

MASARYKOVA
UNIVERZITA

MASARYKOVA UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA
ÚSTAV TEORETICKÉ FYZIKY A ASTROFYZIKY



Radiation and Collisional Processes in Stellar Atmospheres

PhD thesis

Jakub Fišák

Vedoucí práce: doc. RNDr. Jiří Kubát, CSc.

Brno 2022

Bibliografický záznam

Autor:	Mgr. Jakub Fišák Přírodovědecká fakulta, Masarykova univerzita Ústav teoretické fyziky a astrofyziky
Název práce:	Zářivé a srážkové procesy ve hvězdných atmosférách
Studijní program:	Fyzika
Studijní obor:	Teoretická fyzika a astrofyzika
Vedoucí práce:	Doc. RNDr. Jiří Kubát, CSc.
Konzultant:	Mgr. Brankica Kubátová, Ph.D.
Akademický rok:	2022
Počet stran:	xi + 85
Klíčová slova:	hvězdy: atmosféry, hvězdy: vítr, metody přenosu záření: numerické

Bibliographic Entry

Author: Mgr. Jakub Fišák
Faculty of Science, Masaryk University
The Department of Theoretical Physics and Astrophysics

Title of Thesis: Radiative and collisional processes in stellar atmospheres

Degree Programme Physics

Field of Study: Theoretical Physics and Astrophysics

Supervisor: Doc. RNDr. Jiří Kubát, CSc.

Consultant: Mgr. Brankica Kubátová, Ph.D.

Academic Year: 2022

Number of Pages: xi + 85

Keywords: stars: atmospheres – stars: winds,
outflows – radiative transfer – methods: numerical

Abstrakt

V této práci prezentujeme Monte Carlo kód pro přenos záření v trojrozměrném expandujícím prostředí. Tento kód načte vstupní model atmosféry, spočítá ionizační a excitační rovnováhu látky za předpokladu LTE aproximace, vytvoří vlastní kartézský grid (propGrid) a pomocí metody Monte Carlo vyřeší přenos záření v příslušném expandujícím prostředí. V této práci se zaměříme na testování propagační numerické sítě, testování interakce v čarách a samotné aplikace na trojrozměrné modely: testovací modely a hydrodynamický model.

Abstract

In this thesis, we present a Monte Carlo radiative transfer code in 3-D expanding outflows. This program reads an input model of an atmosphere, then, assuming LTE approximation, calculates ionization and excitation equilibrium of matter. The code creates its own propagation cartesian grid (propGrid) and solves the radiation transfer in the expanding environment. This thesis is focused on testing of propGrid and interaction in lines, and finally, radiative transfer in the 3-D cases, namely, testing models, and, finally, hydrodynamical model.

ZADÁNÍ
DISERTAČNÍ PRÁCE

Akademický rok: 2022/2023

Ústav:	Přírodovědecká fakulta
Student:	Mgr. Jakub Fišák
Program:	Fyzika
Specializace:	Astrofyzika

Ředitel ústavu PŘF MU Vám ve smyslu Studijního a zkušebního řádu MU určuje disertační práci s názvem:

Název práce:	Srážkové a zářivé procesy ve hvězdných atmosférách
Název práce anglicky:	Collisional and radiative processes in stellar atmospheres
Jazyk závěrečné práce:	angličtina

Oficiální zadání:

Model hvězdné atmosféry je určen řešením soustavy hydrodynamických rovnic a rovnice přenosu záření. Struktura NLTE modelu hvězdné atmosféry je značně ovlivněna interakcí hmoty a záření. V NLTE aproximaci je tato interakce explicitně vzata v úvahu a rozdělení do excitačních a ionizačních stavů atomů je dáno současným řešením rovnic kinetické (statistické) rovnováhy a rovnice přenosu záření. Do tohoto řešení se běžně započítávají excitace, deexcitace, ionizace a rekombinace způsobené zářením nebo nepružným srážkami atomů s elektrony. Existuje však řada dalších procesů (například anizotropní rozptyl záření na elektronech, Rayleighův a Ramanův rozptyl, autoionizace, dielektronická rekombinace, Augerova ionizace, nepružné srážky s kladně nabitými ionty, přenos náboje), které jsou běžně zanedbávány, ale které mohou v některých případech strukturu atmosféry a vystupující záření významně ovlivnit. Cílem práce bude zjistit důsledky těchto procesů.

Literatura:

RYBICKI, George B. a Alan P. LIGHTMAN. *Radiative processes in astrophysics*. New York: John Wiley & Sons, 1979. xv, 382. ISBN 0471827592.

MIHALAS, Dimitri. *Stellar atmospheres*. 2nd ed. San Francisco: W.H. Freeman and Company, 1978. xx, 632. ISBN 0716703599.

Vedoucí práce:	doc. RNDr. Jiří Kubát, CSc.
Konzultant:	Mgr. Brankica Kubátová, Ph.D.
Datum zadání práce:	25. 7. 2014
V Brně dne:	4. 11. 2022

Acknowledgement

I would like to thank my supervisor, Jiří Kubát, for his patience, all discussions we had. He forced me to truly understand everything in the thesis. Markus Kromer for his advices during Macro-atom approach and parallelization of the code, and the opportunity to stay twelve weeks in total in Heidelberg, which was supported by the program Erasmus+. I would like to thank Brankica Kubátová, Petr Kurfürst, Jiří Krtička, and Milan Prvák for their help, discussions and support. Finally, I would like to thank Jon Sundqvist, Nico Moens and the other members of Jon's research group for a great stay in Leuven, where I completed many important tasks. The stay in Leuven was supported by a mobility project CZ.02.2.69/0.0/0.0/18_053/0016972 provided by EU and MŠMT. This research has made use of the NASA's Astrophysics Data System Abstract Service.

This research made use of Tardis, a community-developed software package for spectral synthesis in supernovae ([Kerzendorf & Sim, 2014](#), [Kerzendorf et al., 2019](#)). The development of Tardis received support from GitHub, the Google Summer of Code initiative, and from ESA's Summer of Code in Space program. Tardis is a fiscally sponsored project of NumFOCUS. Tardis makes extensive use of Astropy and Pyne.

Declaration

Prohlašuji, že jsem svoji disertační práci vypracoval samostatně s využitím informačních zdrojů, které jsou v práci citovány.

Brno 15.12.2022

.....
Jakub Fišák

Contents

1	Hot stars and their winds	1
1.1	Beyond the stellar winds: other objects with strong outflows	2
1.1.1	Supernovae	2
1.1.2	Be stars	3
1.2	Observations of stellar winds	4
1.3	The physics of stellar wind	6
1.3.1	Hydrodynamical equations	7
1.3.2	Radiative force	7
1.4	Commonly used models of stellar winds	9
2	NLTE Monte Carlo radiative transfer	11
2.1	Monte Carlo method	11
2.2	Radiative transfer equation	13
2.3	Probabilistic interpretation of the radiative transfer equation	14
2.4	Radiative packets	15
2.4.1	Creation of radiative packets	15
2.4.2	Propagation and interaction of radiative packets	17
2.4.3	Calculation of emergent spectra	20
2.5	Macro-atom transition probabilities	21
2.5.1	Internal energy packets	22
2.5.2	Kinetic energy packets	26
3	The general code structure	28
3.1	Atomic data	28
3.2	Model grid	30
3.3	Propagation grid	31
3.4	Adaptive grid	33
3.5	The connection between the modGrid and the propGrid	35
3.6	The packet propagation through the propGrid	36
4	Tests of the propGrid	39
4.1	The model setup	39
4.2	The tested properties	40
4.3	The basic propagation grid	41
4.4	The adaptive propagation grid	45
5	A comparison with the Tardis code	49

6	Basic applications in multidimensional cases	54
6.1	A simple testing 3-D model	54
6.1.1	Velocity calculation	55
6.1.2	Emergent spectra	58
6.1.3	Computational details	58
6.1.4	Results	58
6.2	Hydrodynamical model MPI-AMRVAC	61
6.2.1	Set up of model	62
6.2.2	Numerical grids	62
6.2.3	A treatment of high-opacity domains	64
6.2.4	Preliminary results	67
6.2.5	Further improvements	67
7	Summary	71
7.1	Future work	72
A	Radiative and collisional rates used in the code	76
A.1	Radiative rates	76
A.1.1	Bound-free and free-bound rates	77
A.2	Collisional rates	78
A.2.1	Bound-bound transitions	78
A.2.2	Bound-free transitions	78
A.2.3	Free-bound transitions	78
A.3	Emissivities	79
B	Ionisation and excitation equilibrium	80
	Bibliography	81

