracing algorithm from MAGRITTE [De Ceuster et al., 2019], as illustated in Fig. 5.1. Starting from a given position, we draw a ray in a given direction. We then find the next discretized position on the ray by calculating the distance of the neighboring

[2]We require all data needed to compute radiative transport, instead of a single intensity.

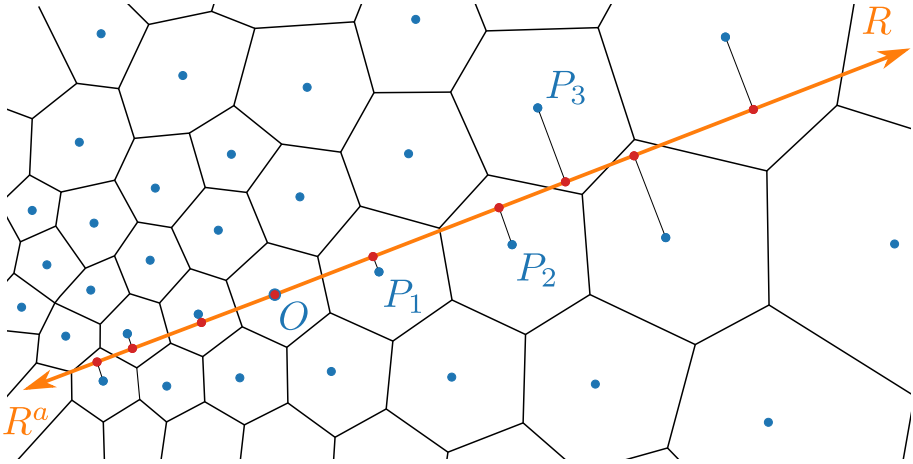


Figure 5.1: Illustration of the ray tracing algorithm used in MAGRITTE. The ray starting at the origin O in direction R traces the points P_1, P_2, until the end is reached. Figure taken from De Ceuster [2022]

points to the ray. The neighbor which lies closest to the ray, and lies in the advancing direction, will then be the next point on the ray. This repeats until the edge of the model is reached.

Porting this algorithm to perform operations on large arrays of data (see Listing 5.2) is nontrivial, as the set of neighbors at each position has a different length. We create a data structure which exactly fits the irregular data[3], map the data to this shape using advanced indexing operations, compute the distances as normal, and finally make sure that we compare the distances with respect to the ray for each starting point individually. For computational efficiency, we have pre-calculated the neighbors which lie in the positive ray-direction, i.e. $\hat{\boldsymbol{n}}_{\text{ray direction}} \cdot (\boldsymbol{x}_{\text{neighbor}} - \boldsymbol{x}_{\text{current}}) > 0$, eliminating some if-clauses during computation. Starting at line 17, we first obtain the relevant neighbors for ray tracing. At line 21, we use a custom function which generates indices according to `numpy.arange` (generating lists of indices from start to start+n) and puts the resulting segments after each other. Then for every ray origin, we duplicate the position according to the shape of the neighbors structure

[3]In the beginning, we considered using an upper bound for the number of neighbors, such that for every point we use exactly the same number of neighbors, filling the nonexistent ones with bogus data. This would make it easier to port the algorithm to GPU, as the data structure size can be assumed uniform. However, it turns out that there is a significant amount of variation between the number of neighbors at each position (at least an order of magnitude), which would result in a lot of wasted computation time on the bogus neighbors.

(lines 24). Afterwards, line 26 converts the indices in the list of neighbors to indices of the neighbors themselves. Then we compute the distances from the neighbors to the ray (perpendicular distance) and the distance to the current point on the ray (projected on the ray) on line 29. Now we compute the minimal distance to each ray on lines 32-33, using a temporary array to store them. Finally, by comparing the ray distances versus the minimal distance, we can retrieve the index of the closest neighbor to the ray on line 35-37. However, multiple neighbors might be at the same distance from a ray. In this case, we translate all neighbor matches for the closest neighbor back to ray indices $\in [0, \ldots, N_{\text{points}} - 1]$ (corresp_scatter_ids) on line 39, and only keep the first occurence per ray on line 42. This can then be translated back to the index of the corresponding neighbors and distances on lines 45-46.

Listing 5.2: GPU algorithm for finding the next points on a rays in the same direction, as implemented in MAGRITTETORCH; Data shape notation: parameters.* refers to a general model parameter, FULL CAPS refers to some constant dimension in this algorithm specifically.

```
1  def get_next_3D_geometry(self, origin_coords : torch.Tensor,
       raydirection : torch.Tensor, curr_points_index : torch.Tensor,
       device : torch.device, positions_device :torch.Tensor,
       neighbors_device : torch.Tensor, n_neighbors_device :
       torch.Tensor, cum_n_neighbors_device : torch.Tensor) ->
       Tuple[torch.Tensor, torch.Tensor]:
2    """Returns the next point on the ray in a general 3D geometry,
         given the raydirection.
3
4    Args:
5    origin_coords (torch.Tensor): 3D Coordinates of the origin
         points. Has dimensions [NPOINTS, 3]
6    raydirection (torch.Tensor): Ray direction. Has dimensions [3]
7    curr_points_index (torch.Tensor): Current point indices. Has
         dimensions [NPOINTS]
8    device (torch.device): Device on which to compute.
9    positions_device (torch.Tensor): Positions of all points in the
         model. Has dimensions [parameters.npoints].
10   neighbors_device (torch.Tensor): Linearized neighbors of all
         points in the given direction. Has dimensions
         [sum(n_neighbors_device)]
11   n_neighbors_device (torch.Tensor): Number of neighbors per point
         in the given direction. Has dimensions [parameters.npoints]
12   cum_n_neighbors_device (torch.Tensor): Cumulative number of
         neighbors per point in the given direction (starts at 0). Has
         dimensions [parameters.npoints]
13
14   Returns:
```

```
    Tuple[torch.Tensor, torch.Tensor]: Next point indices,
        accumulated distance to the next points. Both torch.Tensors
        have dimensions [NPOINTS]
    """
    masked_n_neighbors = n_neighbors_device[curr_points_index]# dims:
        [NPOINTS]
    input_size = origin_coords.size(dim=0)

    #obtain which (indices of) neighbors to check in the linearized
        neighbors list
    indices_to_check =
        multi_arange(cum_n_neighbors_device[curr_points_index],
        n_neighbors_device[curr_points_index], device) #dims:
        [sum(n_neighbors_device[curr_points_index])]
    #duplicate index of curr_points_index correspondingly
    lengthening_indices =
        torch.arange(masked_n_neighbors.size(dim=0),
        device=device).repeat_interleave(masked_n_neighbors) #dims:
        [sum(n_neighbors_device[curr_points_index])]
    lengthened_origin_coords = origin_coords[lengthening_indices, :]#
        dims: [sum(n_neighbors_device[curr_points_index]), 3]

    neighbors_to_check = torch.gather(neighbors_device, 0,
        indices_to_check) # dims:
        [sum(n_neighbors_device[curr_points_index])]

    #Get distances and compare with distance of current point;
        ydistance contains distance squared from the ray
    xdistance, ydistance =
        self.distance_in_direction_3D_geometry(lengthened_origin_coords,
        raydirection, positions_device[neighbors_to_check]) # dims:
        [sum(n_neighbors_device[curr_points_index])] for both

    #Figure out the indices of the closest points
    tempstuff = torch.zeros(input_size, device=device,
        dtype=Types.GeometryInfo) #dims: [NPOINTS]
    minydists_per_point = tempstuff.scatter_reduce(0,
        lengthening_indices, ydistance, reduce="amin",
        include_self=False) #dims: [NPOINTS]
    #broadcast these minimal distances once more, using gather
    minydists = minydists_per_point.gather(0, lengthening_indices)
        #dims: [sum(n_neighbors_device[curr_points_index])]
    #Ties can arise with the computed ydistance's, so the dimension
        can be larger than NPOINTS
    minindices = torch.nonzero(minydists == ydistance).flatten()
        #dims: [>=NPOINTS]
```

```
38    #torch.nonzero likes to transpose the matrix for some reason, so
          it needs to be flattened again
39    corresp_scatter_ids = torch.gather(lengthening_indices, 0,
          minindices) #dims: [>=NPOINTS]
40
41    #If equal distances would arise, the resulting dimension would be
          wrong. Thus I only use the first of each scatter
          (corresponding to the curr_points_indices)
42    first_result_of_each_scatter =
          torch.searchsorted(corresp_scatter_ids,
          torch.arange(input_size, device=device).type(torch.int64))
          #dims: [NPOINTS]
43    next_idx_of_neighbors_to_check =
          minindices[first_result_of_each_scatter] #dims:[NPOINTS]
44
45    next_index = neighbors_to_check[next_idx_of_neighbors_to_check]
          #dims: [NPOINTS]
46    next_dist_travelled = xdistance[next_idx_of_neighbors_to_check]
          #dims: [NPOINTS]
47
48    return next_index, next_dist_travelled
```

Efficiently evaluating opacities/emissivities 5.2.2

In Section 3.4.4, we implemented a significant optimization pertaining to the evaluation of the total emissivity η and opacity χ at any given frequency ν. By only evaluating the opacity/emissivity at the relevant line transitions[4], we eliminate a significant amount of wasted computation effort, as the other contributions to the sum are insignificant.

The original CPU algorithm evaluates the criterion for every position individually, which does not immediately translate well to GPU, as this criterion might result in a different amount of lines being needed for the same (static frame) frequency at different positions. To implement this optimization using uniformly sized data arrays, we opted for a global bound instead, based on the maximal variation in Doppler shift, which accounts for the velocity field $\boldsymbol{v}$. This allows us to precalculate which lines can be important for a given evaluated frequency. A line will be included when calculating emissivities and opacities at any position if the line frequency lies within a factor

$$\left(1 + C\left(\frac{\delta\nu}{\nu}\right)_{\max} + 2\max\left(\frac{||\boldsymbol{v}||_2}{c}\right)\right) \tag{5.1}$$

[4]Which have line profile functions with line center near the given frequency.

of the co-moving frame frequency ν_c of the origin point of the ray. In this equation, $\left(\frac{\delta\nu}{\nu}\right)_{\max}$ is the maximum relative line width over all positions. The constant C, taken to be 10 by default, is the frequency distance in units of line widths, at which we can start to ignore the line contribution.

Memory management 5.2.3

The computation algorithms in this chapter are designed such that they do operations on large vectors of data. However, GPU memory is limited and data transfer between host and device is slow (as it incurs latency). Ideally, we would like to put as much data as possible on the GPU, without exceeding memory limits. We have opted to always map basic model data (such as positions and velocities) and precomputed quantities for any NLTE iteration step (such as line opacities and emissivities) to GPU. These data are used throughout the entire calculations and only have moderate storage requirements[5].

Intermediate data vectors during computation can blow up in size. For example, an intensity vector $I(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu)$ can contain up to $O(N_{\text{angles}} N_{\text{points}} N_{\text{frequencies}})$ data points. To slightly alleviate this problem, we split the NLTE iteration by computing the intensity one angle at a time and immediately evaluating its contribution to the mean line intensity, such that we never need to keep the intensity at different angles into memory. Even with such memory optimizations, some (accumulation of) intermediate data vectors might still be too large to fit into limited GPU memory. As a simple example, for an intermediate size model with model parameters $N_{\text{points}} = 10^6$, $N_{\text{frequencies}} = 7 \cdot 40$ (7 frequency quadrature points for each of the 40 CO lines), the resulting intensity matrix has a memory size of 1.1GB[6]. Taking into account that multiple data structures of such size are needed to calculate the intensity (e.g. opacity, emissivity), the total memory cost can easily become multiple GBs.

To prevent the code from exceeding GPU memory limits, we exploit the functional programming paradigm. Our computational algorithms accept arbitrary size input data, on any given compute device. Thus instead of computing everything at once on the GPU, we can split the data in as many pieces as necessary, before doing the computation (on GPU) on each individual piece separately. Our memory management algorithm is described in Listing 5.3. Technically, one could predict the total size of intermediate data required for the different computations. However, coding this up in practice is error-prone, and the estimates needs to be changed when making changes to any algorithm.

[5] The most expensive are the line emissivities/opacities, which each contain $O(N_{\text{points}} N_{\text{line transitions}})$ data values.

[6] Assuming 32 bit floating point data types are used.

Thus, we do not know a priori how small the data segments must be. We first try to compute any algorithm using the original data vectors. If the algorithm fails due to memory errors, we try again, but this time splitting most data in two[7]. This gets repeated until either the algorithm fully succeeds or it fails a given amount of times[8]. Afterwards, the individual compute results are merged together on the host.

Listing 5.3: Psuedocode for the memory management algorithm used in MAGRITTETORCH. The actual implementation can be found in the file utils/memorymapping.py.

```
Given:  input data x vector of size N, and operation f(x)
N_splits = 1
while True:
  try:
    split x into Nsplits chuncks x_i, i∈ [0,...,N_splits − 1]
    for i in N_splits:
      compute f(x_i)
    merge results f(x_i)
    return merged results
  except Memory error:
    N_splits* = 2
    if N_splits not too large
      continue
    else:
      raise error #The data vector is too large
```

Uncertainty quantification 5.2.4

Finally, we should also explain how we can practically quantify uncertainties using PYTORCH. But first, we must clarify that we aim to capture the effect of input uncertainties on the model output. We do not consider uncertainties on the radiative transfer simulation itself. Thus our goal is to quantify how a perturbation on the input propagates through the non-linear radiative transfer calculations. We do this using the forward mode auto differentiation module `torch.autograd.forward_ad`.

Given that we have programmed the entire model using Pytorch, applying perturbation analysis is straightforward, as illustrated in Listing 5.4. We start

[7]Not all data can be subdivided. As an example, for ray-tracing, we pre-compute the neighbors of every point in a given direction and store them on GPU.

[8]This can happen if the non-subdividable data fills all available memory. In this case, the model is simply too large for the used hardware.

by defining a forward gradient on any input parameter of interest. This gradient will get propagated throughout all computations, using a first order accurate approximation. Any model output, which indirectly depends on the specified input parameter, will then also contain a gradient. The obtained gradient is then a first order accurate representation of the change dictated by the input.

Listing 5.4: Illustration of how to use MAGRITTETORCH for uncertainty quantification. The temperature input gradient gets propagated throughout all computations. We color the lines required for the uncertainty quantification in red.

```
from magrittetorch.model.model import Model
import magrittetorch.algorithms.solvers as solvers
import torch
import torch.autograd.forward_ad as fwAD

with fwAD.dual_level():
  model = Model(inputpath)
  model.read()

  # add gradient to model input, before computing any derived
      quantities
  temp = model.thermodynamics.temperature.gas.get()
  dtemp = 0.1*temp
  dual_input_temp = fwAD.make_dual(temp, dtemp)
  model.thermodynamics.temperature.gas.set(dual_input_temp)

  # finish model setup
  model.dataCollection.infer_data()

  # If you have a GPU, we can use it to speed up the computations
  device = torch.device("cuda:0" if torch.cuda.is_available() else
      "cpu")

  # Compute NLTE level populations
  solvers.compute_level_populations(model, device=device,
      max_n_iterations=20)
  lspec = model.sources.lines.lineProducingSpecies[0]

  # For this example, we will compute the cooling rates,
  cooling_rates =
      lspec.compute_line_cooling(lspec.population.get(),
      torch.device("cpu"))
  # which are also dual numbers, as the gradients get propagated
      throughout all calculations
  dual_cooling_rates = fwAD.unpack_dual(cooling_rates)
```

Computing a simulation with the forward gradient enabled takes twice the amount of computation time, when compared to a simulation without adding the gradient[9]. Note that this uncertainty quantification method only allows for a single input gradient at a time. Thus it can efficiently answer general statements such as: "what happens when we increase the temperature in the entire model by 5%?", but it cannot efficiently evaluate the effect of parameter tweaks at individual positions, as every tweak requires a full simulation to estimate its effects on the radiation field. An illustration on how to use MAGRITTETORCH for uncertainty quantification is given in Section 6.5. In theory, we could also use backwards derivatives to investigate the simulation result, telling us for example from which part of the model a specific feature in the synthetic observation might arise. However, as explained in the next paragraph, memory limitations prevent us from doing this.

It is theoretically possible to use the differentiable models to automatically fit the input to observations. However, in practice, the memory requirement of the matrices involved in the backpropagation are too high. For example, when computing mean line intensities (which averages intensities according to equation 4.20), we essentially obtain a matrix of dimensions $N_{\text{points}}N_{\text{lines}} \times N_{\text{points}}N_{\text{frequencies}}N_{\text{angles}}$, which for 3D models is too large to store in memory[10].

Computational results 5.3

In this section, we test the GPU performance of MAGRITTETORCH using the AGB model previously described in Chapter 3, reduced using the method described therein. We compare our GPU implementation versus the CPU implementation of MAGRITTE. Given that MAGRITTE has seen some updates which might have an impact on the computational performance, we will compare MAGRITTETORCH against the latest version of MAGRITTE. The main computational difference between both versions is the inclusion of the interpolation described in Section 3.4.5 in the newer version. In Table 5.1, we have put the wall clock time that each code (version) needs to compute NLTE level populations. In order to more easily compare between the different code versions, we did not enable Ng-acceleration.

[9] So why do we not instead do simulations for two slightly different inputs, without any fancy gradient propagation? Calculating twice, with slightly differences in model input, most certainly gives two (different) output values. However, the perturbation on the input propagates in a very non-linear manner to a perturbation on the output. Using these gradients, we have a guarantee that we know the local behavior of any perturbation.

[10] At least for a non-sparse implementation. At time of testing, not all functions in PYTORCH had sparse implementations available.

Table 5.1: Time required for convergence MAGRITTE and MAGRITTETORCH. All tests are run on a laptop, equipped with an Intel i7-11850H CPU and an Nvidia RTX A2000 mobile GPU. MAGRITTE is run on CPU, while the MAGRITTETORCH run utilizes the GPU. As the wall clock time is proportional to the number of NLTE iterations, we have also added this information.

Code version	# NLTE iterations	Wall clock time [s]
MAGRITTE 0.9.9	6	$3.5 \cdot 10^3$
MAGRITTETORCH	6	$1.0 \cdot 10^3$

We notice a difference of a factor 3.5 in computation time required for MAGRITTE and MAGRITTETORCH. We do remark, however, that we require a significant amount of interpolation points in this strongly reduced model. We expect the impact of interpolation on the computation time to be lower for less reduced models. Compared to MAGRITTE, we have implemented one additional performance improvement in MAGRITTETORCH: using 32-bit floats for most computations[11], which improved computational performance by a factor 2. We note that this modest improvement in computation time is lower than we expect, as one typically gains (an) order(s) of magnitude improvement in computation time by porting code to GPU, given the massive difference in compute units between the devices[12]. This result might partially arise from running interpreted python code versus compiled C++ code, or from the specific implementation, which contains a significant amount of advanced indexing operations, and might not be ideal to run on GPU.

To evaluate the accuracy, we compare the mean line intensity J_{ij} and the level populations n_i between MAGRITTE and MAGRITTETORCH. We find that for 94% of the data points, we have differences less than 10% difference in J_{ij} between both codes. The difference in the level populations n_i are negligible, with a mean absolute relative difference of $6 \cdot 10^{-5}$. To verify that these relatively large differences for mean line intensity J_{ij} only result in a tiny differences for the level populations, we fill in the mean line intensity computed using MAGRITTE in MAGRITTETORCH. The resulting level populations have a mean absolute relative difference of $3 \cdot 10^{-7}$ with respect to those computed using MAGRITTE. We conclude that we compute slightly different mean line intensities using both codes, but still obtain the same level populations n_i.

[11] We only kept the computations related to the level populations n_i in 64 bit floats, as Ng-acceleration is quite sensitive to the numerical accuracy.

[12] Although you cannot compare CPU cores versus the GPU equivalent apples to apples, given their significantly different architectures and capabilities.

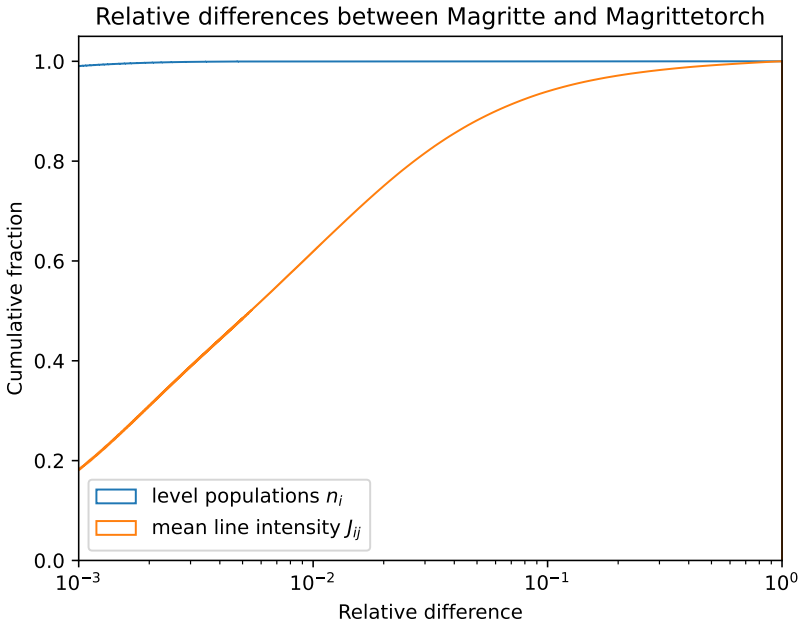


Figure 5.2: Cumulative distribution plot of the relative differences between the computed NLTE results of MAGRITTE and MAGRITTETORCH. Higher values mean that the results are more similar.

Future work 5.4

MAGRITTETORCH is based on Magritte version 0.7.2. The following features are missing in order to be up to date with MAGRITTE: (1) Adaptive ray tracing (see Section 4.4). As stated in that section, using a uniform angular discretization might cause rays to appear in the intensity field, originating from small, bright regions. We can implement this feature, at the cost of reducing the efficiency of the ray tracing algorithm, as we will no longer be able to precompute neighbors per direction; (2) Interpolation in case of high source function gradients (see Section 3.4.5). During the NLTE computations, we can encounter significant differences in the line opacity between successive points, in some lower density regions. As these relative differences can multiple orders of magnitude large, the intensity might not be correctly computed in these regions. However, we do note that adding a continuum contribution to the opacity (from e.g. dust), might also solve this issue in our models. We currently do not allow adding additional ghost points during the calculation step, as this would introduce additional if-clauses during the main calculation step. It could be possible to refactor the code to precompute the path of each ray, but this approach might have consequences for the memory management implementation.

Finally, one might be able to test full model fitting, when all required PYTORCH functions have a sparse implementation. Currently, the back-propagation algorithm tries to compute dense giant matrices, which exceed any reasonable amount of available GPU memory.

CHAPTER 6

CO cooling

A slightly reformatted version of this chapter has been submitted as:

A rotational line CO cooling rate prescription for AGB outflows

Ceulemans, T., De Ceuster, F., Vermeulen O., Decin, L.

in Astronomy & Astrophysics.

CRediT authorship contribution statement

T. Ceulemans: Writing – original draft, Visualization, Validation, Software. F. De Ceuster: Writing – review & editing, Supervision, Funding acquisition. O. Vermeulen: Methodology, Writing – review & editing, L. Decin: Writing – review & editing, Supervision, Funding acquisition.

Abstract

Asymptotic Giant Branch (AGB) stars significantly contribute to the chemical composition of the universe. In their outflows, complex chemistry takes place, which critically depends on the local temperature. Therefore, if we want to accurately model the AGB environment, we need accurate cooling rates. The CO molecule is abundant in AGB outflows, and has a dipole moment, which enables it to cool through emission from its rotational transitions. We therefore expect it to significantly contribute to cooling in this environment, even at low temperatures (10 K $\leqslant T \leqslant$ 3000 K). Currently, CO cooling rates are available for ISM-like conditions, which encompasses a different parameter regime, with lower densities and velocities, compared to AGB winds. Therefore, these ISM cooling rates might not be applicable to the AGB regime. In this paper, we compute CO cooling rates for hydrodynamics simulations of AGB outflows. To evaluate the net cooling rate, we calculate the energy level distribution of CO self-consistently, using the non Local Thermodynamical Equilibrium (NLTE) line radiative transfer code MAGRITTE. We verify whether already existing CO cooling rate prescriptions for the interstellar medium (ISM) are applicable for this regime. We noticed slight differences between these prescriptions and our calculated cooling rates, with the best prescription having a mean relative difference of 0.14 dex and a standard deviation of the error about 0.18 dex. Therefore, we propose a new CO cooling rate prescription for the AGB environment and we study how the computed cooling rate varies depending on input parameters.

Introduction 6.1

Asymptotic Giant Branch (AGB) stars are low-to-intermediate mass stars at the end of their lives. It is estimated that $\sim 85\%$ of dust and $\sim 35\%$ of gas in the interstellar medium originate from their outflows [Tielens, 2005]. In order to numerically evaluate the contribution of AGB outflows on the chemical evolution of the universe, we need to apply non-equilibrium chemistry to the stellar wind [see e.g. Maes et al., 2023]. An important input for the chemical network is the temperature, which is influenced by hydrodynamics, chemical cooling, and also by radiation transport. In this paper, we focus on the cooling impact of CO rotational transition lines, as CO is the second most abundant molecule in the AGB outflows, aside from H_2. In contrast to H_2, CO has a net dipole moment, and thus easily excitable rotational energy transitions. Therefore CO can enable radiative cooling even at lower temperatures.

Radiative cooling forms an important part of the thermal balance of astronomical objects, starting from the interstellar medium [Neufeld and Kaufman, 1993] to supernovae [McLeod et al., 2024]. However, no cooling prescriptions have yet focused on the AGB environment. The environment most similar to AGB outflows is the Interstellar Medium (ISM), for which cooling rates prescription have been computed in various publications, see e.g. Neufeld and Kaufman [1993], Omukai et al. [2010] and Whitworth and Jaffa [2018]. In previous papers, simplifying assumptions have been made about the geometry of such objects, including spherical symmetry. However, for the AGB regime, we cannot assume such simplifying geometry, as significant asymmetries have been observed in the outflows [Decin, 2020]. Furthermore, the physical conditions in the ISM are different from the conditions in the AGB outflow, which can limit the applicability of the previously mentioned cooling rate prescriptions.

In this paper, we will compute CO cooling rates based on hydrodynamics models of AGB winds siumlted with Phantom [Price et al., 2018]. For this, we use Magritte [see e.g. De Ceuster et al., 2019, 2020, De Ceuster et al., 2022, Ceulemans et al., 2024] in order to calculate the non Local Thermodynamical Equilibrium (NLTE) radiation field, which determines the cooling. Afterwards, we will compare the calculated cooling rates with pre-existing prescriptions [Neufeld and Kaufman, 1993, Omukai et al., 2010, Whitworth and Jaffa, 2018]. However, we find that these cooling prescriptions are not fully suited to the AGB environment. Therefore, we propose a new fit in this paper.

This paper is structured as follows: a brief introduction on how to calculate CO cooling rates is given in Section 6.2. We introduce the literature cooling rate prescriptions relevant for this paper in Section 6.3. We explain the model setup and calculate cooling rates in Section 6.4. Afterwards, we discuss the impact of perturbations on the model input to the computed cooling rate in Section 6.5. Finally, we discuss the limitations of our cooling fits in Section 6.6.

Computing the cooling rate 6.2

In order to compute the net radiative cooling rate Λ[W m^{-3}], we need to create radiative transfer models. For this, we use Magritte [De Ceuster et al., 2022]. As input for our models, we use hydrodynamics models of AGB winds, which contain temperatures, densities, and a velocity field. We interpret the density as H_2 density and impose a constant CO/H_2 ratio, which is valid in the outflow as long as no significant photo-dissociation occurs. We then use this Magritte model to compute NLTE line radiative transfer.

Our workflow is as follow: our initial guess for the energy level distribution of

the species CO is given by the assumption of local thermodynamical equilibrium (LTE). We iteratively compute the radiation field $I(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu)$[W sr^{-1}m^{-2}Hz^{-1}] based on the level populations n_i[m^{-3}], and the level populations using statistical equilibrium. To speed up this iterative process, we use Accelerated Lambda Iteration (ALI) [see e.g. Rybicki and Hummer, 1991] and adaptive Ng-acceleration [as described in Ceulemans et al., 2024]. At the end of this iterative process, we have obtained self-consistent values for the radiation field I and the level population n_i. Then, in order to calculate the net radiative cooling rate Λ, we have two options. We subtract the emitted intensity from the absorbed intensity in order to obtain the net cooling rate Λ, as done in Carlsson and Leenaarts [2012],

$$\Lambda = \int \chi(\nu)(S(\nu) - I(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu))\mathrm{d}\nu\mathrm{d}\Omega. \tag{6.1}$$

In which χ[m^{-1}] is the opacity, S[W sr^{-1}m^{-2}Hz^{-1}] is the source function and I is the intensity. In high optical depth regimes, the intensity I is almost equal to the source function S, but not exactly. Therefore Eq. 6.1 can suffer from numerical cancellation errors. In line radiative transfer, an equivalent prescription can be found when assuming statistical equilibrium. In this case, the equation from Sahai [1990] is valid:

$$\Lambda = \sum_{\text{transitions from level i to j}} (n_j C_{ji} - n_i C_{ij}) h\nu_{ij} \tag{6.2}$$

in which n_i are the number densities at the different energy levels and C_{ij}[s^{-1}] are the collisional rates. In this paper, we calculate the cooling rate Λ using the latter equation.

Existing cooling prescriptions 6.3

In the literature, several CO molecular line cooling prescriptions already exist. However, they have been mainly focused on ISM-like conditions. As the environment around AGB stars is significantly different from the ISM conditions, with higher densities and velocity gradients, we first check whether the current prescriptions are valid in this environment. We consider the prescriptions in Neufeld and Kaufman [1993], Omukai et al. [2010] and Whitworth and Jaffa [2018] for this paper. These prescriptions mostly use the following parameters: the H_2 and CO number densities n_{H_2}, n_{CO}[m^{-3}], the local gas temperature T[K] and the velocity divergence $|\nabla \cdot \boldsymbol{v}|$ [s^{-1}], which is related to the Sobolev

optical depth $\tau_{\text{Sobolev}}[.]$ [Sobolev, 1960],

$$\tau_{\text{Sobolev}} = \chi \frac{c}{|\nabla \cdot \boldsymbol{v}|}, \tag{6.3}$$

in which c is the speed of light and χ is the frequency integrated opacity. The densities directly affect the cooling rates (see Eq. 6.2), as the CO density affects the level populations n_i of CO, and the H_2 density is used for the calculation of the collisional rates. The temperature determines the relative distribution of the level populations in LTE, and still impacts them in NLTE. The velocity divergence $|\nabla \cdot \boldsymbol{v}|$ limits how much radiation can escape from the molecular emission lines and therefore imposes an upper bound on the cooling rate [Neufeld and Kaufman, 1993].

In the following subsections, we briefly explain the literature cooling rate prescriptions explored in this paper. We note, however, that we will make different assumptions during our computations, calculate the cooling in a different astrophysical regime, and use different data when computing the cooling rates. Therefore, any difference we observe with the net cooling rate we compute does not invalidate their applicability to their intended astrophysical regime.

Neufeld & Kaufmann 1993 6.3.1

In Neufeld and Kaufman [1993], cooling rates have been calculated for H_2O, H_2 and CO using an escape probability method to compute the level populations. They assume the velocity field to have a large gradient and to be either monotonically increasing or decreasing, such that the Sobolev approximation is valid [Sobolev, 1960]. In this way, NLTE radiative transfer can be computed using only local information. Their rotational line cooling for CO is based on the first 75 rotational transitions, with energy levels taken from HITRAN database [Rothman et al., 1987], and uses the collision rate coefficients from Viscuso and Chernoff [1988]. Their rotation cooling prescription is valid between temperatures of 100 K until 2000 K and values of $\tilde{N}(\text{CO})$ between 10^{15}s m^{-3} en 10^{20}s m^{-3} [Neufeld and Melnick, 1991], which they define as

$$\tilde{N}(\text{CO}) = \frac{n_{\text{CO}}}{|\nabla \cdot \boldsymbol{v}|}, \tag{6.4}$$

in which n_{CO} is the CO number density. Their cooling prescription is given by

$$\Lambda = L_{\text{N93}} n(\text{H}_2) n(\text{CO}), \tag{6.5}$$

in which the cooling rate coefficient $L_{\mathrm{N93}}[\mathrm{W\ m^3}]$ is given by

$$\frac{1}{L_{\mathrm{N93}}} = \frac{1}{L_0} + \frac{n(\mathrm{H_2})}{\mathcal{L}_{\mathrm{LTE}}} + \frac{1}{L_0}\left[\frac{n(\mathrm{H_2})}{n_{1/2}}\right]^{\alpha}\left(1 - \frac{n_{1/2}L_0}{\mathcal{L}_{\mathrm{LTE}}}\right) \tag{6.6}$$

and the coefficients L_0, $\mathcal{L}_{\mathrm{LTE}}$, $n_{1/2}$ and α are tabulated in Neufeld and Kaufman [1993].

Omukai et al. 2010 6.3.2

Omukai et al. [2010] models the ISM during collapse into a protostar. To calculate the CO cooling, they also use an escape probability method to compute the energy level distribution [see Omukai, 2000] in NLTE, and adopt the CO molecular data from the LAMDA database [Schöier et al., 2005], but do not specify which version of the data they use. In contrast to Neufeld and Kaufman [1993], they do not use an approximation for the escape probability, but evaluate it using the column density of their spherically symmetric model.

Their CO cooling rate prescription is based on Neufeld and Kaufman [1993], with the difference that the parameter range for the temperature has a lower bound of 10 K and their definition $\tilde{N}$ has been changed to $\tilde{N} = N/\sqrt{2k_bT/m_{\mathrm{CO}}}$, in which $N[\mathrm{m^{-2}}]$ is the column density, $k_b[\mathrm{J\ K^{-1}}]$ is Boltzmann's constant and $m_{\mathrm{CO}}[\mathrm{kg}]$ is the CO particle mass. This quantity can be derived from the optical depth τ (ignoring constant factors), when assuming no velocity field to be present,

$$\tau = \int \chi_{ij}\phi_{ij}dx \sim \frac{N}{\delta\nu_{ij}} \sim \frac{N}{\sqrt{2k_bT/m_{\mathrm{CO}}}}. \tag{6.7}$$

in which $\chi_{ij}[\mathrm{Hz\ m^{-1}}]$ is the integrated line opacity, $\phi_{ij}[\mathrm{Hz^{-1}}]$ is the line profile function and $\delta\nu_{ij}[\mathrm{Hz}]$ is the line width. However, Omukai et al. [2010] do not mention why they changed the definition of $\tilde{N}$. For this paper, we evaluate their cooling fit using the definition of Neufeld for $\tilde{N}$ instead, given that significant velocity fields are present in our models.

Whitworth & Jaffa 2018 6.3.3

In Whitworth and Jaffa [2018], they aim to model the effect of CO cooling on the structure of molecular clouds in the ISM. As input for their cooling fit, they use the cooling rate results of Goldsmith and Langer [1978]. In that paper, the authors use an escape probability method to determine the level populations n_i and the net cooling rate Λ, and utilize a Sobolev approximation [Sobolev,

1960] to evaluate the optical depth τ. This allows them to approximate NLTE radiative transfer locally.

In contrast to the previous cooling prescriptions, Whitworth and Jaffa [2018] deduce their cooling rate prescription from first principles in two separate regimes, defining a theoretically motivated fit function for both the low and high density regime. In the low density regime, they find

$$\Lambda_{\rm LO} = \lambda_{\rm LO} X_{\rm CO} n_{\rm H_2}^2 T^{3/2}, \tag{6.8}$$

in which $\lambda_{\rm LO}$ is the fit coefficient, $X_{\rm CO} = n_{\rm CO}/n_{\rm H_2}$, $n_{\rm H_2}$ is the H_2 density, while in high densities, they obtain

$$\Lambda_{\rm HI} = \lambda_{\rm HI} T^4 \left|\nabla \cdot \mathbf{v}\right|, \tag{6.9}$$

in which $\lambda_{\rm HI}$ is the fit coefficient. Afterwards, they combine both formulae using

$$\frac{\Lambda}{n_{\rm CO}} = \left(\left(\frac{\Lambda_{\rm LO}}{n_{\rm CO}} \right)^{-1/\beta} + \left(\frac{\Lambda_{\rm HI}}{n_{\rm CO}} \right)^{-1/\beta} \right)^{-\beta} \tag{6.10}$$

in which β is a fit parameter. Their fit prescription is valid for temperatures between 10 and 60 K and for values of $\frac{n_{\rm CO}/n_{\rm H_2}}{|\nabla \cdot \boldsymbol{v}|}$ between 10^5 and 10^9 s.

All cooling rate prescriptions described in this section treat radiative transfer using the escape probability method, and therefore limit the geometry to which it can be applied. Given that we calculate radiative transfer on 3D AGB wind models, we use a different calculation method, and might therefore obtain slightly different results for the net cooling rate.

CO Cooling rates for AGB outflows 6.4

The goal of this paper is to obtain a CO cooling rate prescription for AGB binary simulations. We therefore took a variety of Phantom [Price et al., 2018] AGB binary simulations with various circular orbit configurations and also a single star model for verification purposes. The Phantom smoothed particle hydrodynamics (SPH) models generate a wind by periodically launching particles from the star [as explained in Siess et al., 2022] with initial velocity as given in Table 6.1. All models use a mass loss rate of $1 \cdot 10^{-7} M_\odot$/yr. The final binary model in this Table, v15a35, is taken from Malfait et al. [2024a], where it was named v15m06.

Table 6.1: Model properties of the PHANTOM models used in this paper. $v_{\rm in}$ denotes the initial velocity, $M_\star$ denotes the stellar mass, $M_{\rm com}$ denotes the companion mass and a the semi-major axis.

Name	$v_{\rm in}$ [km/s]	$M_\star$ [$M_\odot$]	$M_{\rm com}$ [$M_\odot$]	a [AU]
v05a06	5	1.5	1	6
v05a20	5	1.5	1	20
v20a06	20	1.5	1	6
v20a20	20	1.5	1	20
v15a35	15	2	0.6	35
v10_single	10	1.5		

General model setup 6.4.1

Starting from a PHANTOM SPH model, we calculate the NLTE CO level populations using Magritte [Ceulemans et al., 2024, De Ceuster et al., 2022], reducing the model size for increased computational efficiency, using the procedure detailed in Ceulemans et al. [2024], with maximal recursion level $l = 12$ and maximal variation $r_{\rm max}$. We utilize the ^{12}CO line data from the LAMDA database[1] [Schöier et al., 2005], in which the collision rates are taken from Yang et al. [2010]. We only consider rotational transitions in this paper, as collisional data for the CO vibrational transitions is currently only available up to 300 K [Yang et al., 2016]. For each of the 37 hydrodynamics snapshots from these AGB models, taken at different times, we generate different MAGRITTE models, each with a different, but constant CO/H_2 abundance $\in \{1, 3, 5, 8, 10\} \cdot 10^{-4}$. This range spans the values used in the literature [see e.g. Khouri et al., 2014, Danilovich et al., 2014] for the AGB envelope. We have limited our models to a radius of 500 AU around the primary star, where we assume no significant CO dissociation to have taken place. In this way, a constant CO/H_2 abundance ratio should be a valid assumption. For most of this paper, we restrict ourselves to only using the binary models, because including this data would add too many data points in less interesting regions of the parameter space. We will however validate in 6.A that our obtained fit of Section 6.4.3 works on the single star data.

Our combined data set of AGB binary models contains 185 MAGRITTE models, having 18 million data points in total, with each data point consisting of a vector containing the gas temperature T, H_2 density $n_{\rm H_2}$, CO density $n_{\rm CO}$, velocity gradient $|\nabla \cdot \boldsymbol{v}|$ and computed net cooling rate Λ. The parameter ranges spanned by these models can be found in Table 6.2. As inputs for the

[1] available at `https://home.strw.leidenuniv.nl/~moldata/`

Table 6.2: Parameter limits for this paper

	Min	Max
Temperature [K]†	14	3000
H_2 density [m^{-3}]	$1.4 \cdot 10^8$	$1.9 \cdot 10^{15}$
CO/H_2 ratio	10^{-4}	10^{-3}
$n(CO)/\lvert\nabla \cdot v\rvert$ [s m^{-3}]	$1.5 \cdot 10^{13}$	$4.2 \cdot 10^{23}$

† Limited by CO collisional data. SPH model temperatures can exceed 3000K.

fit function, densities, temperatures and the velocity divergence $|\nabla \cdot \boldsymbol{v}|$ at each position are required, all of which can be extracted from the SPH model.

Existing cooling rate prescriptions 6.4.2

None of the existing prescriptions offer a parameter range which encompasses the entire parameter range of our models (see Figs. 6.1, 6.2). Therefore, we do not expect the cooling rate prescriptions to be appropriate for the entire parameter range of our models. However, all of them offer a reasonable proxy for the computed cooling rate (see Figs. 6.3, 6.4, 6.5). We show the relative differences with the computed cooling rate in Figs 6.6, 6.7 and 6.8 respectively. From these, we observe that on average, the fit from Neufeld and Kaufman [1993] performs best, with a mean relative difference of 0.013 dex, however, some very far away outliers exist. In contrast, the Omukai et al. [2010] prescription slightly underestimates the cooling rate, with a factor of 0.06 dex and has a standard deviation on the error of about 0.11 dex. Finally, the fit from Whitworth and Jaffa [2018] slightly overestimates the cooling rate on average (by 0.14 dex), and has a higher standard deviation of about 0.18 dex. To complete our analysis, we check the number of outliers for each fit prescription, which we define as

$$|\log_{10}(\Lambda_{\text{fit}}/\Lambda_{\text{this work}})| > 2. \tag{6.11}$$

Each outlier therefore corresponds to a prediction for the cooling rate which is off by at least two orders of magnitude. As seen in Table 6.3, the Whitworth and Jaffa [2018] prescription has the least outliers. However, as any outlier present in the fit function is one too many, we will create a new fit in Section 6.4.3.