Introduction

The very first discovering of space was not scientific at all. People did not understand physics, as proved by the fact that they were describing the stars in the sky creating tales to explain their origin, the reason of their existence and they assumed that stars do not change in time. Some of those myths are still evident in today's astronomy: for example, the names of constellations are originated in the Greek mythology. Throughout history, Man discovered the laws of this world. In the ancient Egypt, the helical rise of Sirius predicted forthcoming floods. Later, sailors were calculating their position based on the stars. Johannes Kepler was the first astrophysicist, to found out the laws of planets movement and used actual physical laws to describe objects in space and he definitely removed the Earth from the centre of solar system. Since those times, the astronomers deciphered many other interesting questions; furthermore, they found out, that our planet is not in a special place in the space. The research of the Space continued beyond our dreams; we discovered the planets of the Solar system (the Voyager), astronauts started exploring the universe. Nowadays, our research is no less exciting; our measurements are precise enough to measure gravitational waves created during a merge of two black holes, we saw very first photo of a black hole. Our computational sources are large enough to simulate the sun photosphere in three dimensions.

The presented work is just only a water-drop in an ocean of intriguing research. However, mankind did not have so many informations about stars in previous ages, paradoxically, we have not been so far away from space. Light pollution limits significantly the number of objects visible from the cities; citizens from bigger cities cannot see, what the Milky way or a sky full of stars looks like.

Stars are very important objects in the Universe, especially from the life's point of view. There would not be any heavier elements than primordial helium and lithium if no star existed. There would be no planet with a solid surface, and organic chemistry does not occur. The heavier elements are developed via thermonuclear reactions which take place in stellar cores. Firstly helium via the p-p chain and CNO cycle (core of massive stars), and, finally, heavy elements up to iron are created. The atoms have to be transported back to space by stellar winds or by more cataclysmic process: the supernova ejecta. Both these processes: stellar winds are found in the case of hot stars, moreover, hot stars can end as supernovae.

Hot stars used to be treated with simple radiative transfer models. The atmospheres used to be believed to be static, because the dominant energy transport is due the radiative transfer. However, later the instabilities were measured and proven by numerical calculations. The 3-D modelling is starting to be the main method to study physical phenomena in stellar winds, for this reason the presented code is being developed to be full 3-D; the structure of this thesis is following.

Chapter 1 is a brief introduction to the stellar wind phenomena, physics of stellar winds and their parameters. Mainly the characteristic spectra profiles are described, and, finally, the existing codes used for the stellar wind research. Some pictures in this chapter are used with a kind agreement by Jiří Krtička. Chapter 2 describes the physics of radiative transfer; a Monte Carlo approximation 'Macro atom' is introduced and all possible transitions are listed in this chapter. The next chapter,

number 3, brings a description of our code. Starting with atomic data, then introducing modGrid and propGrid. Chapter 4 is a discussion about the propGrid efficiency and accuracy. Several tests were run, for artificial physical conditions. In Chapter 5, an implementation of line interactions is tested. The emergent spectra were compared with the spectra calculated with the Tardis code. And, finally, the very first applications 3-D cases are showed. First case is a simple pseudo 3-D model, the second case describes the hydro-model MPI-AMRVAC and first spectra calculated by our code for this code.

CHAPTER 1

Hot stars and their winds

Searching for the basic definition of ‘star’, one would find: ‘A star is an object with thermonuclear reactions inside its core are processed in a significant period of time.’ There is a wide range of stars in a different evolutionary state. Let us focus on main sequence stars: main-sequence stars are burning hydrogen via the p-p chain and CNO cycle, besides, the stars can be divided into two main groups: hot stars and cool stars. The main difference consists in the mechanism of energy transport from the stellar interior outwards, as shown in Fig. 1.1. Cool stars are characterized by a solar-type activity, mainly driven by turbulent plasma, such as protuberations, eruptions, sun spots, which are not present in the cases of atmospheres of hot stars; objects which we study in this thesis. The category of hot-stars does not include only main-sequence stars; this chapter and the whole thesis is focused on hot stars and related objects.

Hot stars is a category of stars with effective temperature higher than 7000K, concretely, main sequence: O, B, A type stars, and also Wolf-Rayet stars (post main-sequence stadium). There are other stellar types in the list of hot stars, for example, Herbig Ae/Be stars, giants or supergiants of the population I, central stars of planetary nebulae, RR Lyrae variables, etc. Their importance is due to the fact that they enrich the interstellar environment with heavy elements and create large bubbles via their strong winds.

The phenomenon of stellar winds is described as a steady emission of matter from the star to the circumstellar environment consisting of charged particles and dust. We can observe stellar winds ourselves, concretely, aurora borealis is a consequence of the solar mass ejecta which hits earth magnetosphere. However, Sun’s stellar wind changes during the time depending on solar activity and it can cause damage to electronics and communication technology. Nevertheless, this thesis is focused

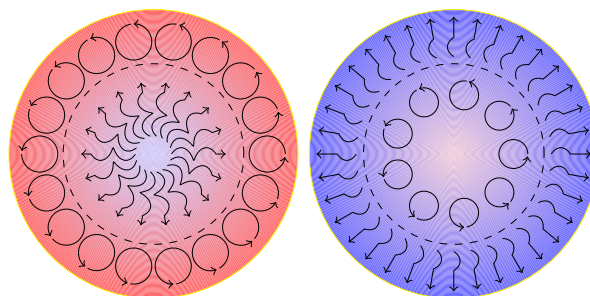


Figure 1.1: Dominant mechanism of energy transport in cool and hot stars. *Left:* Radiation provides energy transport inside cores of cool stars, whilst convection dominates in atmospheres. *Right:* the radiative transfer is not efficient inside hot star cores, therefore convection is leading mechanism, however, energy in atmospheres is transported by radiation.

Name of a star	Spectral type	L	\dot{M}	v_∞
		L_\odot	M_\odot/yr	km/s
ζ Pup	O4 I (f)	7.9×10^5	6×10^{-6}	2200
λ Cep	O6 I	6.8×10^5	5×10^{-6}	2250
ε Ori	B0 Ia	4.6×10^5	4×10^{-6}	1500
P Cyg	B1 Ia ⁺	7.2×10^5	2×10^{-5}	210
HD 12953	A1 Iae	1.3×10^5	4×10^{-7}	150
HD 93250	O3V	1.9×10^6	5×10^{-6}	3250
HD 15629	O5V	8.9×10^5	8×10^{-7}	3000
τ Sco	B0 V	3.2×10^4	7×10^{-9}	2000
WR 1	WN5	1.0×10^5	6×10^{-5}	2000
WR 111	WC5	2.8×10^5	1×10^{-5}	2200

Table 1.1: Selected parameters of several OBA supergiants, main-type hot stars, Wolf-Rayet stars and central stars of planetary nebulae; adopted from [Carroll & Ostlie \(2017\)](#) ([Kudritzki et al., 1999](#), [Lamers & Cassinelli, 1999](#), [Pauldrach et al., 2004](#)).

mainly on hot stars winds, which differ from the solar (and from cool stars in general) winds, for example, hot star can lose significant mass via the stellar wind, which affects its evolution.

Stellar winds are characterized with two parameters: the first parameter refers to amount of matter lost per unit time via the wind – mass loss rate. It is denoted with \dot{M} ; the second parameter refers to the velocity of outflow in a large distance from the star, where all forces are negligible; theoretically in the infinite distance, therefore this velocity is denoted v_∞ . The basic wind parameters are listed in the Table 1.1.

1.1

Beyond the stellar winds: other objects with strong outflows

We do not study just objects with strong stellar winds, but supernovae too. Therefore, this section is oriented on supernovae description. In addition, Be stars are mentioned, mainly because of common modelling of stellar wind and disc combined, furthermore, an interaction of supernova shock wave with circumstellar disc can be modelled. The information is quoted from [Carroll & Ostlie \(2017\)](#).