New cooling rate prescription 6.4.3

We fit the data according to a polynomial fit function in logarithmic space. As we do not know how many fit parameters we require a priori, we allow the

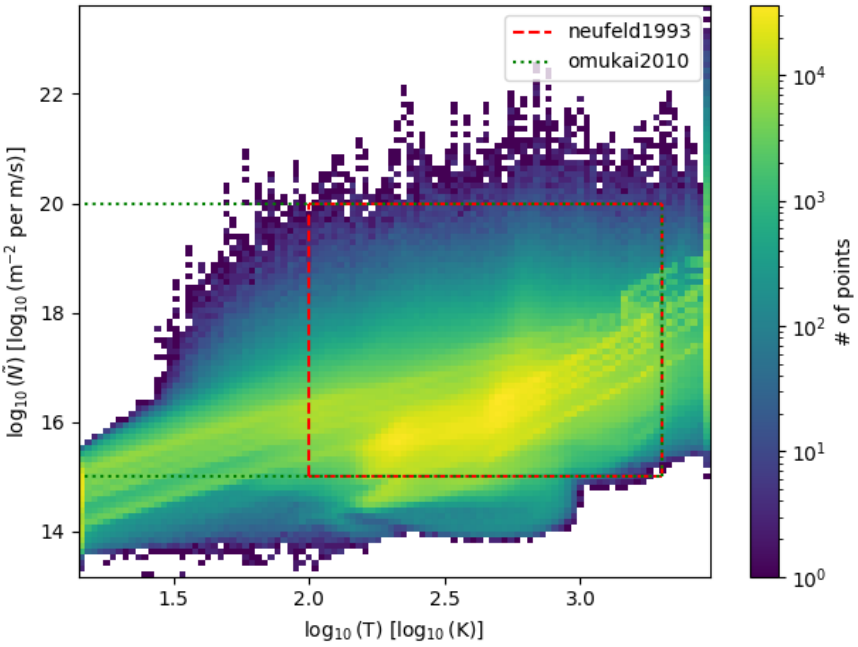


Figure 6.1: Distribution of our model points, compared to the parameter ranges in which the Neufeld and Omukai prescriptions are valid. A significant fraction of our data points lie outside the ranges covered by Neufeld and Kaufman [1993] and Omukai et al. [2010].

Table 6.3: Number of outliers for each fit prescription, out of 18 million data points in our data set.

Fit	# outliers
Neufeld	698
Omukai	5245
Whitworth	136

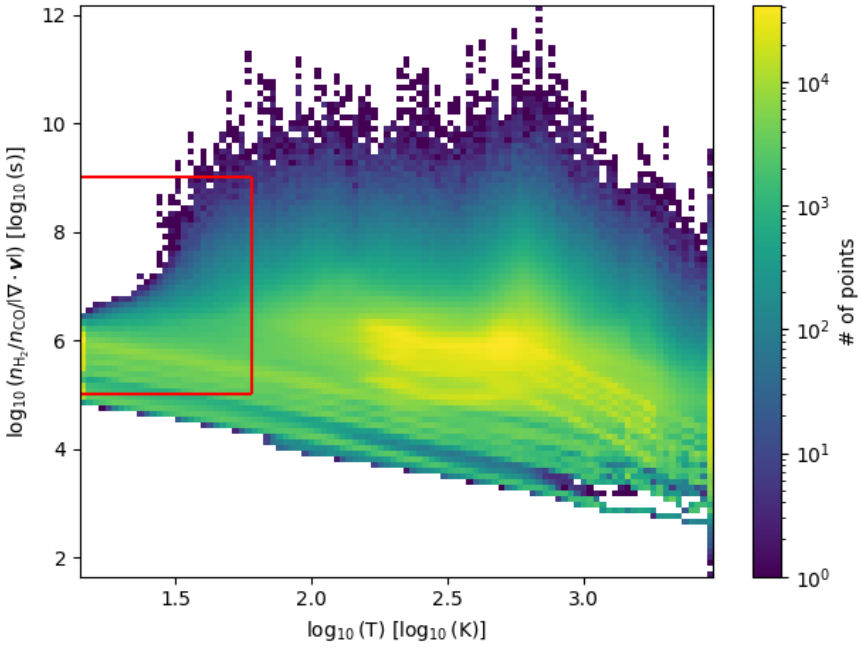


Figure 6.2: Distribution of our model points, compared to the parameter range (box) in which the Whitworth prescription is valid. Most of our data points lie outside the range covered by Whitworth and Jaffa [2018].

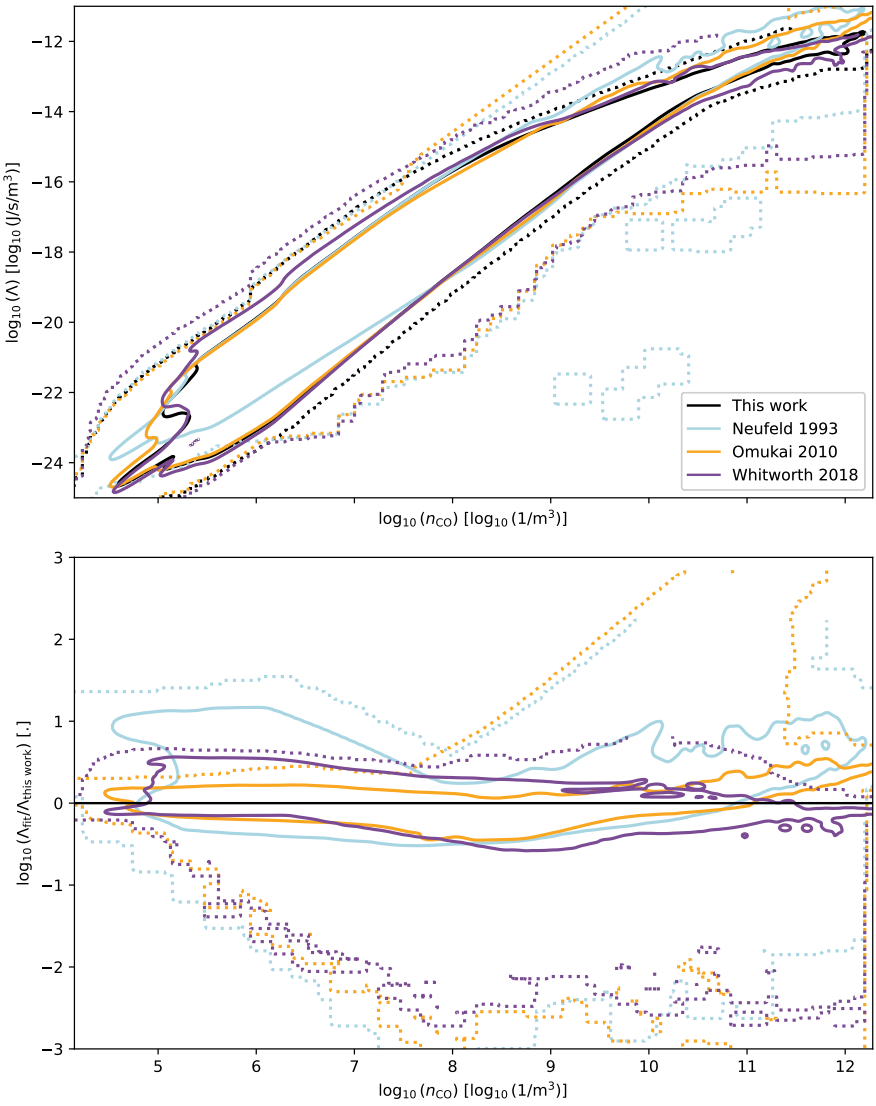


Figure 6.3: Contour plots of the cooling rate versus the CO density, smoothed using a Gaussian kernel. The solid line contains 99% of the data points. The dotted line denotes the outer contour where data is present. Bottom: contour plot of the relative differences between the fits and our computed cooling rate.

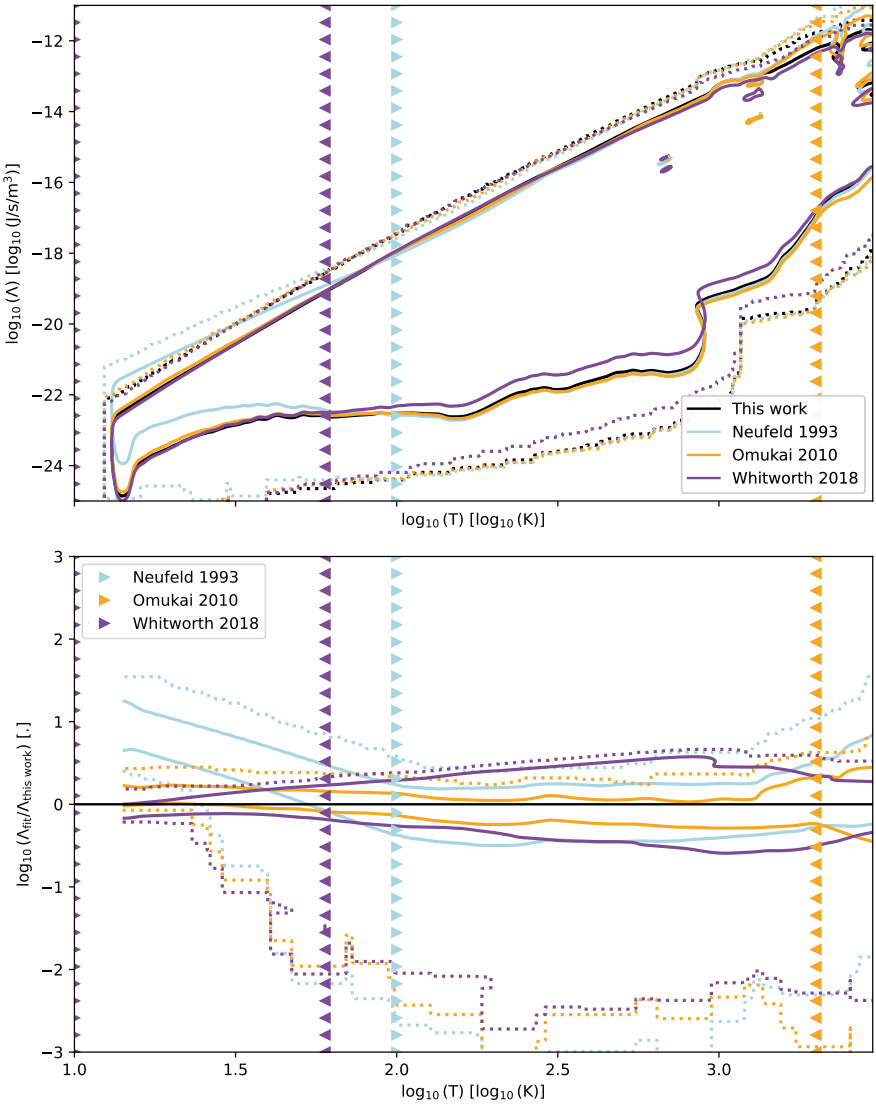


Figure 6.4: Contour plots of the cooling rate versus the temperature, smoothed using a Gaussian kernel. The solid line contains 99% of the data points. The dotted line denotes the outer contour where data is present. The vertical arrow lines denote the temperature bounds for which the literature fit functions are defined. Bottom: contour plot of the relative differences between the fits and our computed cooling rate.

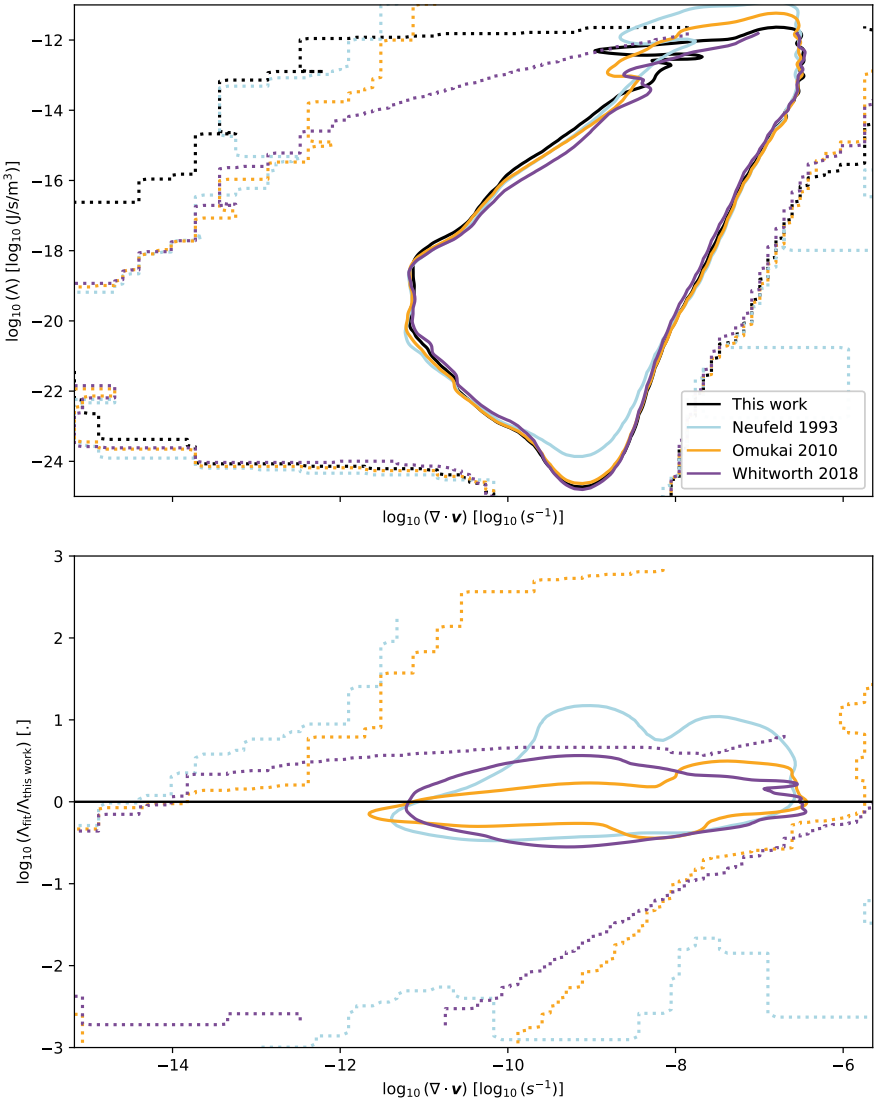


Figure 6.5: Contour plots of the cooling rate versus the velocity divergence, smoothed using a Gaussian kernel. The solid line contains 99% of the data points. The dotted line denotes the outer contour where data is present. Bottom: contour plot of the relative differences between the fits and our computed cooling rate.

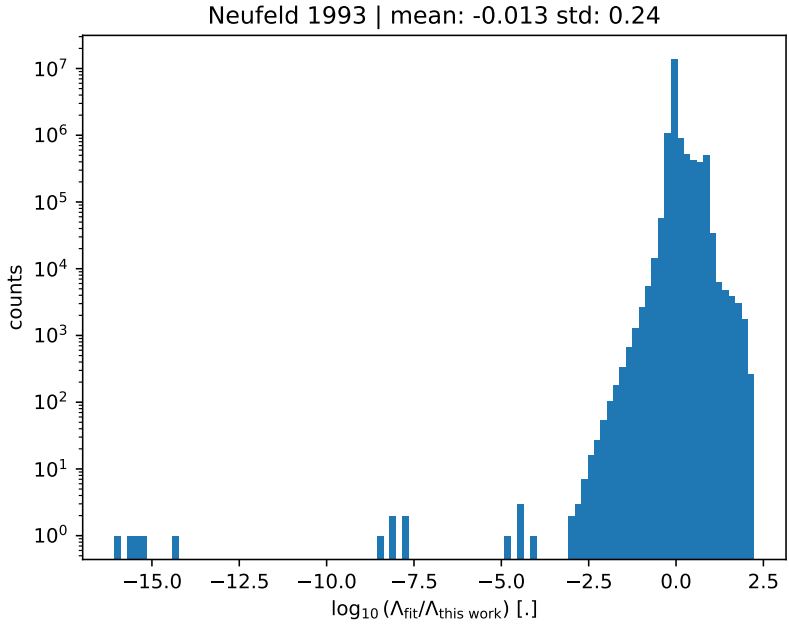


Figure 6.6: Histogram of relative differences between the reference results and the cooling fit result calculated using the Neufeld and Kaufman [1993] prescription, in log space. Note that the y-axis is also in log space to show the outliers.

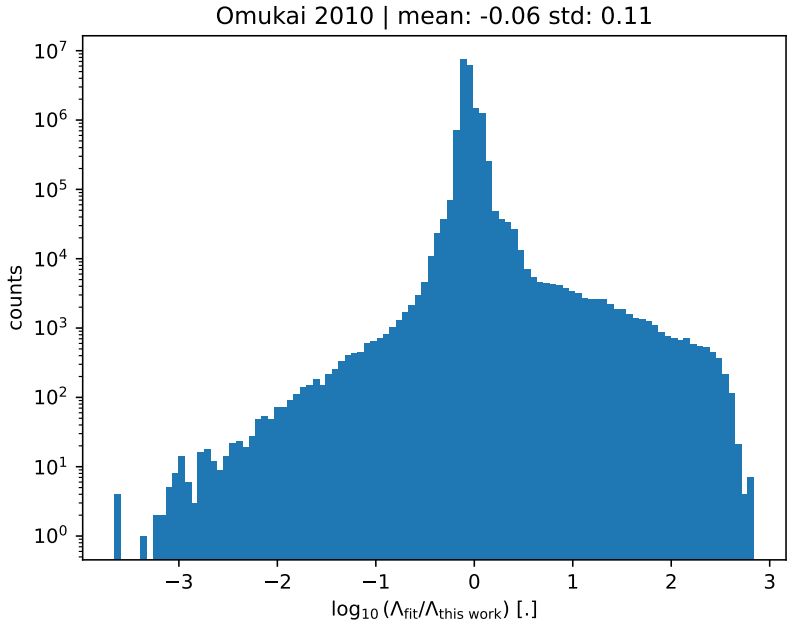


Figure 6.7: Histogram of relative differences between the reference results and the cooling fit result calculated using the Omukai et al. [2010] prescription, in log space. Note that the y-axis is also in log space to show the outliers.