1.1.1

Supernovae

A supernova is a cataclysmic object raising suddenly its luminosity and ejecting an extremely large amount of material into space. It can be a late massive star or a component of a binary star. Spectra of these objects are different, therefore it is a basement for the supernova classification.

Supernova type I: spectra do not exhibit hydrogen lines. These types are divided into three sub-categories: a, b, and c. Ia–strong SiII line (6150 Å), b and c: presence (Ib) or absence (Ic) of strong helium lines. Supernova type II: spectra contain strong hydrogen lines.

There are several possible scenarios of supernova explosion. Stars more massive than $8M_\odot$ can burn helium, and heavier elements, such as carbon, nitrogen, oxygen, and silicon. Inside those stars,

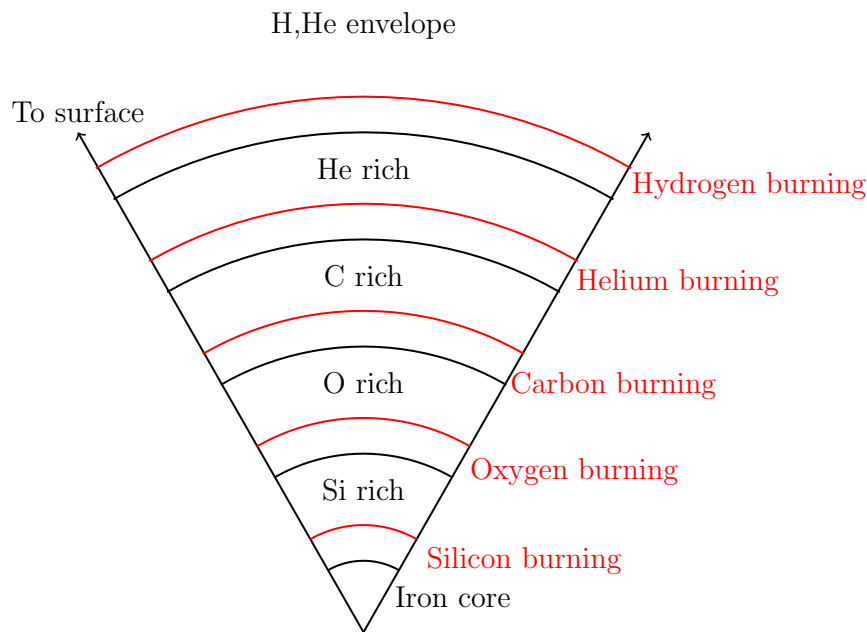


Figure 1.2: An onion structure interior of a massive star a few days before a collapse. Red-labeled regions with burning material are between inert zones with no thermonuclear reactions. Copied from [Carroll & Ostlie \(2017\)](#).

an onion structure is modeled, see Fig. 1.2. The main reason of this structure is a sensitivity of the nuclear reactions to temperature and mass density. The conditions are sufficient enough only in thin zones, which leads to the creation of an inert zone upwards to the burning zone. The iron core is located in the centre. Iron cannot be burned whilst energy is released, although photodisintegration processes occur. The core collapses when the density exceeds $\rho \approx 8 \times 10^{17} \text{ kg} \cdot \text{m}^3$. The strong attractive force changes to strong repulsive force, the shock wave is triggered propagating outwards. The shock drives the envelope expansion. The total energy in the expanding material is released $\sim 10^{44} \text{ J}$; when the ejecta becomes optically thin, the radiation with power about 10^{36} W ($10^9 L_{\odot}$) (comparable with the brightness of an entire galaxy).

The supernovae type Ib and Ic may correspond to Wolf-Rayet stars (types WN and WC). The supernovae Ia are usually exploding remnants in the binary systems. Those objects do not explode simultaneously; the explosion is triggered by an accretion of mass from an accompanying star, while temperature rises continuously, until thermonuclear reactions occur in the whole volume, then the gas degeneration is destroyed and gas expands explosively to the outer space. Over than half of carbon and oxygen are burned to iron. Another alternative is a merge of two white dwarfs.

1.1.2

Be stars

Be stars are main-sequence B-type stars, in which some of Balmer lines are in emission. This emission profile is called emission Be-type. This specific profile is caused by a disc in the equatorial plane of a star. Electron temperature is approximately 10000K, electron density 10^{12} cm^{-3} . The disc reaches a size of several stellar radii. The mechanism leading to the creation of a disc are not fully known by now. Several theories were proposed, such as the creation of a disc caused by the critical rotation of a star. Several studies, such as [Porter \(1997\)](#) derive velocities $\sim 70 - 80\%$ of critical velocity.

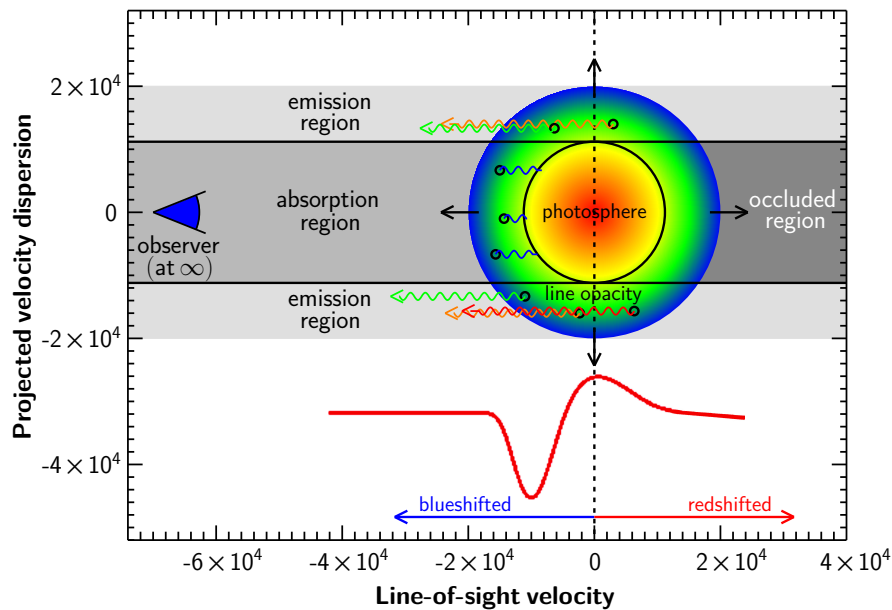


Figure 1.3: Scheme of P Cyg profile creation. Three zones affecting an observed spectrum are depicted here. Published with a kind agreement by the author of this picture (Petr Kurfürst).

Townsend et al. (2004) predict equatorial velocities of Be stars close to the critical velocity, because gravitation darkening may make velocity measurements more complicated.

Strong stellar winds are observed in the case of Be stars, then one theory of creating of circumstellar disc Bjorkman & Cassinelli (1993) is based on the stellar wind existence, nevertheless, this theory was found to be incorrect.

1.2

Observations of stellar winds

Stellar winds create characteristic spectral line profiles; which are called P Cygni, containing blue-shifted absorption component and red-shifted emission component. The basic concept of creation is drawn in Fig. 1.3, where three spatial zones with different contribution to observed spectra exist. The zones are marked with capital letters, zone by zone, an absorption affects spectra in the zone A. Re-emitted photons come mostly from the zone B, creating an emission profile, and finally, photons from the zone C do not affect spectrum for the observer, therefore, summing up spectra from the zone A and the zone B, we get a P Cyg profile. The line profiles are used for the mass-loss rate and the terminal velocity determination. Some profiles are more sensitive on these wind characteristics.

Lines including P Cygni profiles are listed in Tab. 1.2. Several atlases of P Cygni lines were published; namely Snow & Morton (1976), Snow et al. (1994). Lamers & Morton (1976) determined the first mass loss rates from P Cygni profiles. An example of UV spectra is shown in the Fig. 1.4 with important lines labeled.

Another important factor is the infrared and radio excess in the spectra. The excess is caused by free-free emission (Bremsstrahlung) in the wind. The mass loss rate can be calculated from the known velocity, and temperature structure. The radio excess was measured by Bieging et al. (1989), Leitherer et al. (1997, 1995); the infrared excess was measured by the IRAS satellite (e.g., Waters et al., 1987) or from the ground (Barlow & Cohen, 1977).

Ion	Abundance	Ion. pot eV	λ Å	Exc. pot. eV	g_1	f
CII	3.7×10^{-4}	11.26	1334.532	0.00	2	0.128
			1335.708	0.01	4	0.319
CIII	3.7×10^{-4}	24.38	1175.67	6.50	9	0.257
CIV	3.7×10^{-4}	47.89	1548.195	0.00	2	0.191
			1550.770	0.00	2	0.095
NIV	1.1×10^{-4}	47.45	1718.551	16.20	3	0.179
NV	1.1×10^{-4}	77.47	1238.821	0.00	2	0.157
			1242.804	0.00	2	0.078
OVI	6.8×10^{-4}	113.90	1031.928	0.00	2	0.130
			1037.619	0.00	2	0.065
MgII	3.5×10^{-4}	7.65	2795.528	0.00	2	0.612
			2802.705	0.00	2	0.305
SiII	3.5×10^{-5}	8.15	1526.707	0.00	2	0.230
			1533.431	0.00	2	0.229
SiIII	3.5×10^{-5}	16.35	1206.500	0.00	1	1.669
SiIV	3.5×10^{-5}	33.49	1393.755	0.00	2	0.514
			1402.770	0.00	2	0.255
FeII	2.5×10^{-5}	7.87	2585.876	0.00	10	0.065
			2598.370	0.05	8	0.099
			2599.396	0.00	10	0.224

Table 1.2: Some important lines which are used to measure mass loss rate. Copied from [Lamers & Cassinelli \(1999\)](#).

The very first models assumed the velocity and temperature structure of the wind to be spherically symmetric and the velocity magnitude was estimated to be [Chandrasekhar \(1934\)](#)

$$v(r) = v_{\infty} \left(1 - \frac{R_*}{r}\right)^{\beta}, \quad (1.1)$$

where v_{∞} is terminal velocity magnitude, R_* stellar radius and β is a free parameter. More general velocity profile takes a form

$$v(r) = v_{\infty} \left(1 - \frac{b}{r}\right)^{\beta}, \quad b = R_* \left[1 - \left(\frac{v_0}{v_{\infty}}\right)^{\frac{1}{\beta}}\right], \quad (1.2)$$

here, v_0 is the velocity magnitude at the lower boundary. The density profile can be derived from the continuity equation

$$\rho(r) = \frac{\dot{M}}{4\pi r^2 v(r)}. \quad (1.3)$$

Nowadays, this velocity model not used, since velocity structures are complicated non-symmetric and non-monotonic functions, however the mean values of velocity profile agree with the β velocity law.

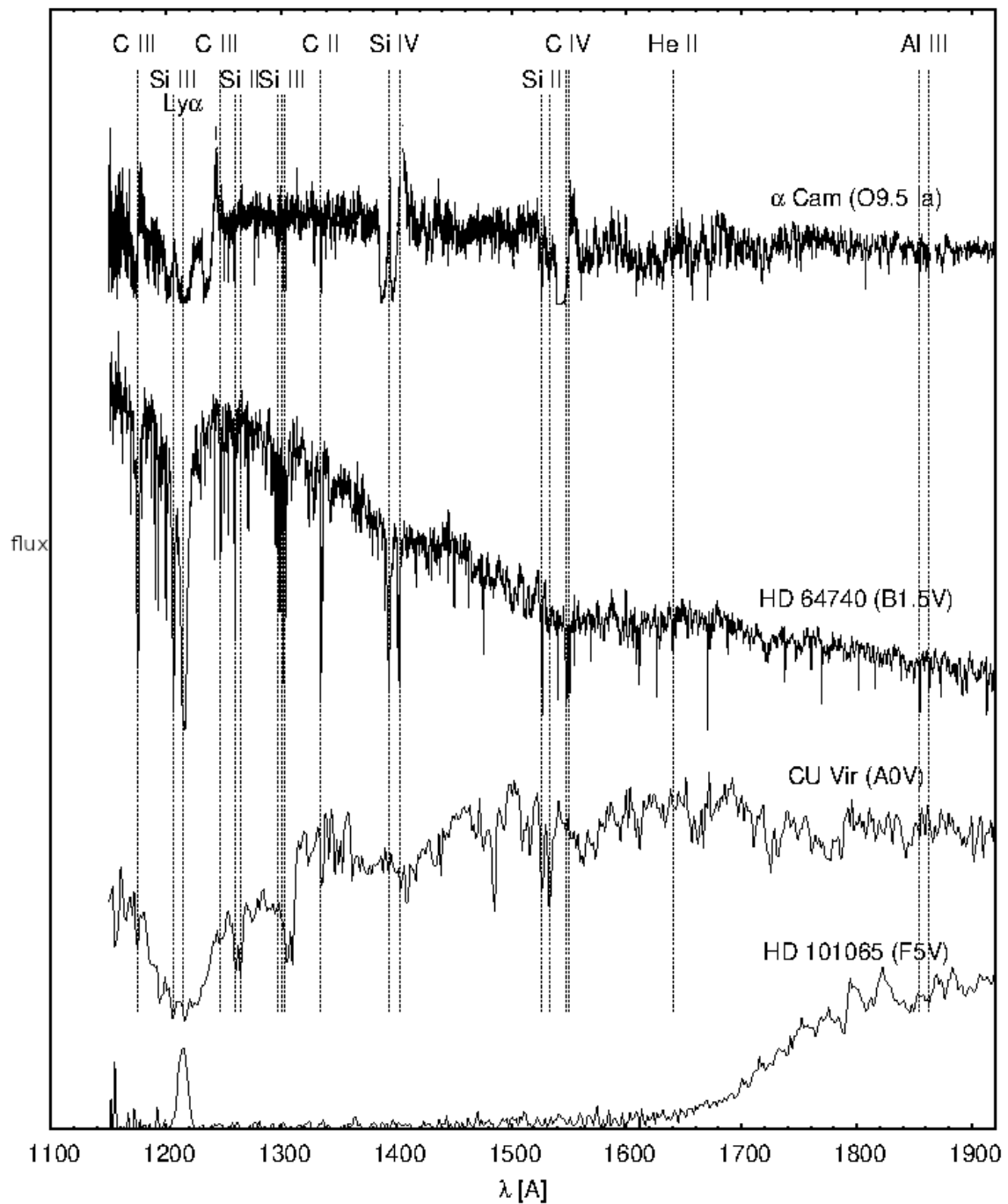


Figure 1.4: Measured spectra of several stars. Taken from [Krtićka \(2011\)](#).

1.3

The physics of stellar wind

The stellar wind phenomenon is originated via a strong radiative force in the case of massive stars. Hot stars possess the strong radiation field and the rotation is fast. Both of those effects decrease the escape velocity. We describe the basic physics in the following paragraphs.

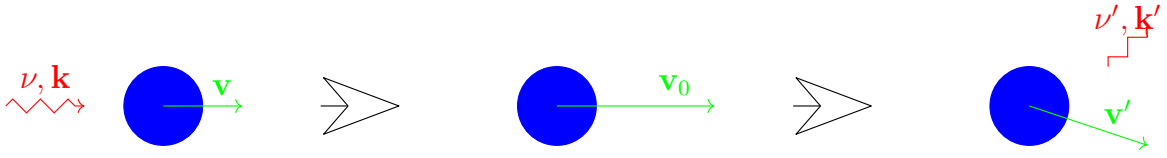


Figure 1.5: Scheme of a wind particle acceleration; a packet with frequency ν and wavevector \vec{k} is absorbed by an atom. The absorption is followed by emission of a new photon with a frequency ν' and wavevector \vec{k}' . Scheme adopted from [Krtička \(2011\)](#).

The basic physical phenomenon causing the acceleration is the momentum conservation. Radiation flux is coming mainly outward the stellar centre and photons interact with matter. Since the total momentum must be conserved, an atom is pushed outwards the photosphere and it can be excited. During deexcitation a pair atom-photon is produced again. The photon direction is random. Only if a new direction of photon is identical to its original direction, no acceleration happens. Otherwise, the velocity magnitude of the atom increases in comparison with its original velocity magnitude. The whole process is depicted in the Fig. 1.5.