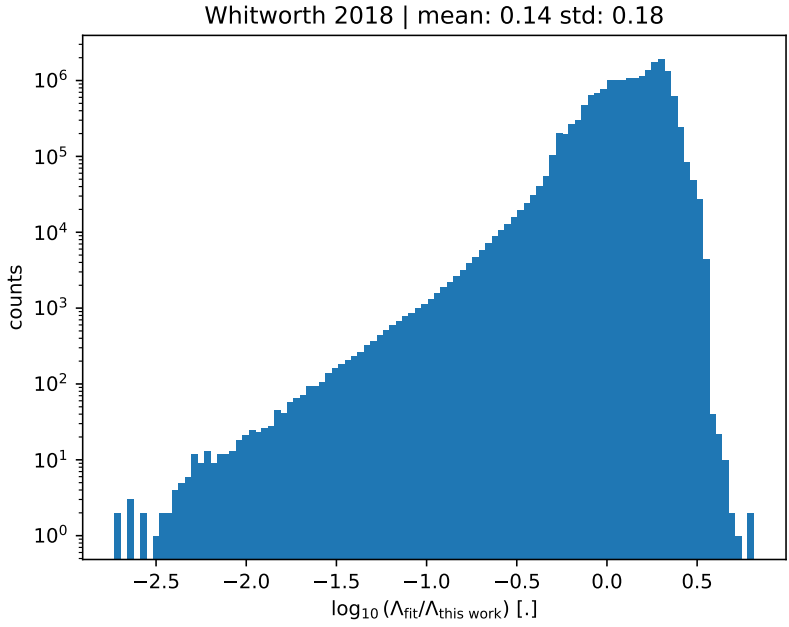


Figure 6.8: Histogram of relative differences between the reference results and the cooling fit result calculated using the Whitworth and Jaffa [2018] prescription, in log space. Note that the y-axis is also in log space to show the outliers.

number of polynomial terms per input parameter to be variable. In the fit, we use N_1 terms for the temperature T, N_2 terms for the H_2 density, N_3 terms for the abundance ratio n_{CO}/n_{H_2}, N_4 terms for $\tilde{N}(CO)$, resulting in the following formula:

$$\log_{10}(\Lambda)_{\text{fit}} = a + \sum_{i=1}^{N_1} b_i \log_{10}(T) + \sum_{i=1}^{N_2} c_i \log_{10}(n_{H_2}) \tag{6.12}$$

$$+ \sum_{i=1}^{N_3} d_i \log_{10}\left(\frac{n_{CO}}{n_{H_2}}\right) + \sum_{i=1}^{N_4} e_i \log_{10}\left(\frac{n_{CO}}{|\nabla \cdot \boldsymbol{v}|}\right).$$

In this, a is a constant term, and b_i, c_i, d_i, e_i are constants with the number of parameters per quantity N_i to be determined. The input terms for the fits are all in SI units. Note that we chose the abundance ratio n_{CO}/n_{H_2} instead of the CO number density in our fit. This is because we have modeled the CO number density to be proportional to the H_2 number density, and is therefore equal to the log H_2 number density plus a constant in log space. Therefore, also using a CO density term would result in an ill-defined fit function. The lower bound of $N = [N_1, N_2, N_3, N_4]$ we consider, is $N = [1, 1, 1, 0]$. The upper bound we consider is $N = [4, 4, 4, 4]$. We limit the order of the polynomial terms to 4, in order to limit any issues with numerical precision, as the optimal fit coefficients would significantly increase when more higher order terms are included. As the fit coefficients try to cancel each other out, small perturbations on the coefficients (by e.g. rounding), can result in large perturbations on the fit result. We therefore do not round our obtained fit coefficient values in this paper. Furthermore, in order to further reduce the values of the obtained fit coefficients, we add a regularization term for the coefficients to the least-squares fitting we use, defining the loss function $\mathcal{L}$ to be:

$$\mathcal{L} = \sum_{\text{data points}} \left(\log_{10}(\Lambda)_{\text{fit}} - \log_{10}(\Lambda)_{\text{data}}\right)^2 \tag{6.13}$$

$$+ r^2 \left(a^2 + \sum_{i=1}^{N_1} \left((i+1)b_i\right)^2 + \sum_{i=1}^{N_2} \left((i+1)c_i\right)^2 \right.$$

$$\left. + \sum_{i=1}^{N_3} \left((i+1)d_i\right)^2 + \sum_{i=1}^{N_4} \left((i+1)e_i\right)^2 \right)$$

in which the coefficient of the regularization term r, is taken to be 1. As we will later on use this loss term to fit different size data sets, we have chosen this regularization coefficient to be variable. This allows us to tweak the coefficient

such that the regularization term has a similar impact, relative to the loss on the fit error, which scales proportionally to the number of data points. We start from the most complicated configuration $N = [4, 4, 4, 4]$, and removed polynomial terms, starting from the last parameter, unless the mean fit error would change significantly, by more than 1%. However, we make an exception by still removing a term when the maximal spread of the error $\log_{10}(\Lambda_{\text{fit}}/\Lambda_{\text{this work}})$ would be reduced by more than 10%. Afterwards, we repeat this procedure for all other parameters, from (second to) last to first. The final configuration uses $N = [4, 4, 1, 2]$, with parameter values

$$
\begin{aligned}
a &= -1.21240570 \cdot 10^{1} \\
\boldsymbol{b} &= [2.72210207, -1.46925093, 5.33555446 \cdot 10^{-1}, \\
&\quad - 6.44150584 \cdot 10^{-2}] \\
\boldsymbol{c} &= [-8.47598504, 1.27675907, -6.49227392 \cdot 10^{-2}, \\
&\quad 1.06609653 \cdot 10^{-3}] \\
\boldsymbol{d} &= [9.55136991 \cdot 10^{-1}] \\
\boldsymbol{e} &= [4.04944142 \cdot 10^{-1}, -1.18855989 \cdot 10^{-2}].
\end{aligned}
\tag{6.14}
$$

The fit residual has a standard deviation of 0.039 dex.

Upon further inspection of the residual, we found that the general fit struggles with the highest density region (see Fig. 6.10). The parameter region for which $\log_{10}(n_{\text{CO}}) > 10 \log_{10}(\text{m}^{-3})$ contains about 3% of all model points, but is fitted significantly worse than the lower density regions. This result can be expected from theory, as the high density regime is optically thick, in contrast to the low density regime, and should therefore be treated differently. As there are 2 orders of magnitude less data points in the former regime compared to the latter, the former will be fitted worse. To illustrate this, we create a corner plot using the fit residuals, squared[2], shown in Fig. 6.11. This figure shows in which part of the parameter space the fit performs worst, and is not corrected for data point density, similar to the actual fitting procedure.

We will therefore fit the optically thick and thin regimes separately. To decide on the exact parameter cutoff, we separate the different parameter regions in which the fit performs worst. Based on Fig. 6.11, which shows us the loss of the

[2]Such that we mimic the L2 norm loss function used for least squares fitting.

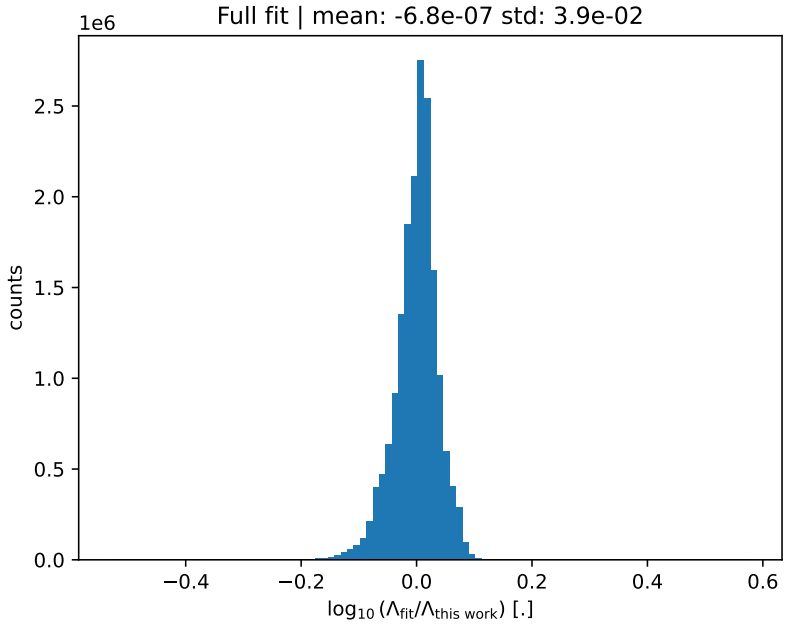


Figure 6.9: Histogram of relative differences between the reference results and the cooling fit of this paper for the entire parameter range.

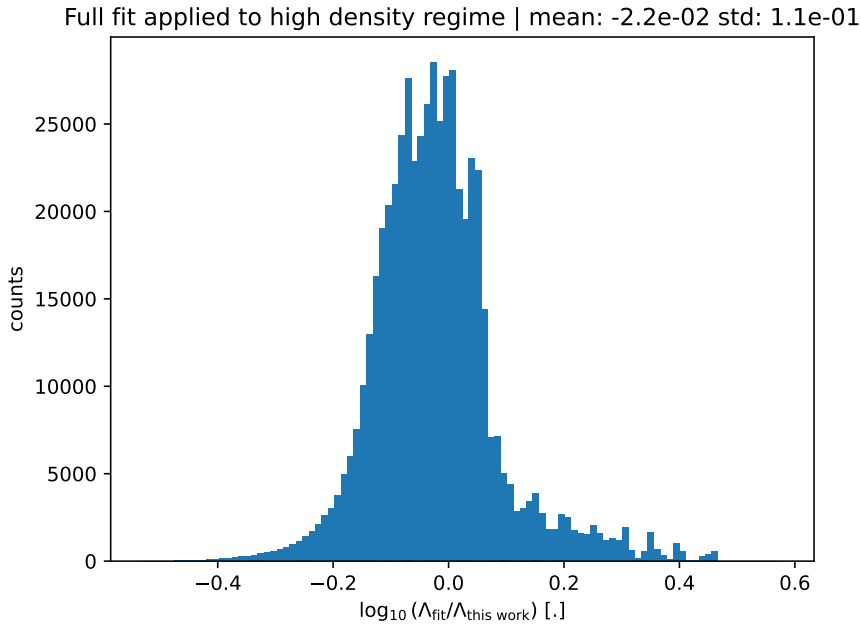


Figure 6.10: Histogram of relative differences between the reference results and the cooling fit of this paper for the entire parameter range, evaluated in the high density regime $\log_{10}(n_{\mathrm{CO}}) > 10$.

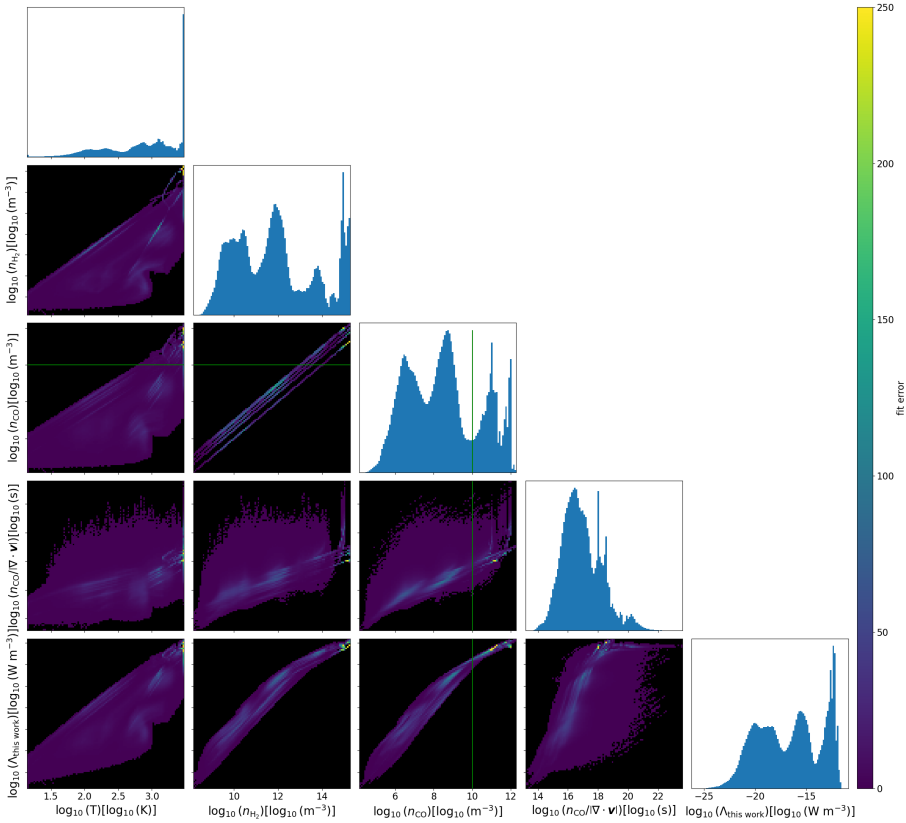


Figure 6.11: Corner plot of the fit error corresponding to the full parameter space fit of Section 6.4.3, in which we define the fit error as $(\log_{10}(\Lambda_{\text{fit}}) - \log_{10}(\Lambda_{\text{this work}}))^2$. The parameter space without data is colored black. The solid green line indicates where we will split the parameter domain for defining separate fits for the low and high density regimes.

fit without the regularization term in the different parameter regimes, we decide on using $\log_{10}(n_{\mathrm{CO}}) = 10.0\log_{10}(\mathrm{m}^{-3})$ as cutoff, as it visually separates the more difficult high density region from the rest of the parameter space. From this figure, we notice a few peculiarities. First, we see that a significant part of the fit error is found at a temperature of 3000 K, which is the result of imposing an upper bound on the temperature. Second, we notice stripes appearing in the $n_{\mathrm{H_2}}$ versus n_{CO} plot, which arise from defining the CO density as a fraction of the $\mathrm{H_2}$ density (see Section 6.4.1).

We find the following fit parameter values for the low density regime $\log_{10}(n_{\mathrm{CO}}) < 10.0\log_{10}(\mathrm{m}^{-3})$

$$
\begin{aligned}
N &= [4, 3, 1, 0] \\
a &= -2.12040965 \cdot 10^{1} \\
\boldsymbol{b} &= [3.01783489, -1.59256830, 5.51104820 \cdot 10^{-1}, \\
&\quad - 6.45904646 \cdot 10^{-2}] \\
\boldsymbol{c} &= [-3.69490411, 5.83898651 \cdot 10^{-1}, -2.04180720 \cdot 10^{-2}] \\
\boldsymbol{d} &= [9.84099436 \cdot 10^{-1}].
\end{aligned}
\tag{6.15}
$$

The low density fit has residuals with a standard deviation of 0.034 dex in the low density regime (see Fig. 6.12). As the high density regime contains roughly 2 orders of magnitude less data points, we changed the regularization coefficient r from 1 to 0.1 in Eq. 6.13, such that the regularization term has roughly same impact on the loss function, compared to fitting using all data. Using this, we obtain the following fit parameter values for the high density regime $(\log_{10}(n_{\mathrm{CO}}) > 10.0\log_{10}(\mathrm{m}^{-3}))$

$$
\begin{aligned}
N &= [2, 4, 1, 3] \\
a &= -6.15379992 \cdot 10^{1} \\
\boldsymbol{b} &= [4.53170350, -5.48993030 \cdot 10^{-1}] \\
\boldsymbol{c} &= [-1.24741994 \cdot 10^{1}, 2.09789393, -1.28957948 \cdot 10^{-1}, \\
&\quad 2.77218176 \cdot 10^{-3}] \\
\boldsymbol{d} &= [7.86957705 \cdot 10^{-1}] \\
\boldsymbol{e} &= [8.21468631, -4.17788945 \cdot 10^{-1}, 6.94393690 \cdot 10^{-3}].
\end{aligned}
\tag{6.16}
$$

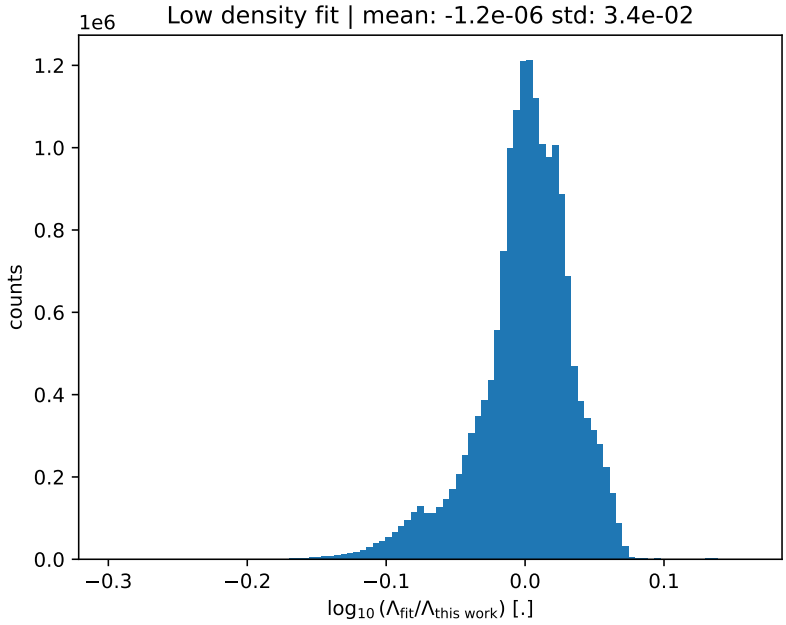


Figure 6.12: Histogram of relative differences between the reference results and the cooling fit of this paper for the low density regime.

The high density fit has residuals with a standard deviation of 0.053 dex in the high density regime (see Fig. 6.13). From these fits, we notice that in general, the abundance ratio $n_{\mathrm{CO}}/n_{\mathrm{H_2}}$ can be described with a linear term. Confirming Whitworth and Jaffa [2018], we find that we do not need to include the $n_{\mathrm{CO}}/\left|\nabla \cdot \boldsymbol{v}\right|$ term for the low density regime. Furthermore, we notice that the temperature dependence in the high density regime is less complicated compared to the low density regime. We suspect this is due to having a smaller parameter range for the temperature in the high density regime, as the data points in that regime all have temperatures above 635K. Finally, we notice that the higher density regime is harder to fit. Two different regions in our models have high densities: the region near the star, where the wind is launched, and the region near the companion object, where material is accumulated. We therefore suspect that we map two different physical regimes to the same parameter space, resulting in a worse fit.

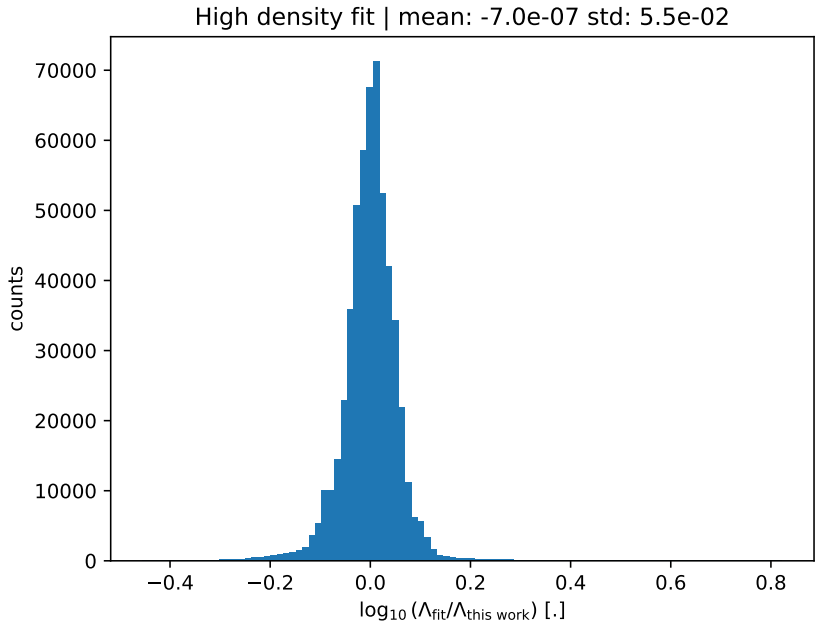


Figure 6.13: Histogram of relative differences between the reference results and the cooling fit of this paper for the high density regime.

Estimating the fit error 6.4.4

When using least-squares minimization, one obtains by default the optimal fit parameters $\boldsymbol{w}$, but one can also obtain a covariance matrix $\Sigma_{\boldsymbol{w}}$ for the obtained fit parameters, which can be used for estimating the uncertainty on the fit parameters. In this paper, we use the function `scipy.optimize.curve_fit` from SCIPY [Virtanen et al., 2020] to compute both simultaneously. To derive uncertainties for the fit, we use

$$\Sigma_f = X \Sigma_{\boldsymbol{w}} X^T, \tag{6.17}$$

in which Σ_f is the covariance matrices for the fit function value. In this, X denotes the $(N_{evaluations} \times N_{\text{fit parameters}})$ design matrix, in which each row corresponds to the evaluation of a single data point at all individual polynomial terms, such that the fit result $f(x_i)$ of a single data point x_i can be evaluated using matrix multiplication, i.e.

$$f(x_i) = X_i \boldsymbol{w}, \tag{6.18}$$

in which X_i denotes the i-th row of the matrix X. We can now estimate the residuals by calibrating the fit uncertainty to the variance of the fit residual, using

$$\Sigma_r = c^2 \Sigma_f \tag{6.19}$$

in which c is given by

$$c = \frac{\sigma}{\sqrt{E(\Sigma_{f,ii})}} \tag{6.20}$$

and σ is the standard deviation on the fit residual and $\sqrt{E(\Sigma_{f,ii})}$ is the square root of the average of the diagonal elements of Σ_f, evaluated on all data points used for the fit[3]. To shorten notation, we now define σ_{fit} to be the square root of the diagonal of Σ_r. Both $\Sigma_{\boldsymbol{w}}$ and c can be found in the online supplementary data.

To check whether this calibrated fit uncertainty agrees with actual residual, we calculate this estimate for our first fit in this paper (Eq. 6.14), for all data points. We find that 67% of the fit residuals lie between $\pm 1\sigma_{\text{fit}}$ and 95% lie between $\pm 2\sigma_{\text{fit}}$ (see Fig. 6.14). From Fig. 6.15, we notice that in the intermediate density regime, $8 < \log_{10}(n_{\text{CO}}) < 10$, the calibrated fit error σ_{fit} seems to

[3]If one is not particularly interested in the covariance, but just aims to use Σ_r for estimating the pointwise error, one does not need to compute the off-diagonal elements of Σ_f.