In addition to radiatively driven winds, there are dust driven winds and winds accelerated by sound waves, moreover, each type plays a key role in a different type of star.

1.3.1

Hydrodynamical equations

Stellar wind is described by a set of equations: conservation of mass

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0. \quad (1.4)$$

Here, ρ is mass density and \vec{v} is a vector of drift velocity. The Euler equation takes the form

$$\frac{\partial(\rho \vec{v})}{\partial t} + \nabla \cdot (\rho \vec{v} \otimes \vec{v} + \mathcal{P}) = \vec{g} + \vec{f}_{\text{rad}}, \quad (1.5)$$

with gravity force \vec{g} and radiative force \vec{f}_{rad} on the right side, and the pressure tensor \mathcal{P} on the left side, \otimes is a symbol of tensor product. This equation is non-linear which causes strong dependence of solution the initial conditions in the liquid/gas dynamics. The last equation is conservation of energy

$$\frac{\partial e}{\partial t} + \nabla \cdot (e \vec{v} + p \vec{v}) = \vec{g} \cdot \vec{v} + \vec{f}_{\text{rad}} \cdot \vec{v} + \dot{q}. \quad (1.6)$$

Here e denotes energy density, p scalar pressure and \dot{q} represents the heating and cooling of gas by radiation. To close this set of equation, we need a dispersion relation between energy density and pressure.

1.3.2

Radiative force

The radiative force plays a crucial role in the stellar winds acceleration and it takes the form

$$\vec{f}_{\text{rad}} = \int_0^{\infty} d\nu \int_{\omega} d\Omega (\kappa(\nu) \cdot I(\nu, \vec{n}) - \eta(\nu)) \cdot \vec{n}, \quad (1.7)$$

where $\kappa(\nu)$ represents a total mass extinction (absorption and scattering) and $\eta(\nu)$ is emission coefficient, \vec{n} is a unit vector of a direction propagation of radiation. If we assume isotropic emissivity and opacity, the radiative acceleration can be written in the form

$$\vec{f}_{\text{rad}} = \frac{1}{c} \int_0^{\infty} d\nu \kappa(\nu, r) \cdot \vec{F}(\nu, r), \quad (1.8)$$

r is radial coordinate and F is the monochromatic flux

$$\vec{F}(\nu, r) = \oint d\Omega I(\nu, r, \vec{n}) \cdot \vec{n}. \quad (1.9)$$

The radiative force depends on the opacity. The opacity consists of two significant parts: continuum and line. Therefore, matter is accelerated in lines and in continuum.

The continuum opacity consists of Thomson scattering, free-free and bound-free absorption. We define the Eddington parameter: a ratio of electron scattering acceleration and gravitational acceleration

$$\Gamma = -\frac{g_{\text{Th}}}{g} = \frac{\frac{\sigma_{\text{Th}} L_*}{\mu_e}}{4\pi G M_* c}, \quad (1.10)$$

g_{Th} , g denotes acceleration due Thomson scattering, and gravitation acceleration respectively, σ_{Th} the Thomson scattering cross section, and G the gravitation constant, which is approximately constant in the stellar wind, therefore it cannot accelerate the wind, however, this force can lower the gravity. In the case of an O star, $\Gamma \lesssim 1$; the gravitation field is lowered significantly. The escape velocity for the ionized gas takes the form

$$v_{\text{esc}} = \sqrt{\frac{2GM_*(1-\Gamma)}{R_*}}, \quad (1.11)$$

here, r denotes a radial distance from the center. The radiative force calculation is very challenging, hence several approximation are used. We describe the Sobolev approximation and the CAK theory in the further sections.

Sobolev approximation

The main idea of this approximation is a simplification of interaction of photons with lines. The interaction area is reduced to a single point (Sobolev point), accordingly, line profiles are considered to be a delta function with no broadening. This approximation is valid in the cases of large velocity gradients. A calculation of optical depth is much more simple, since all quantities can be evaluated only in one point and no integration has to be processed. The Sobolev approximation is frequently used in the supernova ejecta modelling, since the assumption of large velocity gradients is well satisfied. A radiative force due optically thick lines depends only on the velocity gradient in the wind.

CAK theory

This theory was published by [Castor et al. \(1975\)](#), absorption in optical thick and thin lines is assumed. Two parameters are defined here: α – a ratio of radiative force due optically thick lines over the total radiative force, and k defining the magnitude of radiative force; the radiative force then takes a form

$$f_{\text{rad}} = \frac{f\Gamma GM_* L_*}{r^2} \left(\frac{1}{\sigma_{\text{Th}} \rho v_{\text{th}}} \frac{d\nu}{dr} \right)^\alpha. \quad (1.12)$$

In this equation, f is a correction factor due to finite size of a star, v_{th} mean velocity due thermal motion of wind particles; according to [Gayley \(1995\)](#) a parameter \bar{Q} can be introduced; then the equation takes the form

$$g_{\text{rad}} = \frac{f\bar{Q}L_*}{r^2} \left(\frac{1}{4\pi\rho\bar{Q}} \frac{dv}{dr} \right)^\alpha. \quad (1.13)$$

The equation of motion takes the form (non-dynamical atmosphere $\frac{\partial v}{\partial t} = 0$).

$$v \frac{dv}{dr} = -\frac{GM(1-\Gamma)}{r^2} + \frac{f\Gamma GM_* L_*}{r^2} \left(\frac{vr^2}{\dot{M}\bar{Q}} \frac{dv}{dr} \right)^\alpha. \quad (1.14)$$

Only one appropriate solution exists for a specific star, determining mass-loss rate and escape velocity.

1.4

Commonly used models of stellar winds

There is a wide range of commonly used models for the stellar wind modelling. The stellar wind model is not calculated using general physical laws, although, some simplifications are used routinely. Spherical symmetry (i. e. 1-D) is often assumed, using the NLTE approximation is necessary. The codes solve the radiative transfer equation (hereafter RTE) and the kinetic equilibrium equations (the NLTE line formation problem), in some cases supplemented by temperature-defining equation (either radiative equilibrium or thermal balance). The codes are used for analysis of observed spectra.

One-dimensional codes

The most simple codes assume spherical symmetry, such as CMFGEN ([Hillier, 1987, 1990, Hillier & Miller, 1998](#)) includes detailed RTE solving. The code is NLTE and account the line blanketing. The code is designed for W-R, O stars and Luminous Blue-Variabiles, see the code websites¹. Other codes are FASTWIND ([Santolaya-Rey et al., 1997, Puls et al., 2005, Puls et al., 2020](#)), and PoWR [Hamann & Gräfener \(2003, 2004\)](#), [Sander et al. \(2015\)](#). The METUJE code ([Krtićka & Kubát, 2004, 2009, Krtićka, 2006](#)) calculates the global model of atmosphere – includes both photosphere and wind and considers affection of wind onto the photosphere additionally. This code still assumes the spherical symmetry.

Is one-dimensional approach good enough?

One-dimensional codes seemed to be reliable for the stellar winds modelling, however, several works showed that one dimensional models are not capable to fully describe non-symmetric phenomena, such as stellar rotation ([Puls et al., 1993, Owocki & Cramner, 1994, Petrenz & Puls, 2000](#)), by the magnetic field ([ud Doula, 2014](#)), or by the accretion onto the compact object ([Blondin et al., 1990, Feldmeier et al., 1999](#)).

Furthermore other studies has shown natural instabilities originating from light-atoms interactions. The density, velocity field and other physical quantities are not simple monotonic functions: [Carlberg \(1980\)](#), [Owocki & Rybicki \(1984\)](#), [Owocki & Puls \(2002\)](#), [Feldmeier et al. \(1997\)](#), and

¹<http://kookaburra.phyast.pitt.edu/hillier/web/CMFGEN.htm>

many others. The instability was studied by radiative hydrodynamical simulations in 1-D (Feldmeier et al., 2003, Owocki et al., 1988, Runacres & Owocki, 2002) or in 2-D (e.g. Dessart & Owocki, 2003, 2005, Sundqvist et al., 2018, Driessen et al., 2021).

More than one-dimensional codes

As it was shown by 3-D calculations of line profiles in inhomogeneous winds, 1-D models can not describe the effects of wind clumping both on line profiles and mass-loss rate determinations properly (Sundqvist et al., 2010, 2011a, Šurlan et al., 2012, 2013).

A transition from 1-D to 3-D models is not straightforward; full 3-D model requires significantly larger computational sources, which is the main reason to calculate pseudo 2-D models Oskinova et al. (2004, 2006) or pseudo 3-D models Sundqvist et al. (2011b). Full radiation-hydrodynamics LTE codes, (such as Nordlund & Stein, 2009, Ludwig & Steffen, 2016, Freytag et al., 2019) are simulating convection in cool stars. Several steps to full NLTE 3-D have been already taken; since a general velocity field must be implemented, only simplified problems were solved. The codes differ by a way of solving radiative transfer: a finite volume method Adam (1990), Lobel & Blomme (2008), and Hennicker et al. (2018), short characteristics method Papkalla (1995), Korčáková & Kubát (2005), Georgiev et al. (2006), Zsargó et al. (2006), Leenaarts & Carlsson (2009), Ibgui et al. (2013), Štěpán & Trujillo Bueno (2013), and Hennicker et al. (2020). Every code from the list requires large computational resources to solve even the simplest line formation problem.

There are several codes for the MC 3-D radiative transfer calculations. HDUST (Carciofi et al. 2004, Carciofi & Bjorkman 2006, Carciofi et al. 2017) a code for calculation of radiative transfer for a wide range of objects, like Be stars, hot star winds, etc. The PYTHON code (Long & Knigge, 2002), extended by (Higginbottom et al., 2013). PYTHON is a code for spectra calculation. It can calculate with several astrophysical objects including analytical wind models. The Python code using cylindrical or spherical grid but it also can read in arbitrary geometries. The source code is available online here ². (Šurlan et al., 2012, 2013) studied the clumping phenomena in the hot star winds. The radiative transfer is solved in doublet resonance lines.

Radiative transfer through the supernova ejecta is treated with Monte Carlo as well. Tardis (Kerzendorf & Sim, 2014, Vogl et al., 2019, Kerzendorf et al., 2019) is a supernova code for the quick calculation of spectra. The code is written in C and Python languages. The procedures written in the C language provide fast calculations whereas the Python part guarantee a user friendly working environment. The parameters of the calculation are stored in a yml file and can be easily edited. User can modify a wide number of conditions like the lower boundary condition, ejecta physical parameters, applied physical approximations, numerical parameters, etc. More sophisticated codes, such as (Lucy, 2005, Kromer, 2009, Kromer & Sim, 2009), simulate radioactive reactions in the ejecta.

²<https://github.com/agnwinds/python>.

CHAPTER 2

NLTE Monte Carlo radiative transfer

Many methods to solve the radiative transfer equation were already developed. Classical codes solve the radiative transfer equation as a classical partial differential equation. Another approach was developed later – solution of radiative transfer equation in the probabilistic interpretation which is described in this chapter.

In the nature, the radiation field is quantized into photons. Every photon can interact with an atom and these interactions lead to non-linear coupling between radiation and matter. It is not possible to quantize the radiation field to physical photons, hence, in the Monte Carlo approach, numerical quanta are created. The quanta are denoted as packets and each packet represents an energy flow, it is indivisible and possesses a weight (importance), strictly constant during the whole propagation procedure, which determines a contribution of the packet to the results (spectra). It may be understood as an energy of the packet. The matter can be introduced in several approaches. We implemented Lucy's Macro atom approach: matter is divided into finite volume elements and the properties correspond to the properties of real matter in a limit.

This Chapter begins with the Monte Carlo method description, then a 'classical' form of the radiative transfer equation is introduced in the Section 2.2. Then, a probabilistic form of radiative transfer equation is derived, the radiative packets are introduced and described: the creation, the propagation, and the spectra calculation. The Section 2.5 redefines the statistical equilibrium equation in the case of finite volumes, moreover, this approach is embraced in the final sections – Macro-atom approach is introduced and all implemented rates are listed.

2.1

Monte Carlo method

The Monte Carlo method was used for a simulation of the neutron transport by Nicholas Metropolis and his colleagues, see [Metropolis \(1985\)](#). The basic inspiration was the game of chance, which is always played at the heart of Monte Carlo calculations ([Noebauer & Sim, 2019](#), see, Chap. 3). The heart of Monte Carlo simulations is a random number generator. A large amount of pseudo-random¹ number generators is known. Usually an initial seed is used; a different seed generate a different series of random numbers. The true random number generators can generate numbers based on a

¹Pseudo-random generator generates a sequence of numbers which distribution is approximately random.

measured physical quantity changing in the time. If we want to create a regular distribution in an interval $[a, b]$, $b > a$, we calculate

$$c = a + (b - a) \cdot z, \quad z \in [0, 1]. \quad (2.1)$$

We show two standard methods of random sampling of variables based on a distribution function. The first one is based on analytical function $f(x)$:

$$z = \int_y^b dx f(x), \quad (2.2)$$

where $f(x)$ is a non-negative, upper-bounded function, with

$$\lim_{x \rightarrow \pm\infty} f(x) = 0, \quad (2.3)$$

the probability of choosing a value within an interval dx is then equal to

$$w(x, x + dx) = \frac{f(x)dx}{\int_{-\infty}^{+\infty} dx f(x)}. \quad (2.4)$$

Firstly $z \in [0, 1]$ is randomly generated, secondly, the lower boundary y is determined. This method is used less frequently than the following method, simply, because we usually work with non-analytical functions.

The second method is named the acceptance-rejection method, see Fig. 2.1. The distribution function is defined by a table of values, or by a non-integrable function. Firstly, the definition of intervals on both axis is necessary and the basic rectangle is defined. Secondly, two random numbers, x_{rand} and y_{rand} are generated. A value of function $f(x_{\text{rand}})$ is determined. Thirdly two cases are possible

- acceptance if $y_{\text{rand}} \leq f(x_{\text{rand}})$,
- rejection if $y_{\text{rand}} > f(x_{\text{rand}})$.

If a value is accepted, we can use this value in forthcoming calculations; concretely, the calculated value can be a packet frequency. The resulting distribution is in the right panel of Fig. 2.1. Rejection means a new generation of two numbers, which will be accepted or rejected. This method can be used for a numerical integration, such as a calculation of π , which is the first Monte Carlo program, which every student probably writes. The scheme of a calculation is plotted in Fig. 2.2. Let the total number of generated points be N , the number of points under the curve (accepted points) is equal N_+ . The area of a circle is equal to $S = \pi r^2$; if $r = 1$, then $S = \pi$. Since a symmetry can be applied, we get

$$\frac{\pi}{4} = \frac{N_+}{N}. \quad (2.5)$$

This calculation can give a representative image of accuracy of the Monte Carlo method; the number of points must be very large to achieve an acceptable precision.

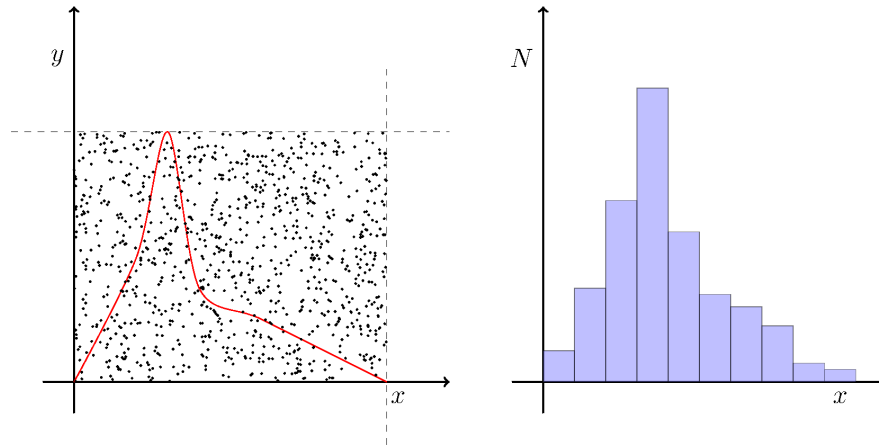


Figure 2.1: Acceptance-rejection method. *Left panel:* The red line represents the distribution function $f(x)$ and random points ($N = 1000$) are sampled in the bounded area. This construction can be used both for calculation of an area below the curve and for generation of random x values with the distribution $f(x)$. *Right panel:* a histogram of generated values. Random points were generated with the implicit Tikz random number generator.