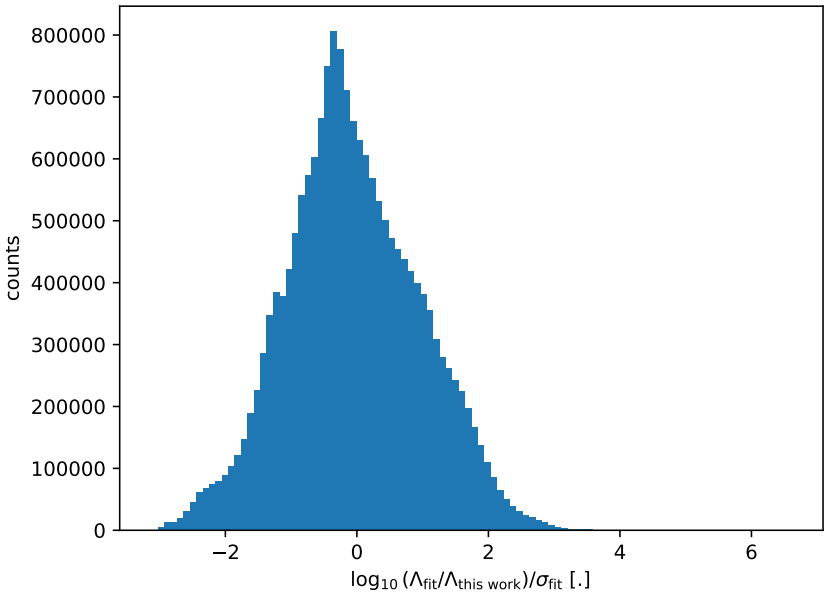


Figure 6.14: Histogram of the fit residual, divided by the calibrated fit error, calculated using the fit based on the entire parameter range.

underestimate the residual, as the residual shows a two-peaked distribution in this part of the parameter space. Conversely, in the very low density regime, $\log_{10}(n_{\mathrm{CO}}) < 5$, σ_{fit} overestimates the residual. We conclude that the calibrated fit error σ_{fit} can be used as proxy for (the width of the distribution of) the fit residual, however, it might not be fully accurate when looking at specific regions of the parameter space.

Model perturbations 6.5

In Section 6.4.4, we have computed uncertainties on the accuracy of the obtained fit. However, the physics included in the hydrodynamics simulations which we used for calculating the CO cooling rates, are not entirely complete, missing for example the effect of CO cooling itself. To estimate how much the cooling rates may change with a slightly different physics description, we study the influence of using slightly perturbed parameter values on the cooling rate. For

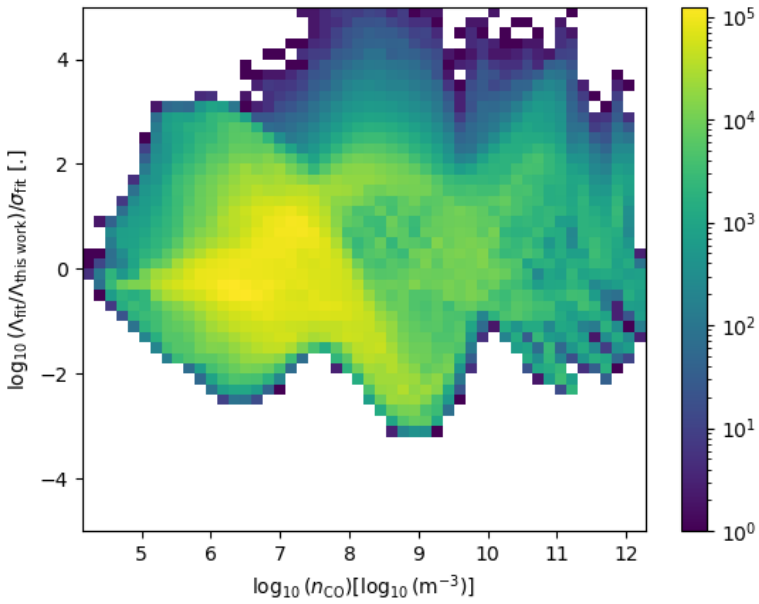


Figure 6.15: 2D histogram of the fit residual, divided by the calibrated fit error, plotted against the CO density, using a logarithmic scale for the color bar.

this we use MAGRITTETORCH[4], a port of MAGRITTE [see e.g. De Ceuster, 2022] to PYTORCH Paszke et al. [2019] [5]. We have built this code on top of a machine learning library to ensure that the resulting models are fully differentiable. This allows us to track a gradient throughout the non-linear radiative transfer calculations. Starting from a regular radiative transfer model, we can use forward gradients to add a perturbation term on the model input, which is propagated throughout the entire NLTE computation, eventually giving us a gradient on the computed cooling rates.

To get a global idea of how the cooling rates change, depending on perturbations of the model input, we rerun the last snapshot of each AGB binary model used in this paper, for a CO/H_2 abundance ratio of $5 \cdot 10^{-4}$. We successively apply perturbations δ of 10% on the temperature, CO density and velocity. Note that the exact percentage of the perturbations does not matter, as we obtain a linear approximation for perturbation through evaluating the gradient, and we will rescale the obtained perturbation on the cooling rate by the given percentage on the input gradient. In this way, we obtain the change in (log) cooling rate per (log) input parameter. For example, when applying this for the input temperature, we obtain

$$\alpha = \frac{\Lambda(T+\delta T)}{\Lambda(T)/\delta} = \frac{\Lambda(T+\delta T)}{\Lambda(T)} \frac{T}{\delta T} \simeq \frac{\partial \log(\Lambda)}{\partial \log(T)}, \tag{6.21}$$

in which α is the computed value. By integrating and exponentiating both sides, we obtain a (local) scaling relation for the cooling rate

$$\Lambda \sim T^{\alpha}. \tag{6.22}$$

A similar argument can be made to obtain scaling relations for the other input parameters.

We start by first exploring the impact of the input parameters on the low density regime, $\log_{10}(n_{CO}) < 10$. In Fig. 6.16, we see that an increase in temperature in general leads to a (super)linear increase in cooling. During the analysis, however, we have noticed some outliers. Data points with temperatures above 3000K are treated inconsistently during the radiative transfer calculation, due to collisional data being limited to $\leq$ 3000K. In our code, temperatures above 3000K still impact the line width, but no longer influence the collision rates C_{ij}. Therefore, we have removed these data points from our analysis for this section. Next, we plot the impact of a perturbation for the CO density on the cooling rate in Fig. 6.17. We find that the cooling rate depends linearly on the density, which is as we expect for the optically thin regime. Finally, in Fig. 6.18, we

[4] Available at `https://github.com/Magritte-code/Magritte-torch`

[5] MAGRITTETORCH is currently based on version 0.7.2 of MAGRITTE.

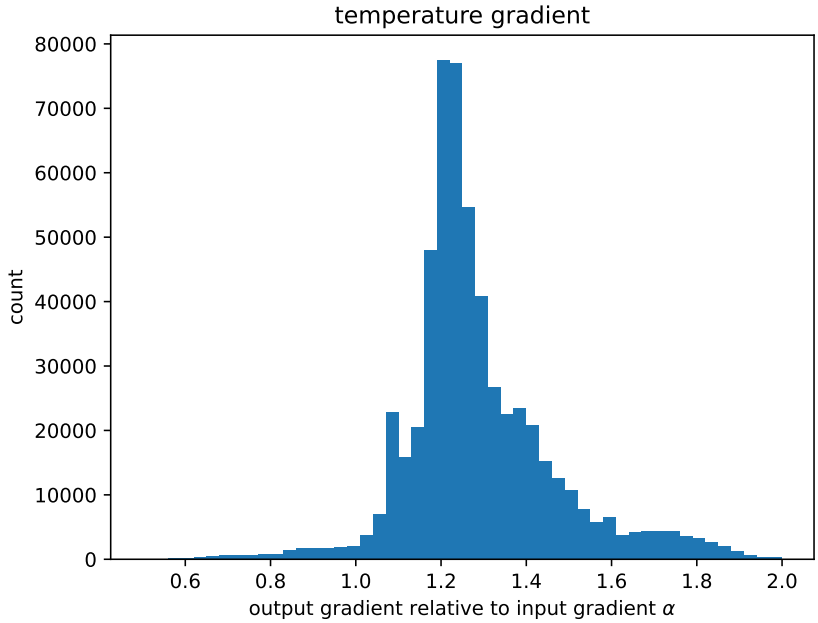


Figure 6.16: Histogram of the impact of a temperature perturbation on the cooling rate in the low density regime.

find that the cooling rate does almost not depend on the velocity field in this regime, which can be expected based on the argumentation in Section 6.3.

For the high density regime, $\log_{10}(n_{\mathrm{CO}}) > 10$, we plot the cooling rate dependence on the same input parameters in Figs. 6.19, 6.20 and 6.21. Here, we find similar results for the cooling rate scaling as in the low density regime, but with a weaker dependence on the input parameter, except for the input velocity field. Increasing the velocity does increase the cooling rate. We note that the scaling relations found in this section can be derived from analytical approximations [see e.g. Whitworth and Jaffa, 2018], but we hereby confirm these to be valid for radiative transfer simulations in the complex AGB environment. However, the exact power laws of the scaling laws might not entirely match. In particular, Whitworth and Jaffa [2018] predicts for the temperature scaling in the high density regime that $\Lambda \sim T^4$, and for the velocity gradient scaling $\Lambda \sim |\nabla \cdot \boldsymbol{v}|$. Our results do not come close to the aforementioned temperature scaling, and do not reach the velocity gradient scaling for any significant amount of model points. We must note however, that these discrepancies might arise

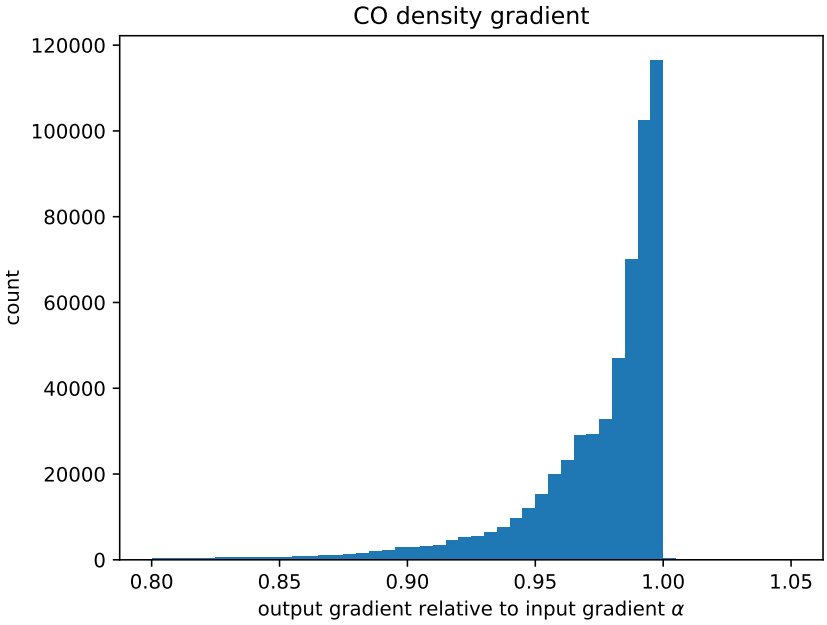


Figure 6.17: Histogram of the impact of a CO density perturbation on the cooling rate in the low density regime.

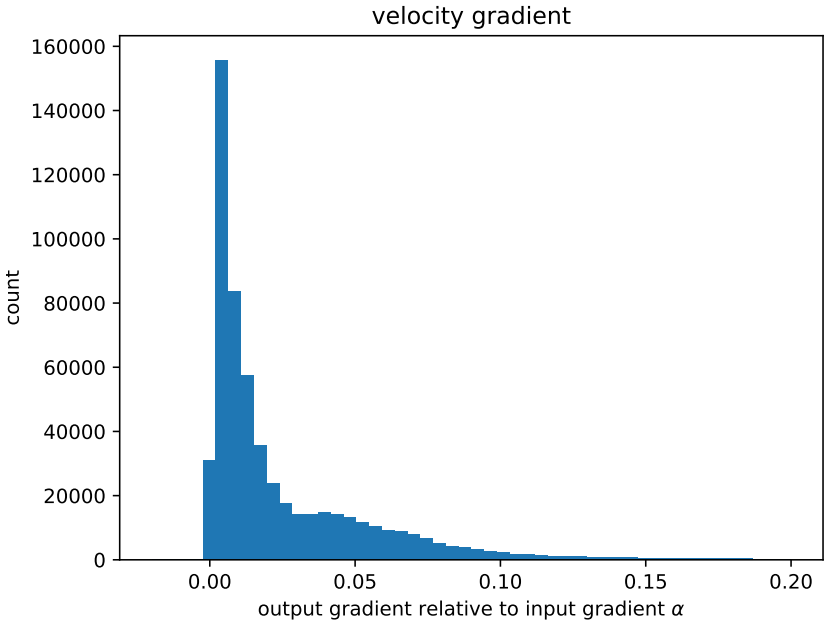


Figure 6.18: Histogram of the impact of a perturbation of the velocity field on the cooling rate in the low density regime.

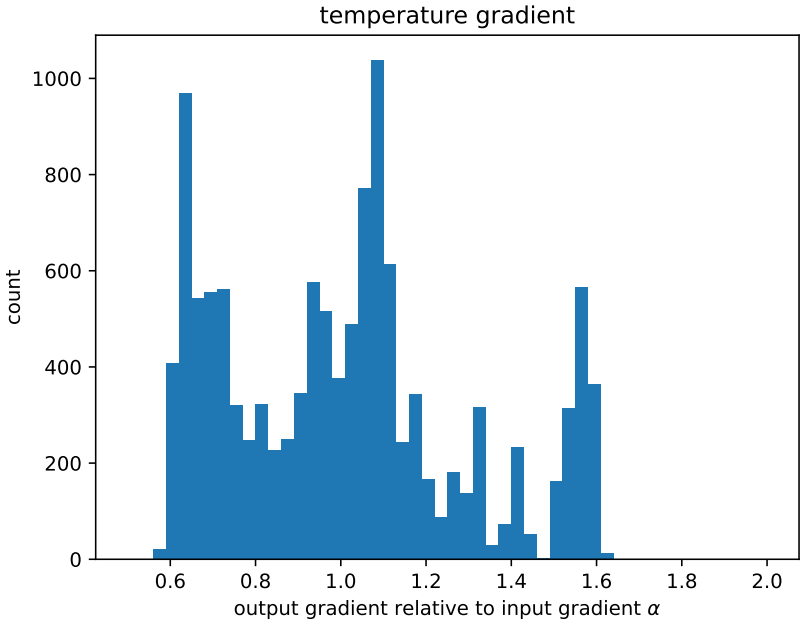


Figure 6.19: Histogram of the impact of a temperature perturbation on the cooling rate in the high density regime.

from be due to our high density model points not being dense enough, as the locally emitted radiation might not yet be completely limited by the velocity field.

Limitations 6.6

The cooling fits we propose in this paper come with some limitations to the regime they are applicable to. Aside from the parameter limits described in Table 6.2, we list some shortcomings of our radiative transfer modeling and estimate their impact on the obtained rotational CO line cooling rates.

Our radiative transfer models in this paper do not include the intensity contribution from the star. We have done this to allow our results to be comparable with literature prescriptions for ISM-like conditions in Section 6.3. As the regions just outside the star are approximately in LTE conditions, and have temperatures similar to the effective temperature of the star, we do not

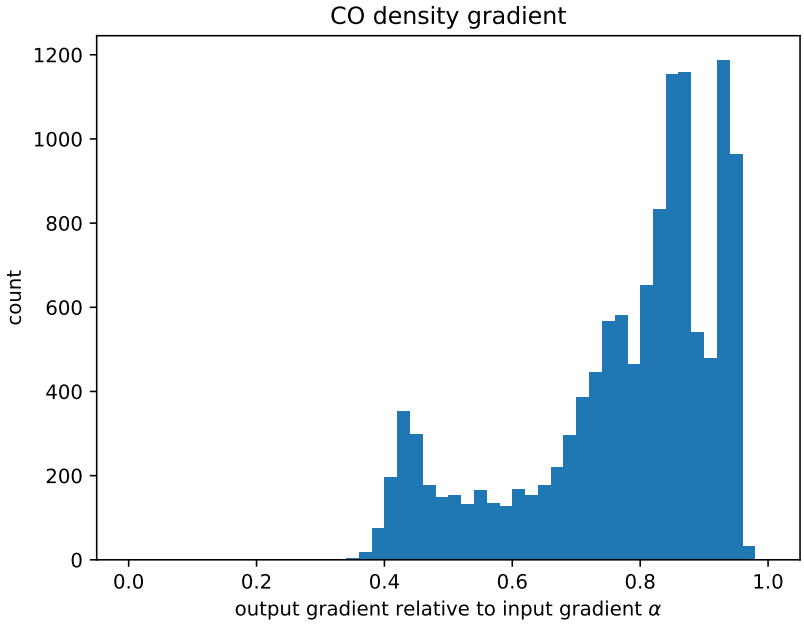


Figure 6.20: Histogram of the impact of a CO density perturbation on the cooling rate in the high density regime.

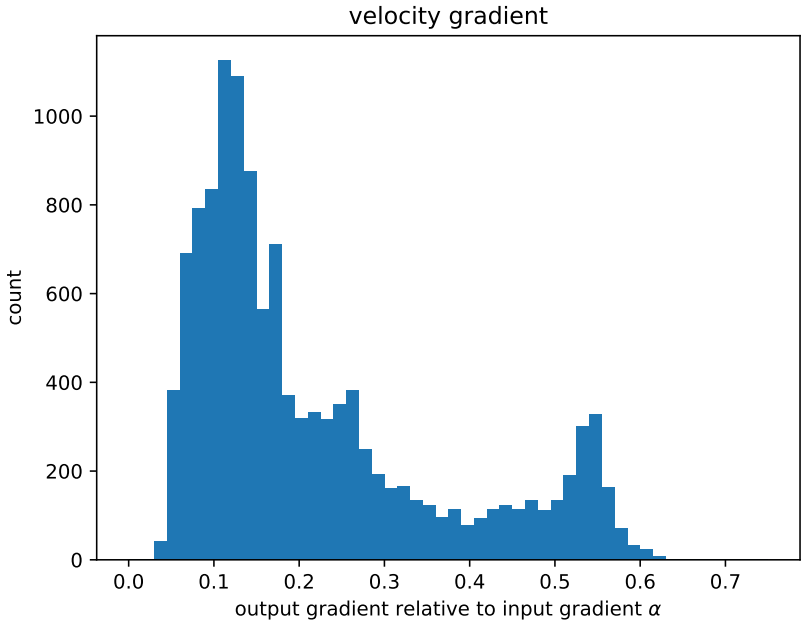


Figure 6.21: Histogram of the impact of a perturbation of the velocity field on the cooling rate in the high density regime.

think this omission has a significant impact on the radiation field in the models. However, an extra intensity contribution from the star might cause the closest regions to heat instead of cool. We hereby note that we not encountered any net radiative heating in any of our MAGRITTE models.

We have simulated NLTE line radiative transfer using the first 41 rotational levels of CO (v=0) and corresponding collisional rates up to 3000K [from Yang et al., 2010]. We have therefore implicitly imposed an upper bound on the model temperature of 3000K, when simulating NLTE radiative transfer. Correspondingly, we also imposed an upper bound of 3000K on the model temperatures used for the fitting functions. The limited amount of rotational lines might affect the cooling rates in the hottest regions, as the emitted light could be distributed over more lines, possibly increasing the radiative cooling rate.

Conclusion 6.7

In this paper, we provide CO cooling rates for the AGB environment. For this, we create various models of the AGB binary environment in Section 6.4, to which we apply NLTE line radiative transfer. We first compare the computed cooling rates with some cooling rate prescriptions for the ISM, which we describe in Section 6.3. It turns out that all of these prescriptions are based on different parameter ranges than we have in our AGB models. We found that the cooling rate prescription from Whitworth and Jaffa [2018] works best, having the least amount of outliers. This prescription slightly overestimates the cooling rate by 0.14 dex on average, and has a standard deviation on the error of 0.18 dex. To improve upon this for the AGB parameter regime, we propose a polynomial fit for the cooling rate in Section 6.4.3. We find that using a single fit for the entire parameter range, results in a worse fit in the high density regime, compared to the low density regime. Therefore, we proposes different fits for the low and high density regimes, with a cutoff at $n_{\mathrm{CO}} = 10^{10}\mathrm{m}^{-3}$. In the low density regime, the standard deviation for the error is 0.034 dex, while the corresponding fit for the high density regime obtains a standard deviation of 0.053 dex. We suspect this higher error in the high density regime to arise from the more complicated behavior of the cooling rate in this regime. In high optical depths, the velocity field can limit how much radiation can escape, thereby limiting the cooling rate. The cooling rates obtained in this paper valid are for the AGB outflow, with temperatures limited to < 3000 K, H_2 densities between $1.5 \cdot 10^8$ m^{-3} and $1.9 \cdot 10^{15}$ m^{-3}, CO/H_2 ratios between $10^{-4}, 10^{-3}$ and $\tilde{N}$ between $1.5 \cdot 10^{13}$ and $4.2 \cdot 10^{23}$ s m^{-3}. Finally, we numerically verified in Section 6.5 some analytic scaling relations for the cooling rate, both in the low and high density regimes in the AGB environment. In particular, we confirm

that the velocity gradient $|\nabla \cdot \boldsymbol{v}|$ has little impact on the cooling rate in the low density regime, but matters more in the higher density regime.