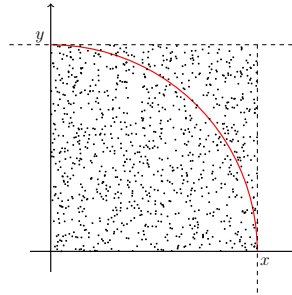


Figure 2.2: Scheme of numerical calculation to calculate a value of π .

2.2

Radiative transfer equation

The radiation is an oscillating electromagnetic field – given by the electric field $\vec{E}(\vec{r}, t)$ and the magnetic field $\vec{B}(\vec{r}, t)$. This idea was firstly published by James Clerk Maxwell. One can find the calculation in every classical electromagnetic field theory book, for example [Griffiths \(1999\)](#). Making simple calculations, we can derive the following physical quantities for every propagating light ray, which is characterized by specific intensity, which can be defined as energy flowing through an area dS

$$\delta\epsilon = I(\nu, \vec{n}) \cdot \vec{n} \cdot d\vec{S} \cdot d\Omega \cdot d\nu \cdot dt, \quad (2.6)$$

where \vec{n} is an arbitrary unit vector, $d\vec{S}$ is an element of an arbitrary area (oriented), $d\Omega$ is an angle element, $d\nu$ is a frequency element and dt is a time element. Analogically, we can define a decrease of radiation energy by the extinction coefficient

$$\delta\epsilon = \chi(\nu, \vec{n}) \cdot I(\nu, \vec{n}) \cdot dS \cdot dl \cdot d\Omega \cdot d\nu \cdot dt, \quad (2.7)$$

and an increase of energy by the emission coefficient

$$\delta\epsilon = \eta(\nu, \vec{n}) \cdot dS \cdot dl \cdot d\Omega \cdot d\nu \cdot dt, \quad (2.8)$$

dl is an element of length.

The light propagation through matter is a basic physical problem in the stellar atmospheres modelling. The equation describing the light propagation is called the radiative transfer equation, it is written in the form, (see [Hubeny & Mihalas, 2015](#), Eq. (11.21))

$$\frac{\partial I(\vec{n}, \nu, \vec{r}, t)}{\partial s} = \eta(\vec{n}, \nu, \vec{r}, t) - \chi(\vec{n}, \nu, \vec{r}, t) \cdot I(\vec{n}, \nu, \vec{r}, t). \quad (2.9)$$

Here, the left-side derivative is a derivative on a path with a parameter s , the right-side quantities: η is emissivity, and χ is opacity. Opacity and emissivity characterize the interaction of the matter with electromagnetic radiation. Equation (2.9) describes an intensity change on a path ds . This intensity is decreased by opacity and increased by emissivity.

We can define a quantity optical depth between two points \vec{r}_1 and \vec{r}_2 on a path \mathcal{C}

$$\tau = \int_{\mathcal{C}} ds \chi(\vec{n}, \nu, \vec{r}, t), \quad \mathcal{C} = \{x(s), y(s), z(s); t \in [s_1, s_2]\}. \quad (2.10)$$

Here, the \mathcal{C} denotes a curve parametrized by a parameter s . It expresses a transparency of an environment. If $\tau < 1$, then the probability of interaction (scattering, etc.) is very small. This environment is denoted as optically thin. In another case, $\tau > 1$ describes optically thick environment and the most of photons interact. The optical depth $\tau = 1$ corresponds to average free photonic path which is equal to

$$\langle l \rangle = \frac{1}{\chi(\vec{n}, \nu, \vec{r}, t)}. \quad (2.11)$$

One can imagine optically thin environment as a sunny day with a great visibility; optically thick environment corresponds to a foggy weather – the main free photonic path is much smaller than it is needed for discovering of a surrounding environment.

If we put (2.10) into (2.9), then we get

$$\frac{\partial I(\vec{n}, \nu, \vec{r}, t)}{\partial s} = \frac{\partial \tau}{\partial s} \frac{\partial I(\vec{n}, \nu, \vec{r}, t)}{\partial \tau} = \chi(\vec{n}, \nu, \vec{r}, t) \frac{\partial I(\vec{n}, \nu, \vec{r}, t)}{\partial \tau} = \eta(\vec{n}, \nu, \vec{r}, t) - \chi(\vec{r}, \nu, \vec{r}, t) \cdot I(\vec{n}, \nu, \vec{r}, t), \quad (2.12)$$

after dividing by $\chi(\vec{n}, \nu, \vec{r}, t)$,

$$\frac{\partial I(\vec{n}, \nu, \vec{r}, t)}{\partial \tau} = \frac{\eta(\vec{n}, \nu, \vec{r}, t)}{\chi(\vec{n}, \nu, \vec{r}, t)} - I(\vec{n}, \nu, \vec{r}, t), \quad (2.13)$$

we can now define the source function

$$S(\vec{n}, \nu) = \frac{\eta(\vec{n}, \nu, \vec{r}, t)}{\chi(\vec{n}, \nu, \vec{r}, t)}, \quad (2.14)$$

and put this definition into (2.13)

$$\frac{\partial I(\vec{n}, \nu, \vec{r}, t)}{\partial \tau} = S(\vec{n}, \nu, \vec{r}, t) - I(\vec{n}, \nu, \vec{r}, t). \quad (2.15)$$

2.3

Probabilistic interpretation of the radiative transfer equation

The standard radiative transfer equation includes the specific intensity which is a statistical quantity. If we want to calculate a propagation of a single photon, we have to switch to the probabilistic terms.

Let us assume an environment with pure absorption only, consequently, without emission ($\eta = 0$). Therefore, the equation (2.15) takes the form

$$\frac{\partial I(\vec{n}, \nu, \vec{r}, t)}{\partial \tau} = -I(\vec{n}, \nu, \vec{r}, t), \quad (2.16)$$

which is a differential equation of the first order. Using a standard method ‘separation of variables’, the intensity can be expressed in a form

$$I(\tau) = I(0) \exp(-\Delta\tau). \quad (2.17)$$

The probability that the photon is not absorbed in optical depth $\Delta\tau$ is

$$w(\Delta\tau) = \exp(-\Delta\tau).$$

The inverse function is then

$$\Delta\tau = -\ln(w(\Delta\tau)), \quad (2.18)$$

In this equation, $w(\Delta\tau) \in (0, 1)$ is a random number. It is important for calculation of the probability interaction during the propagation of packets, it can be found in the forthcoming text.

2.4

Radiative packets

The radiative packets represent a numerical quantization of the radiation field. The real procedure of propagation of photons depends on time, since light propagates with a finite velocity. The simulation can be done for a single time snapshot. Furthermore, packets can be propagated one by one, which is the great advantage of MC simulation in the case of parallelization. The packet life cycle is displayed in the Fig.2.3. The packets are created at the lower boundary, then the packets are propagated until:

1. it reaches lower boundary,
2. it reaches upper boundary,
3. a number of interactions $N_{\text{int}} > N_{\text{max}}$ (N_{max} is a variable defined by user),
4. an interaction occurs.

The first point leads to a packet deactivation, unless a ‘reflexive’ lower boundary condition is set; a new r-packet is re-emitted instead of the previous one. The second and the third items cause the packet deactivation. The interactions change the type of packet. The whole packet machinery is described in the following parts.