Acknowledgments

T. Ceulemans has been supported by the Research Foundation - Flanders (FWO), grant 1166724N. F. De Ceuster has been supported by the FWO, grants I002123N and 1253223N. O. Vermeulen has been supported by the FWO, grant 1173025N. L. Decin has been supported by the FWO, grants G099720N, G0B3823N and the KU Leuven methusalem grant METH/24/012. The authors express their thanks to J. Malfait for providing a PHANTOM model which has been used in this paper.

Data Availability

MAGRITTE is an open-source software library available at `https://github.com/Magritte-code/Magritte`. MAGRITTETORCH is an open-source software library available at `https://github.com/Magritte-code/Magritte-torch`. The models in this paper are available upon reasonable request to the authors.

Applying the full fit to single star models 6.A

In the main text, we focused on creating a CO cooling rate fit for hydrodynamics models of AGB stars with a companion. Evidently, AGB stars can also exist without a stellar mass companion object. Therefore, we verify in this appendix whether our fit prescriptions also work for single star models.

In Section 6.4.1, we have explained how we created the radiative transfer models for all AGB models in this paper, including the single star models. Given that the parameter space of data points within the single star model is a subset of the parameter space of data points for the binary models (see Fig. 6.22), we expect that our fits of Section 6.4.3 will be adequate for the single star models. In Fig. 6.23, we confirm this by applying both the low and high density fits (in their relevant parameter ranges) to the single star data, obtaining a mean relative difference for the log cooling rate $\log_{10}(\Lambda_{\text{fit}}/\Lambda_{\text{this work}})$ of $-9.9 \cdot 10^{-3}$ dex, with standard deviation of $2.9 \cdot 10^{-2}$ dex.

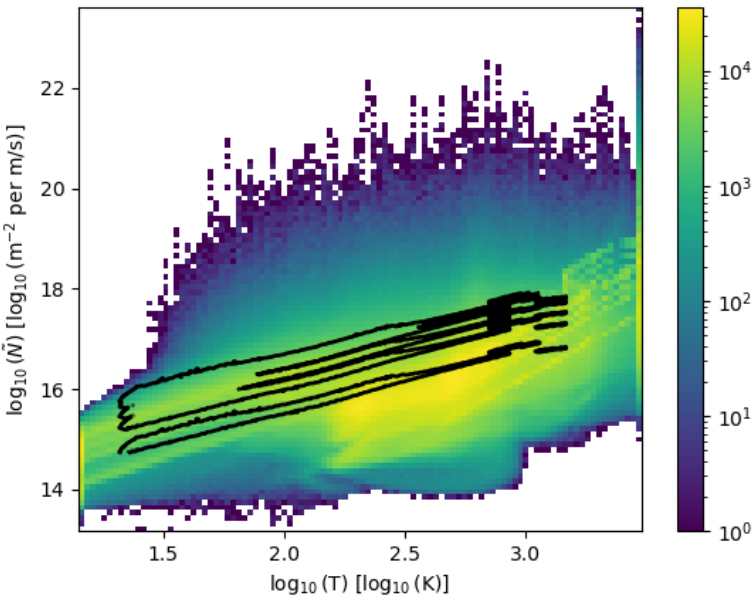


Figure 6.22: Distribution of our AGB binary model points. The black contours denote the extent of the parameter range for the single star model points.

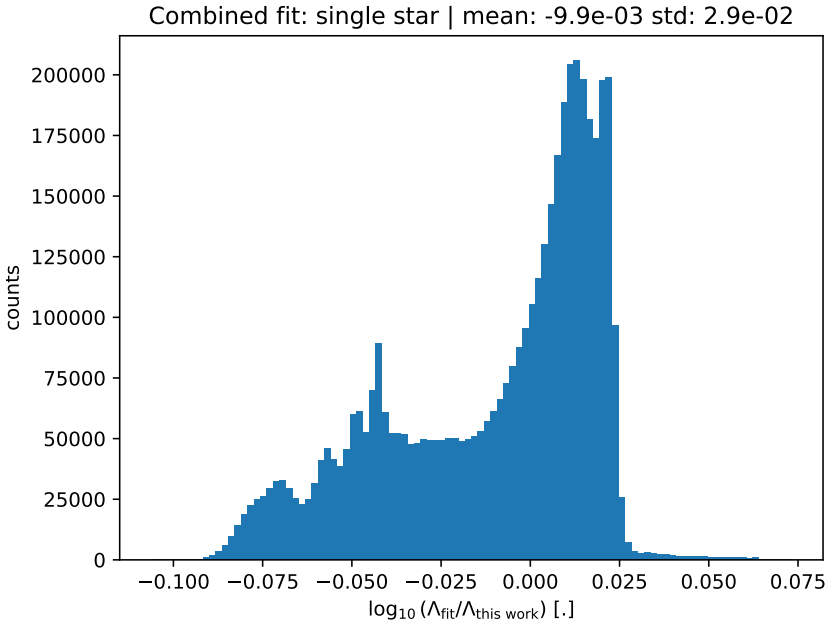


Figure 6.23: Histogram of relative differences between the reference results for the single star data and the combined cooling fits for the low and high density regimes.

CHAPTER 7

Future work

In this chapter, we outline a few possibilities to improve upon this work. Some future work pertaining to particular codes, has been mentioned in the corresponding chapters. We will mention what can still be done.

Moment method in line radiative transfer 7.0.1

In radiative transfer, the moment method relies integrating different moments of the radiative transfer (see Eq. 2.2), supplemented by a closure relation. The method has better computational scaling, as it only solves the equation for the integrated solid angle, instead of needing an angular discretization. This method is currently used in continuum radiative transfer. One can also apply it in line radiative transfer, if one approximates opacities and emissivities using mean values [see e.g. Moens et al., 2022]. However, it has not yet been used for detailed line radiative transfer, in which all lines are treated individually. This due to a mismatch which frequencies get combined in the angular averaging, due to influence of Doppler shifts. For a simple example, we construct the zero moment of the radiative transfer equation, assuming a single line,

$$\int_\Omega \hat{\boldsymbol{n}} \cdot \nabla I(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu) d\hat{\boldsymbol{n}} = \int_\Omega \eta(\boldsymbol{x}, \nu) - \chi(\boldsymbol{x}, \nu) I(\hat{\boldsymbol{n}}, \boldsymbol{x}, \nu) d\hat{\boldsymbol{n}}. \tag{7.1}$$

In theory, we can write down the moment equations for every position independently. However, each one is defined with a different local velocity, so it would be more convenient to integrate out the frequency dependency. However, the opacity term on the right-hand side cannot easily be decoupled from the intensity, as they are both frequency-dependent.

It is theoretically possible to use a co-moving formulation of the radiative transfer equation (see Eq. 4.5) as a start for the moment equation to not have to integrate out the frequency. The main challenge would be to figure out how to implement correct frequency boundary conditions everywhere.

Comoving solver in 1D 7.0.2

In chapter 4, we implement a numerically stable comoving solver to be used in 3D radiative transfer. It reduces the scaling power from $\sim O(N_{\text{POINTS}}^{4/3})$ to $O(N_{\text{points}})$. In 1D radiative transfer, the computational gains are even higher, reducing scaling from $O(N_{\text{points}}^2)$ to $O(N_{\text{points}})$. Several papers [see e.g. Baron and Hauschildt, 2004, Hauschildt and Baron, 2004] have already described the comoving method in a non-monotonic setting[1], but did not address any stability concerns. However, the stabilization method in the thesis is still new, and has potential to be applied to an actual 1D radiative transfer code.

Therefore, we can recommend implementing this method if one wants to do model retrieval on spherically symmetric objects with non-monotonic velocity fields. In currently available comoving frame solvers for 1D models, one either assumes a monotonic velocity field or risks having numerical instabilities.

Cooling rates for other molecules 7.0.3

In Chapter 6, we calculate a radiative cooling rate prescription for CO. We can repeat the calculations for other molecules using the same pipeline, with the caveat we do not treat population inversions, which can cause maser effects. Note that other molecules will have more complex density structure than CO. The main task will thus be to calculate the density profile of the molecule of interest. However, similar to radiative transfer, chemistry is also computationally expensive compared to hydrodynamics simulations. One might therefore need to use a chemistry emulator, such as MACE [Maes et al., 2024], in order to efficiently compute the abundance profile.

Applying frequency matching to hydrodynamics 7.0.4

In Chapter 4, we have introduced frequency matching in order to alleviate stability issues when using the co-moving frame formulation of the radiative transfer equation (see Eq. 4.1). The general idea behind frequency matching, is to use an adaptive stencil for the numerical discretization in order to avoid

[1] Although they fail to mention how they implemented their boundary conditions.

stability issues. This could be used in time-stepping methods to allow us to simplify the numerical discretization[2] and allow for larger time steps, at the cost of significantly more complex boundary conditions and stencil calculations.

We think that this method could have applications in numerical problems, in which time stepping is limited by CFL-like stability criteria. We estimate that most effort will need to be spent in optimizing the implementation of the adaptive stencil and the boundary conditions.

[2]For hydrodynamics, for example flux limiters have been used to suppress numerical instabilities.

CHAPTER 8

Conclusion

The ISM is the environment in which stars and planets are formed. Therefore, to understand our origins, we must trace the origin of the material in the ISM. It turns out that a significant amount of the dust and gas in the ISM originates from AGB stars [Tielens, 2005]. These AGB stars lose most of their mass in stellar winds, driven by radiation pressure on dust. In these outflows, complex non-equilibrium chemistry takes place. To simulate this, we require input from hydrodynamics and radiative transfer. Many observations for AGB stars show complex structures, such as arcs and spiral-shape outflows, which hint at the presence of a companion. Therefore, we can no longer treat our simulations in 1D.

The topic of this thesis is simulating radiative transfer, as it allows us to both simulate the appearance of our AGB models and allows to to include additional physics. We are in particular interested in the emission and absorption of light by line transitions. These narrow features in frequency space originate from transitions between discretized energy levels, and have a unique frequency spectrum for each species. Thus from an observed spectrum, we can derive which chemical species are present, based on the presence of these lines.

In order to simulate NLTE line radiative transfer, we need to take into account the mutual dependence of the radiation field on the energy state of the medium and vice versa. However, these simulations are computationally expensive, especially in 3D. Therefore, I focused most of my efforts throughout this PhD on improving the computational efficiency of NLTE line radiative transfer, further developing the open-source NLTE line radiative transfer library Magritte. In Chapter 3, we have shown how to use Magritte, and detail what computational improvements we have made to simulating line radiative transfer.

Further trying to optimize radiative transfer simulations in Chapter 4, we

explored the co-moving frame description of the radiative transfer equation, as it provides better computational scaling, compared to the Feautrier method we use. However, this co-moving method suffers from numerical instabilities [Hauschildt and Baron, 2004, Sampoorna and Nagendra, 2016]. To alleviate the numerical issues, we used a novel discretization of this equation, using what we call frequency matching. This involves using an adaptive stencil for the discretization, minimizing the (static frame) frequency differences. Applying this method to calculate the radiation field of an AGB model with 885501 points, we found a good agreement with the results computed using the Feautrier method, and obtained a speedup of up to 5 times, depending on the exact details of the implementation we used.

Newer compute architectures are optimized for computing using vector operations. Therefore, in Chapter 5, we modified the computation algorithms of our code MAGRITTE to perform operations on large arrays of data. To do this, we created a port of MAGRITTE using the machine learning library PYTORCH. We applied both the new code, called MAGRITTETORCH, and MAGRITTE to simulate radiative transfer on an AGB model. We found that using MAGRITTETORCH on GPU is a few times faster than MAGRITTE on CPU, while providing similar results. Furthermore, we can use MAGRITTETORCH to provide uncertainties on the calculated results, by exploiting the fact that we can compute gradients on the model output with respect to the model input, given that our models are built on top of machine learning libraries.

Finally, in Chapter 6, we computed rotational CO line cooling rates for the AGB environment, which provide an additional cooling term, applicable at relatively low temperatures ($\geqslant 10$ K). Given that detailed line radiative transfer is still too computationally expensive to include in our AGB hydrodynamics simulations, we resorted to finding a prescription for the cooling rate. We first checked whether already existing cooling rate prescriptions for the ISM apply to the AGB outflow, by computing the cooling rate on a variety of AGB binary simulations. However, these prescriptions are not tailored to the parameter regime of the AGB outflow, and proved to be not sufficiently accurate. We therefore propose a new rotational CO cooling prescription, tailored to the AGB environment.

Using the tools developed in this thesis, we can now better compare our AGB models to observations, and can include additional physics to our AGB simulations. This brings us one step closer towards a fully self-consistent model for the AGB environment, which can augment our understanding of the origin of the chemical composition of the ISM, our solar system, the universe, and perhaps, the origin of life.

Bibliography

Conny Aerts. STELLAR STRUCTURE & EVOLUTION course notes. 2021.

M. Agúndez, J. I. Martínez, P. L. de Andres, J. Cernicharo, and J. A. Martín-Gago. Chemical equilibrium in AGB atmospheres: successes, failures, and prospects for small molecules, clusters, and condensates. Astronomy and Astrophysics, 637:A59, May 2020. ISSN 0004-6361. doi: 10.1051/0004-6361/202037496. URL https://ui.adsabs.harvard.edu/abs/2020A&A...637A..59A. ADS Bibcode: 2020A&A...637A..59A.

Sean M. Andrews, Jane Huang, Laura M. Pérez, Andrea Isella, Cornelis P. Dullemond, Nicolás T. Kurtovic, Viviana V. Guzmán, John M. Carpenter, David J. Wilner, Shangjia Zhang, Zhaohuan Zhu, Tilman Birnstiel, Xue-Ning Bai, Myriam Benisty, A. Meredith Hughes, Karin I. Öberg, and Luca Ricci. The Disk Substructures at High Angular Resolution Project (DSHARP). I. Motivation, Sample, Calibration, and Overview. The Astrophysical Journal Letters, 869(2):L41, December 2018. ISSN 2041-8205, 2041-8213. doi: 10.3847/2041-8213/aaf741. URL https://iopscience.iop.org/article/10.3847/2041-8213/aaf741.

Almudena Arcones and Friedrich-Karl Thielemann. Origin of the elements. The Astronomy and Astrophysics Review, 31(1):1, December 2022. ISSN 1432-0754. doi: 10.1007/s00159-022-00146-x. URL https://doi.org/10.1007/s00159-022-00146-x.

Astropy Collaboration, Adrian M. Price-Whelan, Pey Lian Lim, Nicholas Earl, Nathaniel Starkman, Larry Bradley, David L. Shupe, Aarya A. Patil, Lia Corrales, C. E. Brasseur, Maximilian Nöthe, Axel Donath, Erik Tollerud, Brett M. Morris, Adam Ginsburg, Eero Vaher, Benjamin A. Weaver, James Tocknell, William Jamieson, Marten H. van Kerkwijk, Thomas P. Robitaille, Bruce Merry, Matteo Bachetti, H. Moritz Günther, Thomas L. Aldcroft, Jaime A. Alvarado-Montes, Anne M. Archibald, Attila Bódi, Shreyas Bapat, Geert Barentsen, Juanjo Bazán, Manish Biswas, Médéric Boquien, D. J. Burke, Daria Cara, Mihai Cara, Kyle E. Conroy, Simon Conseil, Matthew W. Craig, Robert M. Cross, Kelle L. Cruz, Francesco D'Eugenio, Nadia Dencheva, Hadrien A. R. Devillepoix, Jörg P. Dietrich, Arthur Davis Eigenbrot, Thomas Erben, Leonardo Ferreira, Daniel Foreman-Mackey,

Ryan Fox, Nabil Freij, Suyog Garg, Robel Geda, Lauren Glattly, Yash Gondhalekar, Karl D. Gordon, David Grant, Perry Greenfield, Austen M. Groener, Steve Guest, Sebastian Gurovich, Rasmus Handberg, Akeem Hart, Zac Hatfield-Dodds, Derek Homeier, Griffin Hosseinzadeh, Tim Jenness, Craig K. Jones, Prajwel Joseph, J. Bryce Kalmbach, Emir Karamehmetoglu, Mikołaj Kałuszyński, Michael S. P. Kelley, Nicholas Kern, Wolfgang E. Kerzendorf, Eric W. Koch, Shankar Kulumani, Antony Lee, Chun Ly, Zhiyuan Ma, Conor MacBride, Jakob M. Maljaars, Demitri Muna, N. A. Murphy, Henrik Norman, Richard O'Steen, Kyle A. Oman, Camilla Pacifici, Sergio Pascual, J. Pascual-Granado, Rohit R. Patil, Gabriel I. Perren, Timothy E. Pickering, Tanuj Rastogi, Benjamin R. Roulston, Daniel F. Ryan, Eli S. Rykoff, Jose Sabater, Parikshit Sakurikar, Jesús Salgado, Aniket Sanghi, Nicholas Saunders, Volodymyr Savchenko, Ludwig Schwardt, Michael Seifert-Eckert, Albert Y. Shih, Anany Shrey Jain, Gyanendra Shukla, Jonathan Sick, Chris Simpson, Sudheesh Singanamalla, Leo P. Singer, Jaladh Singhal, Manodeep Sinha, Brigitta M. Sipőcz, Lee R. Spitler, David Stansby, Ole Streicher, Jani Šumak, John D. Swinbank, Dan S. Taranu, Nikita Tewary, Grant R. Tremblay, Miguel de Val-Borro, Samuel J. Van Kooten, Zlatan Vasović, Shresth Verma, José Vinícius de Miranda Cardoso, Peter K. G. Williams, Tom J. Wilson, Benjamin Winkel, W. M. Wood-Vasey, Rui Xue, Peter Yoachim, Chen Zhang, Andrea Zonca, and Astropy Project Contributors. The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package. The Astrophysical Journal, 935:167, August 2022. ISSN 0004-637X. doi: 10.3847/1538-4357/ac7c74. URL `https://ui.adsabs.harvard.edu/abs/2022ApJ...935..167A`. Publisher: IOP ADS Bibcode: 2022ApJ...935..167A.

L. Auer. Formal Solution: EXPLICIT Answers. 288:3, January 2003. URL `https://ui.adsabs.harvard.edu/abs/2003ASPC..288....3A`. Conference Name: Stellar Atmosphere Modeling ADS Bibcode: 2003ASPC..288....3A.

L. H. Auer and D. Mihalas. On the Use of Variable Eddington Factors in Non-LTE Stellar Atmospheres Computations. Monthly Notices of the Royal Astronomical Society, 149(1):65–74, July 1970. ISSN 0035-8711, 1365-2966. doi: 10.1093/mnras/149.1.65. URL `https://academic.oup.com/mnras/article-lookup/doi/10.1093/mnras/149.1.65`.

Lawrence H. Auer and Dimitri Mihalas. Non-Lte Model Atmospheres. III. a Complete-Linearization Method. The Astrophysical Journal, 158:641, November 1969. ISSN 0004-637X. doi: 10.1086/150226. URL `https://ui.adsabs.harvard.edu/abs/1969ApJ...158..641A`. ADS Bibcode: 1969ApJ...158..641A.

M. Baes and P. Camps. SKIRT: The design of a suite of input models for Monte Carlo radiative transfer simulations. Astronomy and Computing, 12:33–44, September 2015. ISSN 2213-1337. doi: 10.1016/j.ascom.2015.05.006. URL `https://www.sciencedirect.com/science/article/pii/S2213133715000608`.

E. Baron and P. H. Hauschildt. Co-moving frame radiative transfer in spherical media with arbitraryvelocity fields. Astronomy & Astrophysics, 427(3):987–994,

December 2004. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361:20040039. URL http://www.aanda.org/10.1051/0004-6361:20040039.

S. Bladh, S. Liljegren, S. Höfner, B. Aringer, and P. Marigo. An extensive grid of DARWIN models for M-type AGB stars. I. Mass-loss rates and other properties of dust-driven winds. Astronomy and Astrophysics, 626:A100, June 2019. ISSN 0004-6361. doi: 10.1051/0004-6361/201935366. URL https://ui.adsabs.harvard.edu/abs/2019A&A...626A.100B. ADS Bibcode: 2019A&A...626A.100B.

C. Brinch and M. R. Hogerheijde. LIME - a flexible, non-LTE line excitation and radiation transfer method for millimeter and far-infrared wavelengths. Astronomy & Astrophysics, 523:A25, November 2010. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/201015333. URL http://arxiv.org/abs/1008.1492. arXiv: 1008.1492.

Peter Camps and Maarten Baes. The Failure of Monte Carlo Radiative Transfer at Medium to High Optical Depths. The Astrophysical Journal, 861(2):80, July 2018. ISSN 1538-4357. doi: 10.3847/1538-4357/aac824. URL https://iopscience.iop.org/article/10.3847/1538-4357/aac824.

M. Carlsson and J. Leenaarts. Approximations for radiative cooling and heating in the solar chromosphere. Astronomy & Astrophysics, 539:A39, March 2012. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/201118366. URL http://www.aanda.org/10.1051/0004-6361/201118366.

John I. Castor. Radiation hydrodynamics. Cambridge University Press, Cambridge, UK ; New York, 2004. ISBN 978-0-521-83309-7 978-0-521-54062-9.

T. Ceulemans, F. De Ceuster, L. Decin, and J. Yates. Magritte, a modern software library for spectral line radiative transfer. Astronomy and Computing, 49:100889, October 2024. ISSN 2213-1337. doi: 10.1016/j.ascom.2024.100889. URL https://www.sciencedirect.com/science/article/pii/S2213133724001045.

Bruce A. Chappell and Tiago M. D. Pereira. SunnyNet: A neural network approach to 3D non-LTE radiative transfer. arXiv:2112.13852 [astro-ph], December 2021. URL http://arxiv.org/abs/2112.13852. arXiv: 2112.13852.

Xingzhuo Chen, David J. Jeffery, Ming Zhong, Levi McClenny, Ulisses Braga-Neto, and Lifan Wang. Using Physics Informed Neural Networks for Supernova Radiative Transfer Simulation, November 2022. URL http://arxiv.org/abs/2211.05219. arXiv:2211.05219 [astro-ph].

A. Coenegrachts, T. Danilovich, F. De Ceuster, and L. Decin. The unusual 3D distribution of NaCl around the AGB star IK Tau, February 2023. URL http://arxiv.org/abs/2302.06221. arXiv:2302.06221 [astro-ph].

T. Danilovich, P. Bergman, K. Justtanont, R. Lombaert, M. Maercker, H. Olofsson, S. Ramstedt, and P. Royer. Detailed modelling of the circumstellar molecular line emission of the S-type AGB star W Aquilae. Astronomy and Astrophysics, 569: A76, September 2014. ISSN 0004-6361. doi: 10.1051/0004-6361/201322807. URL https://ui.adsabs.harvard.edu/abs/2014A&A...569A..76D/abstract.

Frederik De Ceuster. Simulating 3D Radiation Transport, a modern approach to discretisation and an exploration of probabilistic methods. Doctoral, UCL (University College London), January 2022. URL https://discovery.ucl.ac.uk/id/eprint/10141997/. Conference Name: UCL (University College London) Meeting Name: UCL (University College London) Publication Title: Doctoral thesis, UCL (University College London).

Frederik De Ceuster, Thomas Ceulemans, Atulit Srivastava, Ward Homan, Jan Bolte, Jeremy Yates, Leen Decin, Peter Boyle, and James Hetherington. 3D Line Radiative Transfer & Synthetic Observations with Magritte. Journal of Open Source Software, 7(71):3905, March 2022. ISSN 2475-9066. doi: 10.21105/joss.03905. URL https://joss.theoj.org/papers/10.21105/joss.03905.

Frederik De Ceuster, Thomas Ceulemans, Leen Decin, Taïssa Danilovich, and Jeremy Yates. Bayesian model reconstruction based on spectral line observations, February 2024. URL https://ui.adsabs.harvard.edu/abs/2024arXiv240218525D. Publication Title: arXiv e-prints ADS Bibcode: 2024arXiv240218525D.

A. de Vicente, T. del Pino Alemán, and J. Trujillo Bueno. Long Characteristics versus Short Characteristics in 3D Radiative Transfer Simulations of Polarized Radiation. The Astrophysical Journal, 912(1):63, May 2021. ISSN 0004-637X, 1538-4357. doi: 10.3847/1538-4357/abec6b. URL https://iopscience.iop.org/article/10.3847/1538-4357/abec6b.

L. Decin, S. Hony, A. de Koter, K. Justtanont, A. G. G. M. Tielens, and L. B. F. M. Waters. Probing the mass-loss history of AGB and red supergiant stars from CO rotational line profiles: I. Theoretical model – Mass-loss history unravelled in VY CMa. Astronomy & Astrophysics, 456(2):549–563, September 2006. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361:20065230. URL http://www.aanda.org/10.1051/0004-6361:20065230.

L. Decin, A. M. S. Richards, T. Danilovich, W. Homan, and J. A. Nuth. ALMA spectral line and imaging survey of a low and a high mass-loss rate AGB star between 335 and 362 GHz. Astronomy and Astrophysics, 615:A28, July 2018. ISSN 0004-6361. doi: 10.1051/0004-6361/201732216. URL https://ui.adsabs.harvard.edu/abs/2018A&A...615A..28D. Publisher: EDP ADS Bibcode: 2018A&A...615A..28D.

L. Decin, M. Montargès, A. M. S. Richards, C. A. Gottlieb, W. Homan, I. McDonald, I. El Mellah, T. Danilovich, S. H. J. Wallström, A. Zijlstra, A. Baudry, J. Bolte, E. Cannon, E. De Beck, F. De Ceuster, A. de Koter, J. De Ridder, S. Etoka, D. Gobrecht, M. Gray, F. Herpin, M. Jeste, E. Lagadec, P. Kervella, T. Khouri, K. Menten, T. J. Millar, H. S. P. Müller, J. M. C. Plane, R. Sahai, H. Sana, M. Van de Sande, L. B. F. M. Waters, K. T. Wong, and J. Yates. (Sub)stellar companions shape the winds of evolved stars. Science, 369(6510):1497–1500, September 2020. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.abb1229. URL http://science.sciencemag.org/content/369/6510/1497. Publisher: American Association for the Advancement of Science Section: Report.

Leen Decin. Evolution and Mass Loss of Cool Ageing Stars: a Daedalean Story. arXiv:2011.13472 [astro-ph, physics:physics], November 2020. URL http://arxiv.org/abs/2011.13472. arXiv: 2011.13472.

Frederik De Ceuster, Ward Homan, Jeremy Yates, Leen Decin, Peter Boyle, and James Hetherington. Magritte, a modern software library for 3D radiative transfer: I. Non-LTE atomic and molecular line modelling. Monthly Notices of the Royal Astronomical Society, 492(2):1812–1826, December 2019. ISSN 0035-8711, 1365-2966. doi: 10.1093/mnras/stz3557. URL https://academic.oup.com/mnras/article/492/2/1812/5681409. Number: 2.

Frederik De Ceuster, Jan Bolte, Ward Homan, Silke Maes, Jolien Malfait, Leen Decin, Jeremy Yates, Peter Boyle, and James Hetherington. magritte, a modern software library for 3D radiative transfer – II. Adaptive ray-tracing, mesh construction, and reduction. Monthly Notices of the Royal Astronomical Society, 499(4):5194–5204, November 2020. ISSN 0035-8711. doi: 10.1093/mnras/staa3199. URL https://doi.org/10.1093/mnras/staa3199.

C. P. Dullemond, A. Juhasz, A. Pohl, F. Sereshti, R. Shetty, T. Peters, B. Commercon, and M. Flock. RADMC-3D: A multi-purpose radiative transfer tool. Astrophysics Source Code Library, page ascl:1202.015, February 2012. URL https://ui.adsabs.harvard.edu/abs/2012ascl.soft02015D. ADS Bibcode: 2012ascl.soft02015D.

M. Esseldeurs, L. Siess, F. De Ceuster, W. Homan, J. Malfait, S. Maes, T. Konings, T. Ceulemans, and L. Decin. 3D simulations of AGB stellar winds - II. Ray-tracer implementation and impact of radiation on the outflow morphology. Astronomy & Astrophysics, 674:A122, June 2023. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/202346282. URL https://www.aanda.org/articles/aa/abs/2023/06/aa46282-23/aa46282-23.html. Publisher: EDP Sciences.

Paul Feautrier. Sur la resolution numerique de l'equation de transfert. Comptes Rendus Academie des Sciences (serie non specifiee), 258:3189, January 1964.

Antonio Ferrer-Sánchez, José D. Martín-Guerrero, Roberto Ruiz de Austri-Bazan, Alejandro Torres-Forné, and José A. Font. Gradient-annihilated PINNs for solving Riemann problems: Application to relativistic hydrodynamics. Computer Methods in Applied Mechanics and Engineering, 424:116906, May 2024. ISSN 0045-7825. doi: 10.1016/j.cma.2024.116906. URL https://ui.adsabs.harvard.edu/abs/2024CMAME.42416906F. ADS Bibcode: 2024CMAME.42416906F.

D. Field, M. D. Gray, and P. de St. Paer. Lineshapes of intense OH masers in star-forming regions. Astronomy and Astrophysics, Vol. 282, p. 213-224 (1994), 282: 213, February 1994. ISSN 0004-6361. URL https://ui.adsabs.harvard.edu/abs/1994A%26A...282..213F/abstract.

B. Freytag, M. Steffen, H. G. Ludwig, S. Wedemeyer-Böhm, W. Schaffenberger, and O. Steiner. Simulations of stellar convection with CO5BOLD. Journal of Computational Physics, 231:919–959, February 2012. ISSN 0021-9991. doi: 10.1016/j.jcp.2011.09.026. URL https://ui.adsabs.harvard.edu/abs/2012JCoPh.231..919F. ADS Bibcode: 2012JCoPh.231..919F.

Hans-Peter Gail and Erwin Sedlmayr. PHYSICS AND CHEMISTRY OF CIRCUMSTELLAR DUST SHELLS. Cambridge University Press, 2014.

Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering, 79(11):1309–1331, 2009. ISSN 1097-0207. doi: 10.1002/nme.2579. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2579. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2579.

P. F. Goldsmith and W. D. Langer. Molecular cooling and thermal balance of dense interstellar clouds. The Astrophysical Journal, 222:881–895, June 1978. ISSN 0004-637X. doi: 10.1086/156206. URL https://ui.adsabs.harvard.edu/abs/1978ApJ...222..881G. ADS Bibcode: 1978ApJ...222..881G.

C. A. Gottlieb, L. Decin, A. M. S. Richards, F. De Ceuster, W. Homan, S. H. J. Wallström, T. Danilovich, T. J. Millar, M. Montargès, K. T. Wong, I. McDonald, A. Baudry, J. Bolte, E. Cannon, E. De Beck, A. de Koter, I. El Mellah, S. Etoka, D. Gobrecht, M. Gray, F. Herpin, M. Jeste, P. Kervella, T. Khouri, E. Lagadec, S. Maes, J. Malfait, K. M. Menten, H. S. P. Müller, B. Pimpanuwat, J. M. C. Plane, R. Sahai, M. Van de Sande, L. B. F. M. Waters, J. Yates, and A. Zijlstra. ATOMIUM: ALMA tracing the origins of molecules in dust forming oxygen rich M-type stars. Motivation, sample, calibration, and initial results. Astronomy and Astrophysics, 660:A94, April 2022. ISSN 0004-6361. doi: 10.1051/0004-6361/202140431. URL https://ui.adsabs.harvard.edu/abs/2022A&A...660A..94G. ADS Bibcode: 2022A&A...660A..94G.

M D Gray, L Mason, and S Etoka. A new 3D maser code applied to flaring events. Monthly Notices of the Royal Astronomical Society, 477(2):2628–2639, June 2018. ISSN 0035-8711. doi: 10.1093/mnras/sty576. URL https://doi.org/10.1093/mnras/sty576.

M. A. T. Groenewegen. The photodissociation of CO in circumstellar envelopes. Astronomy and Astrophysics, 606:A67, October 2017. ISSN 0004-6361. doi: 10.1051/0004-6361/201731639. URL https://ui.adsabs.harvard.edu/abs/2017A&A...606A..67G. Publisher: EDP ADS Bibcode: 2017A&A...606A..67G.

K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann. HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere. The Astrophysical Journal, 622(2):759, April 2005. ISSN 0004-637X. doi: 10.1086/427976. URL https://iopscience.iop.org/article/10.1086/427976/meta. Publisher: IOP Publishing.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array

programming with NumPy. Nature, 585(7825):357–362, September 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2649-2. URL https://www.nature.com/articles/s41586-020-2649-2. Number: 7825 Publisher: Nature Publishing Group.

P. H. Hauschildt and E. Baron. Improved discretization of the wavelength derivative term in CMF operator splitting numerical radiative transfer. Astronomy & Astrophysics, 417(1):317–324, April 2004. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361:20034473. URL http://www.aanda.org/10.1051/0004-6361:20034473.

Peter H. Hauschildt and Edward Baron. A 3D radiative transfer framework: XI. Multi-level NLTE. Astronomy & Astrophysics, 566:A89, June 2014. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/201423574. URL http://www.aanda.org/10.1051/0004-6361/201423574.

P.H. Hauschildt. A fast operator perturbation method for the solution of the special relativistic equation of radiative transfer in spherical symmetry. Journal of Quantitative Spectroscopy and Radiative Transfer, 47(6):433–453, June 1992. ISSN 00224073. doi: 10.1016/0022-4073(92)90105-D. URL https://linkinghub.elsevier.com/retrieve/pii/002240739290105D.

T. L. Hoffmann, D. N. Sauer, A. W. A. Pauldrach, and P. J. N. Hultzsch. Non-LTE models for synthetic spectra of Type Ia supernovae: IV. A modified Feautrier scheme for opacity-sampled pseudo-continua at high expansion velocities and application to synthetic SN Ia spectra. Astronomy & Astrophysics, 569:A62, September 2014. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/201322265. URL http://www.aanda.org/10.1051/0004-6361/201322265.

I. Hubený and Dimitri Mihalas. Theory of stellar atmospheres: an introduction to astrophysical non-equilibrium quantitative spectroscopic analysis. Princeton series in astrophysics. Princeton University Press, Princeton, 2014. ISBN 978-0-691-16328-4 978-0-691-16329-1.

Susanne Höfner and Hans Olofsson. Mass loss of stars on the asymptotic giant branch. Mechanisms, models and measurements. Astronomy and Astrophysics Review, 26: 1, January 2018. ISSN 0935-4956. doi: 10.1007/s00159-017-0106-5. URL https://ui.adsabs.harvard.edu/abs/2018A&ARv..26....1H. Publisher: Springer ADS Bibcode: 2018A&ARv..26....1H.

T Kaeufer, P Woitke, M Min, I Kamp, and C Pinte. Analysing the SEDs of protoplanetary disks with machine learning. 2023.

T. Khouri, A. de Koter, L. Decin, L. B. F. M. Waters, R. Lombaert, P. Royer, B. Swinyard, M. J. Barlow, J. Alcolea, J. a. D. L. Blommaert, V. Bujarrabal, J. Cernicharo, M. a. T. Groenewegen, K. Justtanont, F. Kerschbaum, M. Maercker, A. Marston, M. Matsuura, G. Melnick, K. M. Menten, H. Olofsson, P. Planesas, E. Polehampton, Th Posch, M. Schmidt, R. Szczerba, B. Vandenbussche, and J. Yates. The wind of W Hydrae as seen by Herschel. I. The CO envelope. Astronomy & Astrophysics, Volume 561, id.A5, <NUMPAGES>13</NUMPAGES> pp.,

561:A5, January 2014. ISSN 0004-6361. doi: 10.1051/0004-6361/201322578. URL https://ui.adsabs.harvard.edu/abs/2014A%26A...561A...5K/abstract.

Rudolf Kippenhahn, Alfred Weigert, and Achim Weiss. Stellar Structure and Evolution. Astronomy and Astrophysics Library. Springer, Berlin, Heidelberg, 2012. ISBN 978-3-642-30255-8 978-3-642-30304-3. doi: 10.1007/978-3-642-30304-3. URL http://link.springer.com/10.1007/978-3-642-30304-3.

Nadiia M. Kostogryz, Friedrich Kupka, Nikolai Piskunov, Damian Fabbian, Daniel Krüger, and Laurent Gizon. Accurate Short-Characteristics Radiative Transfer in A Numerical Tool for Astrophysical RESearch (ANTARES). Solar Physics, 296 (3):46, March 2021. ISSN 1573-093X. doi: 10.1007/s11207-021-01777-6. URL https://doi.org/10.1007/s11207-021-01777-6.

Paul Kunasz and Lawrence H. Auer. Short characteristic integration of radiative transfer problems: Formal solution in two-dimensional slabs. Journal of Quantitative Spectroscopy and Radiative Transfer, 39(1):67–79, January 1988. ISSN 00224073. doi: 10.1016/0022-4073(88)90021-0. URL https://linkinghub.elsevier.com/retrieve/pii/0022407388900210.

Ryan Lagerquist, David Turner, Imme Ebert-Uphoff, Jebb Stewart, and Venita Hagerty. Using Deep Learning to Emulate and Accelerate a Radiative Transfer Model. Journal of Atmospheric and Oceanic Technology, 38(10):1673–1696, October 2021. ISSN 0739-0572, 1520-0426. doi: 10.1175/JTECH-D-21-0007.1. URL https://journals.ametsoc.org/view/journals/atot/38/10/JTECH-D-21-0007.1.xml. Publisher: American Meteorological Society Section: Journal of Atmospheric and Oceanic Technology.

S. Maes, W. Homan, J. Malfait, L. Siess, J. Bolte, F. De Ceuster, and L. Decin. SPH modelling of companion-perturbed AGB outflows including a new morphology classification scheme. Astronomy & Astrophysics, 653:A25, September 2021. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/202140823. URL https://www.aanda.org/articles/aa/abs/2021/09/aa40823-21/aa40823-21.html. Publisher: EDP Sciences.

S. Maes, M. Van de Sande, T. Danilovich, F. De Ceuster, and L. Decin. Sensitivity study of chemistry in AGB outflows using chemical kinetics. Monthly Notices of the Royal Astronomical Society, 522:4654–4673, July 2023. ISSN 0035-8711. doi: 10.1093/mnras/stad1152. URL https://ui.adsabs.harvard.edu/abs/2023MNRAS.522.4654M. Publisher: OUP ADS Bibcode: 2023MNRAS.522.4654M.

Silke Maes. Modelling environments of evolved stars: morphology, chemical sensitivity, and neural surrogate models. PhD thesis, 2024. Book Title: Modelling environments of evolved stars: morphology, chemical sensitivity, and neural surrogate models.

Silke Maes, Frederik De Ceuster, Marie Van de Sande, and Leen Decin. MACE: A Machine-learning Approach to Chemistry Emulation. The Astrophysical Journal, 969:79, July 2024. ISSN 0004-637X. doi: 10.3847/1538-4357/ad47a1. URL https://ui.adsabs.harvard.edu/abs/2024ApJ...969...79M. Publisher: IOP ADS Bibcode: 2024ApJ...969...79M.

J. Malfait, W. Homan, S. Maes, J. Bolte, L. Siess, F. De Ceuster, and L. Decin. SPH modelling of wind-companion interactions in eccentric AGB binary systems. Astronomy & Astrophysics, 652:A51, August 2021. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/202141161. URL https://www.aanda.org/articles/aa/abs/2021/08/aa41161-21/aa41161-21.html. Publisher: EDP Sciences.

J. Malfait, L. Siess, O. Vermeulen, M. Esseldeurs, S. H. J. Wallström, A. M. S. Richards, F. De Ceuster, S. Maes, J. Bolte, and L. Decin. SPH modelling of AGB wind morphology in hierarchical triple systems and a comparison to observation of R Aql. Astronomy & Astrophysics, 691:A57, November 2024a. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/202450844. URL https://www.aanda.org/10.1051/0004-6361/202450844.

Jolien Malfait. Simulating the impact of wind-companion interactions on the 3D morphology of AGB outflows. PhD thesis, 2024. Book Title: Simulating the impact of wind-companion interactions on the 3D morphology of AGB outflows.

Jolien Malfait, Lionel Siess, Mats Esseldeurs, Sofia H. J. Wallström, Alex de Koter, and Leen Decin. Impact of HI cooling and study of accretion disks in AGB wind-companion smoothed particle hydrodynamic simulations, August 2024b. URL https://ui.adsabs.harvard.edu/abs/2024arXiv240813158M. Publication Title: arXiv e-prints ADS Bibcode: 2024arXiv240813158M.

Kosei Matsumoto, Peter Camps, Maarten Baes, Frederik De Ceuster, Keiichi Wada, Takao Nakagawa, and Kentaro Nagamine. Self-consistent dust and non-LTE line radiative transfer with SKIRT. Astronomy and Astrophysics, 678: A175, October 2023. ISSN 0004-6361. doi: 10.1051/0004-6361/202347376. URL https://ui.adsabs.harvard.edu/abs/2023A&A...678A.175M. ADS Bibcode: 2023A&A...678A.175M.

C. F. McKee, J. W. V. Storey, D. M. Watson, and S. Green. Far-infrared rotational emission by carbon monoxide. The Astrophysical Journal, 259:647, August 1982. ISSN 0004-637X, 1538-4357. doi: 10.1086/160200. URL http://adsabs.harvard.edu/doi/10.1086/160200.

Collin McLeod, D. John Hillier, and Luc Dessart. Carbon monoxide cooling in radiative transfer modelling of supernovae. Monthly Notices of the Royal Astronomical Society, 532:549–562, July 2024. ISSN 0035-8711. doi: 10.1093/mnras/stae1513. URL https://ui.adsabs.harvard.edu/abs/2024MNRAS.532..549M. Publisher: OUP ADS Bibcode: 2024MNRAS.532..549M.

Shyam H. Menon, Christoph Federrath, Mark R. Krumholz, Rolf Kuiper, Benjamin D. Wibking, and Manuel Jung. VETTAM: a scheme for radiation hydrodynamics with adaptive mesh refinement using the variable Eddington tensor method. Monthly Notices of the Royal Astronomical Society, 512:401–423, May 2022. ISSN 0035-8711. doi: 10.1093/mnras/stac485. URL https://ui.adsabs.harvard.edu/abs/2022MNRAS.512..401M. Publisher: OUP ADS Bibcode: 2022MNRAS.512..401M.

N. Moens, L. G. Poniatowski, L. Hennicker, J. O. Sundqvist, I. El Mellah, and N. D. Kee. First 3D radiation-hydrodynamic simulations of Wolf-Rayet winds. Astronomy & Astrophysics, 665:A42, September 2022. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/202243451. URL https://www.aanda.org/10.1051/0004-6361/202243451.

A. V. Nesterenok. H2O maser emission in circumstellar envelopes around AGB stars: Physical conditions in gas-dust clouds. Astronomy Letters, 39(10):717–728, October 2013. ISSN 1063-7737, 1562-6873. doi: 10.1134/S106377371309003X. URL http://link.springer.com/10.1134/S106377371309003X.

David A. Neufeld and Michael J. Kaufman. Radiative Cooling of Warm Molecular Gas. The Astrophysical Journal, 418:263, November 1993. ISSN 0004-637X. doi: 10.1086/173388. URL https://ui.adsabs.harvard.edu/abs/1993ApJ...418..263N. ADS Bibcode: 1993ApJ...418..263N.

David A. Neufeld and Gary J. Melnick. Excitation of millimeter and submillimeter water masers. The Astrophysical Journal, 368:215, February 1991. ISSN 0004-637X, 1538-4357. doi: 10.1086/169685. URL http://adsabs.harvard.edu/doi/10.1086/169685.

Kin-Chue Ng. Hypernetted chain solutions for the classical one-component plasma up to Γ=7000. The Journal of Chemical Physics, 61(7):2680–2689, October 1974. ISSN 0021-9606. doi: 10.1063/1.1682399. URL https://aip.scitation.org/doi/abs/10.1063/1.1682399. Publisher: American Institute of Physics.

Gordon L. Olson, L.H. Auer, and J. Robert Buchler. A rapidly convergent iterative solution of the non-LTE line radiation transfer problem. Journal of Quantitative Spectroscopy and Radiative Transfer, 35(6):431–442, June 1986. ISSN 00224073. doi: 10.1016/0022-4073(86)90030-0. URL https://linkinghub.elsevier.com/retrieve/pii/0022407386900300.

Kazuyuki Omukai. Protostellar Collapse with Various Metallicities. The Astrophysical Journal, 534:809–824, May 2000. ISSN 0004-637X. doi: 10.1086/308776. URL https://ui.adsabs.harvard.edu/abs/2000ApJ...534..809O. Publisher: IOP ADS Bibcode: 2000ApJ...534..809O.

Kazuyuki Omukai, Takashi Hosokawa, and Naoki Yoshida. LOW-METALLICITY STAR FORMATION: PRESTELLAR COLLAPSE AND PROTOSTELLAR ACCRETION IN THE SPHERICAL SYMMETRY. The Astrophysical Journal, 722(2):1793–1815, October 2010. ISSN 0004-637X, 1538-4357. doi: 10.1088/0004-637X/722/2/1793. URL https://iopscience.iop.org/article/10.1088/0004-637X/722/2/1793.

Terry D. Oswalt and Martin Adrian Barstow. Planets, stars and stellar systems. Springer, Dordrecht New York, 2013. ISBN 978-94-007-5615-1.

F. Paletou and C. Peymirat. Full non-LTE spectral line formation. I. Setting the stage. Astronomy and Astrophysics, 649:A165, May 2021. ISSN 0004-6361. doi: 10.

1051/0004-6361/202040005. URL https://ui.adsabs.harvard.edu/abs/2021A&A...649A.165P. ADS Bibcode: 2021A&A...649A.165P.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. 2019.

A. W. A. Pauldrach, T. L. Hoffmann, and M. Lennon. Radiation-driven winds of hot luminous stars: XIII. A description of NLTE line blocking and blanketing towards realistic models for expanding atmospheres. Astronomy & Astrophysics, 375(1):161–195, August 2001. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361:20010805. URL http://www.aanda.org/10.1051/0004-6361:20010805.

C. Pinte, F. Ménard, G. Duchêne, and P. Bastien. Monte Carlo radiative transfer in protoplanetary disks. Astronomy and Astrophysics, 459:797–804, December 2006. ISSN 0004-6361. doi: 10.1051/0004-6361:20053275. URL https://ui.adsabs.harvard.edu/abs/2006A&A...459..797P. ADS Bibcode: 2006A&A...459..797P.

L. G. Poniatowski, J. O. Sundqvist, N. D. Kee, S. P. Owocki, P. Marchant, L. Decin, A. de Koter, L. Mahy, and H. Sana. Dynamically inflated wind models of classical Wolf-Rayet stars. Astronomy & Astrophysics, 647:A151, March 2021. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/202039595. URL https://www.aanda.org/10.1051/0004-6361/202039595.

Daniel J. Price, James Wurster, Terrence S. Tricco, Chris Nixon, Stéven Toupin, Alex Pettitt, Conrad Chan, Daniel Mentiplay, Guillaume Laibe, Simon Glover, Clare Dobbs, Rebecca Nealon, David Liptai, Hauke Worpel, Clément Bonnerot, Giovanni Dipierro, Giulia Ballabio, Enrico Ragusa, Christoph Federrath, Roberto Iaconi, Thomas Reichardt, Duncan Forgan, Mark Hutchison, Thomas Constantino, Ben Ayliffe, Kieran Hirsh, and Giuseppe Lodato. Phantom: A Smoothed Particle Hydrodynamics and Magnetohydrodynamics Code for Astrophysics. Publications of the Astronomical Society of Australia, 35:e031, September 2018. ISSN 1323-3580. doi: 10.1017/pasa.2018.25. URL https://ui.adsabs.harvard.edu/abs/2018PASA...35...31P. ADS Bibcode: 2018PASA...35...31P.

J. Puls, M. A. Urbaneja, R. Venero, T. Repolust, U. Springmann, A. Jokuthy, and M. R. Mokiem. Atmospheric NLTE-models for the spectroscopic analysis of blue starswith winds: II. Line-blanketed models. Astronomy & Astrophysics, 435(2): 669–698, May 2005. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361:20042365. URL http://www.aanda.org/10.1051/0004-6361:20042365.

M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378: 686–707, February 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

A. Asensio Ramos and J. Trujillo Bueno. Very Efficient Methods for Multilevel Radiative Transfer in Atomic and Molecular Lines. EAS Publications Series, 18: 25–48, 2006. ISSN 1633-4760, 1638-1963. doi: 10.1051/eas:2006003. URL http://arxiv.org/abs/astro-ph/0512248. arXiv: astro-ph/0512248.

S. Ramstedt, F. L. Schöier, H. Olofsson, and A. A. Lundgren. On the reliability of mass-loss-rate estimates for AGB stars. Astronomy & Astrophysics, 487(2):645–657, August 2008. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361:20078876. URL http://www.aanda.org/10.1051/0004-6361:20078876.

J. Rosdahl, J. Blaizot, D. Aubert, T. Stranex, and R. Teyssier. ramses-rt: radiation hydrodynamics in the cosmological context. Monthly Notices of the Royal Astronomical Society, 436(3):2188–2231, December 2013. ISSN 1365-2966, 0035-8711. doi: 10.1093/mnras/stt1722. URL http://academic.oup.com/mnras/article/436/3/2188/1247446/ramsesrt-radiation-hydrodynamics-in-the.

Laurence S. Rothman, Robert R. Gamache, Aaron Goldman, Linda R. Brown, Robert A. Toth, Herbert M. Pickett, Robert L. Poynter, J. M. Flaud, C. Camy-Peyret, A. Barbe, N. Husson, Curtis P. Rinsland, and Mary Ann H. Smith. The HITRAN database: 1986 edition. Applied Optics, 26:4058–4097, October 1987. ISSN 0003-6935. doi: 10.1364/AO.26.004058. URL https://ui.adsabs.harvard.edu/abs/1987ApOpt..26.4058R. ADS Bibcode: 1987ApOpt..26.4058R.

G. B. Rybicki and D. G. Hummer. An accelerated lambda iteration method for multilevel radiative transfer. I - Non-overlapping lines with background continuum. Astronomy and Astrophysics, 245:171–181, May 1991. ISSN 0004-6361. URL http://adsabs.harvard.edu/abs/1991A%26A...245..171R.

R. Sahai. A New Self-consistent Model of Circumstellar CO Emission for Deriving Mass-Loss Rates in Red Giants. I. The Carbon-rich Star U Camelopardalis. The Astrophysical Journal, 362:652, October 1990. ISSN 0004-637X. doi: 10.1086/169303. URL https://ui.adsabs.harvard.edu/abs/1990ApJ...362..652S. ADS Bibcode: 1990ApJ...362..652S.

M. Sampoorna and K. N. Nagendra. POLARIZED LINE FORMATION IN NON-MONOTONIC VELOCITY FIELDS. The Astrophysical Journal, 833(1):32, December 2016. ISSN 1538-4357. doi: 10.3847/1538-4357/833/1/32. URL https://iopscience.iop.org/article/10.3847/1538-4357/833/1/32.

A. a. C. Sander, W.-R. Hamann, H. Todt, R. Hainich, and T. Shenar. Coupling hydrodynamics with comoving frame radiative transfer - I. A unified approach for OB and WR stars. Astronomy & Astrophysics, 603:A86, July 2017. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/201730642. URL https://www.aanda.org/articles/aa/abs/2017/07/aa30642-17/aa30642-17.html. Publisher: EDP Sciences.

K. Schoenberg. Molecular line spectra from circumstellar envelopes. I. Excitation of12CO, 13CO and CS and the formation of line profiles. Astronomy and Astrophysics, 195:198–207, April 1988. ISSN 0004-6361. URL https://ui.adsabs.harvard.edu/abs/1988A&A...195..198S. ADS Bibcode: 1988A&A...195..198S.

F. L. Schöier, F. F. S. van der Tak, E. F. van Dishoeck, and J. H. Black. An atomic and molecular database for analysis of submillimetre line observations. Astronomy and Astrophysics, 432:369–379, March 2005. ISSN 0004-6361. doi: 10.1051/0004-6361:20041729. URL http://adsabs.harvard.edu/abs/2005A%26A...432..369S.

Avram Sidi. A convergence study for reduced rank extrapolation on nonlinear systems. Numerical Algorithms, 84(3):957–982, July 2020. ISSN 1017-1398, 1572-9265. doi: 10.1007/s11075-019-00788-6. URL http://link.springer.com/10.1007/s11075-019-00788-6.

L. Siess, W. Homan, S. Toupin, and D. J. Price. 3D simulations of AGB stellar winds: I. Steady winds and dust formation. Astronomy & Astrophysics, 667:A75, November 2022. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/202243540. URL https://www.aanda.org/10.1051/0004-6361/202243540.

V. V. Sobolev. Moving Envelopes of Stars. Harvard University Press, 1960. ISBN 978-0-674-86465-8. doi: 10.4159/harvard.9780674864658. URL https://www.degruyter.com/document/doi/10.4159/harvard.9780674864658/html. Publication Title: Moving Envelopes of Stars.

Shiqi Su, Frederik De Ceuster, Jaehoon Cha, Mark I. Wilkinson, Jeyan Thiyagalingam, Jeremy Yates, Yi-Hang Zhu, and Jan Bolte. Emulating CO Line Radiative Transfer with Deep Learning, July 2025. URL https://ui.adsabs.harvard.edu/abs/2025arXiv250711398S. ADS Bibcode: 2025arXiv250711398S.

Jonathan Tennyson, Peter F. Bernath, Alain Campargue, Attila G. Császár, Ludovic Daumont, Robert R. Gamache, Joseph T. Hodges, Daniel Lisak, Olga V. Naumenko, Laurence S. Rothman, Ha Tran, Nikolai F. Zobov, Jeanna Buldyreva, Chris D. Boone, Maria Domenica De Vizia, Livio Gianfrani, Jean-Michel Hartmann, Robert McPheat, Damien Weidmann, Jonathan Murray, Ngoc Hoa Ngo, and Oleg L. Polyansky. Recommended isolated-line profile for representing high-resolution spectroscopic transitions (IUPAC Technical Report). Pure and Applied Chemistry, 86(12):1931–1943, December 2014. ISSN 1365-3075. doi: 10.1515/pac-2014-0208. URL https://www.degruyter.com/document/doi/10.1515/pac-2014-0208/html. Publisher: De Gruyter.

B. Tessore, C. Pinte, J. Bouvier, and F. Ménard. Atomic line radiative transfer with MCFOST: I. Code description and benchmarking. Astronomy & Astrophysics, 647: A27, March 2021. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/202039697. URL https://www.aanda.org/10.1051/0004-6361/202039697.

A G G M Tielens. The Physics and Chemistry of the Interstellar Medium. Cambridge University Press, 2005.

M. Van de Sande, C. Walsh, and T. J. Millar. Chemical modelling of dust-gas chemistry within AGB outflows III. Photoprocessing of the ice and return to the ISM. Monthly Notices of the Royal Astronomical Society, 501(1):491–506, December 2020. ISSN 0035-8711, 1365-2966. doi: 10.1093/mnras/staa3689. URL http://arxiv.org/abs/2011.11563. arXiv: 2011.11563.

G.-J. van Zadelhoff, C. P. Dullemond, F. F. S. van der Tak, J. A. Yates, S. D. Doty, V. Ossenkopf, M. R. Hogerheijde, M. Juvela, H. Wiesemeyer, and F. L. Schöier. Numerical methods for non-LTE line radiative transfer:Performance and convergence characteristics. Astronomy & Astrophysics, 395(1):373–384, November 2002. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361:20021226. URL http://www.aanda.org/10.1051/0004-6361:20021226.

M Van de Sande, C Walsh, T P Mangan, and L Decin. Chemical modelling of dust–gas chemistry within AGB outflows – I. Effect on the gas-phase chemistry. Monthly Notices of the Royal Astronomical Society, 490(2):2023–2041, December 2019. ISSN 0035-8711, 1365-2966. doi: 10.1093/mnras/stz2702. URL https://academic.oup.com/mnras/article/490/2/2023/5579025.

M Van de Sande, C Walsh, T Danilovich, F De Ceuster, and T Ceulemans. Modelling predicts a molecule-rich disc around the AGB star L2 Puppis. Monthly Notices of the Royal Astronomical Society, 532(1):734–754, July 2024. ISSN 0035-8711. doi: 10.1093/mnras/stae1553. URL https://doi.org/10.1093/mnras/stae1553.

Owen Vermeulen, Mats Esseldeurs, Jolien Malfait, Thomas Ceulemans, Lionel Siess, Kosei Matsumoto, Frederik De Ceuster, Taïssa Danilovich, Camille Landri, and Leen Decin. Impact of a binary companion in AGB outflows on CO spectral lines. Astronomy & Astrophysics, June 2025. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/202554905. URL http://arxiv.org/abs/2507.03583. arXiv:2507.03583 [astro-ph].

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods, 17(3):261–272, March 2020. ISSN 1548-7105. doi: 10.1038/s41592-019-0686-2. URL https://www.nature.com/articles/s41592-019-0686-2. Publisher: Nature Publishing Group.

Paul J. Viscuso and David F. Chernoff. CO Cooling Rates: Revised Calculations. The Astrophysical Journal, 327:364, April 1988. ISSN 0004-637X. doi: 10.1086/166199. URL https://ui.adsabs.harvard.edu/abs/1988ApJ...327..364V. Publisher: IOP ADS Bibcode: 1988ApJ...327..364V.

A. P. Whitworth and S. E. Jaffa. A simple approach to CO cooling in molecular clouds. Astronomy & Astrophysics, 611:A20, March 2018. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/201731871. URL https://www.aanda.org/10.1051/0004-6361/201731871.

Benhui Yang, P. C. Stancil, N. Balakrishnan, and R. C. Forrey. Rotational Quenching of CO due to H2 Collisions. The Astrophysical Journal, 718:1062–1069, August 2010.

ISSN 0004-637X. doi: 10.1088/0004-637X/718/2/1062. URL https://ui.adsabs.harvard.edu/abs/2010ApJ...718.1062Y. ADS Bibcode: 2010ApJ...718.1062Y.

Benhui Yang, N. Balakrishnan, P. Zhang, X. Wang, J. M. Bowman, R. C. Forrey, and P. C. Stancil. Full-dimensional quantum dynamics of CO in collision with H2. The Journal of Chemical Physics, 145(3):034308, July 2016. ISSN 0021-9606. doi: 10.1063/1.4958951. URL https://doi.org/10.1063/1.4958951.

J. A. Yates, D. Field, and M. D. Gray. Non-local radiative transfer for molecules: modelling population inversions in water masers. Monthly Notices of the Royal Astronomical Society, 285(2):303–316, February 1997. ISSN 0035-8711, 1365-2966. doi: 10.1093/mnras/285.2.303. URL https://academic.oup.com/mnras/article-lookup/doi/10.1093/mnras/285.2.303.

Y. J. Yun and Y.-S. Park. Non-local radiative transfer of SiO masers in Miras: Implementation of the coupled escape probability method. Astronomy & Astrophysics, 545:A136, September 2012. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/201218953. URL http://www.aanda.org/10.1051/0004-6361/201218953.

Karin I. Öberg, Viviana V. Guzmán, Catherine Walsh, Yuri Aikawa, Edwin A. Bergin, Charles J. Law, Ryan A. Loomis, Felipe Alarcón, Sean M. Andrews, Jaehan Bae, Jennifer B. Bergner, Yann Boehler, Alice S. Booth, Arthur D. Bosman, Jenny K. Calahan, Gianni Cataldi, L. Ilsedore Cleeves, Ian Czekala, Kenji Furuya, Jane Huang, John D. Ilee, Nicolas T. Kurtovic, Romane Le Gal, Yao Liu, Feng Long, François Ménard, Hideko Nomura, Laura M. Pérez, Chunhua Qi, Kamber R. Schwarz, Anibal Sierra, Richard Teague, Takashi Tsukagoshi, Yoshihide Yamato, Merel L. R. Van 'T Hoff, Abygail R. Waggoner, David J. Wilner, and Ke Zhang. Molecules with ALMA at Planet-forming Scales (MAPS). I. Program Overview and Highlights. The Astrophysical Journal Supplement Series, 257(1):1, November 2021. ISSN 0067-0049, 1538-4365. doi: 10.3847/1538-4365/ac1432. URL https://iopscience.iop.org/article/10.3847/1538-4365/ac1432.

Jiří Štěpán and Javier Trujillo Bueno. PORTA: A three-dimensional multilevel radiative transfer code for modeling the intensity and polarization of spectral lines with massively parallel computers. Astronomy & Astrophysics, 557:A143, September 2013. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/201321742. URL http://www.aanda.org/10.1051/0004-6361/201321742.

Jiří Štěpán, Tanausú del Pino Alemán, and Javier Trujillo Bueno. Novel framework for the three-dimensional NLTE inverse problem. Astronomy & Astrophysics, 659: A137, March 2022. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/202142079. URL https://www.aanda.org/articles/aa/abs/2022/03/aa42079-21/aa42079-21.html. Publisher: EDP Sciences.

Statement on the use of Generative AI

I have used generative AI assistance tools, such as Github Copilot. Other tools such as ChatGPT were used for mere language assistance. Generative AI was used for generating programming code. I have used Github Copilot to generate short snippets of code, i.e. using it as autocomplete tool. The resulting code was inspected manually, and when needed, corrected.

The text, code, and images in this thesis are my own (unless otherwise specified). Generative AI has only been used in accordance with the KU Leuven guidelines and appropriate references have been added. I have reviewed and edited the content as needed and I take full responsibility for the content of the thesis.

FACULTY OF SCIENCE
DEPARTMENT OF PHYSICS AND ASTRONOMY
Celestijnenlaan 200D
B-3001 Leuven

https://thomasceulemans.github.io