2.4.1

Creation of radiative packets

The packets are created according to lower boundary physical conditions valid within this area. At least, the total luminosity, and frequency distribution must be defined to create packets and set their properties. A standard boundary shape is assumed to be a sphere and it represents the stellar ‘surface’. A packet initial position is random, and it is

$$\vec{r}_{\text{init}} = (R \sin(\Theta) \cdot \cos(\Phi), R \sin(\Theta) \cdot \sin(\Phi), R \cos(\Theta)). \quad (2.19a)$$

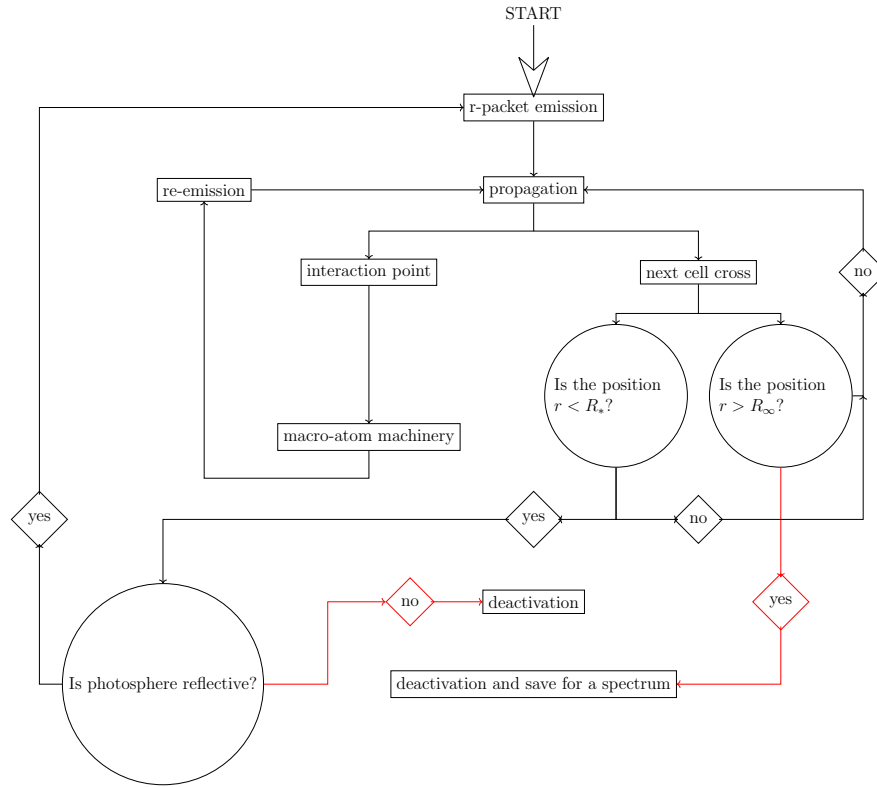


Figure 2.3: Scheme describing a life cycle of a packet. The packet is created (the most upper panel) and propagated through the atmosphere. The propagation distance can be either to the point of interaction, or to the next crossing with the grid. If the next cross with grid is chosen, a new position is tested for to be inside the computational domain ($R_* < r < R_\infty$). If this inequality is not satisfied, the packet is deactivated. The left inequality means packet is inside the photosphere. If the photosphere is reflexive, then the packet is emitted again, otherwise the packet is deactivated. In the case of $r > R_\infty$, the packet is deactivated and can be used to create a spectrum. If the packet is inside the computation domain, the propagation can continue. If an interaction is chosen, the packet is propagated to the point of interaction, the MA machinery is processed until the packet is transformed to r-packet (not necessary with the same frequency) and the propagation can continue.

The angles are calculated by the equations

$$\Phi = 2\pi\tau, \quad (2.19b)$$

$$\cos(\Theta) = 2\sigma - 1. \quad (2.19c)$$

Here τ and σ are randomly generated numbers in the interval $[0, 1]$ (see also [Noebauer & Sim, 2019](#), Eqs. (26) and (27)). The propagation direction of the r-packet is randomly chosen in the outwardly direction from the stellar surface, hence another random unit vector \vec{n}_{init} is generated as

$$\vec{n}_{\text{init}} = (\sin(t) \cdot \cos(p), \sin(t) \cdot \sin(p), \cos(t)), \quad (2.20a)$$

with

$$p = 2\pi\tau', \quad (2.20b)$$

$$\sin^2(t) = 1 - \sigma'. \quad (2.20c)$$

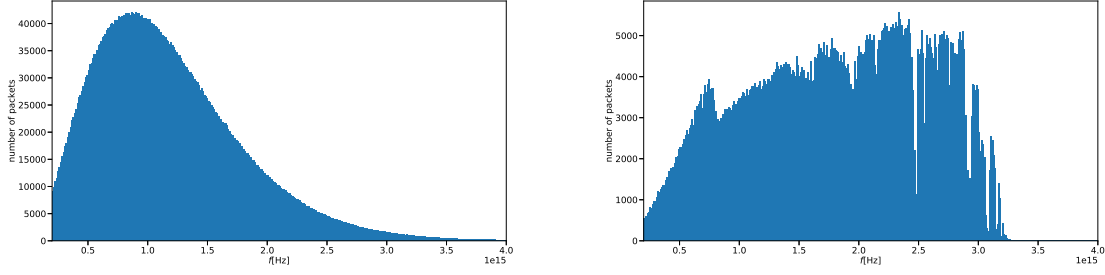


Figure 2.4: Initial packet frequency distributions. *Left panel:* generated with the Planck function. The total number of packets is 5×10^6 ; *right panel:* generated with the emergent spectrum, the total number of packets 5×10^6 . The input spectrum was calculated with the TLUSTY code (Hubeny, 1988).

Here τ' and σ' are randomly generated numbers in the interval $[0, 1]$ (see also Noebauer & Sim, 2019, Eq. (47)). The propagation direction in the stellar coordinate system is calculated as a double rotation (multiplication of two rotation matrices), then the initial packet direction \vec{d}_{pack} takes the form

$$\vec{d}_{\text{pack}} = (r_3 \cdot \sin(t) \cdot \cos(p) + r_1 \cdot \cos(t) \cdot \cos(p) - r_2 \cdot \sin(p), \\ r_3 \cdot \sin(t) \cdot \sin(p) + r_1 \cdot \cos(t) \cdot \sin(p) + r_2 \cdot \cos(p), r_3 \cdot \cos(t) - r_1 \cdot \sin(t)), \quad (2.21)$$

where r_i represent the Cartesian coordinates of the position vector (2.19a). This direction is corrected for aberration

$$\vec{n}_a = \frac{\vec{n} - \frac{\mathbf{v}}{c}}{1 - \frac{\vec{n}\mathbf{v}}{c}}. \quad (2.22)$$

The packet initial frequency follows the radiation distribution, emerging from the stellar surface which may be equal to the Planck distribution. More precisely, it can be the calculated synthetic spectrum of the stellar photosphere (see Fig. 2.4). The packets are created at the beginning of every iteration. Packets' energy differ from photonic energy emitted per unit time and it is defined

$$E = \frac{L_*}{N_{\text{packs}}}, \quad (2.23)$$

where L_* is the stellar (or an inner boundary in a general case) luminosity, and N_{packs} is a total number of created packets. It is clear that it does not depend on packet frequency. This definition is important because of energy conservation.

2.4.2

Propagation and interaction of radiative packets

Radiative packets represent radiation field propagating through the wind. Properties of packets depend on the reference frame. There are two main reference frames in the stellar wind. The first one is the rest frame, the frame connected with the center of the star. We denote physical quantities expressed in the rest frame (hereafter RF) using the index \sharp . The second one is the co-moving frame. The wind is 'frozen' in the origin of this frame, hence the relative velocity of this reference frame depends on position and time, in general. In this case, the physical quantities in co-moving frame (hereafter CMF) are denoted \flat .

During the propagation, a position of interaction must be determined. Interactions depend on frequency of a packet; a rest frame frequency of a packet is ν^\sharp at a position \vec{r} with density $\rho(\vec{r})$ and temperature $T(\vec{r})$. Therefore, CMF frequency is equal to

$$\nu^\flat = \nu^\sharp \left(1 - \frac{\vec{n} \cdot \vec{v}}{c} \right), \quad (2.24)$$

which is Doppler shift formula. In this equation, \vec{v} is the velocity vector of the matter (in the RF) at the given point \vec{r} , and \vec{n} is the direction unit vector of the light ray. The CMF frequency is changing during the propagation.

The point of line interaction has to be determined, then we get a resonance point and line and continuum optical depth can be calculated. Optical depth by continuum opacity is equal to:

$$\tau_{\text{cont}} = \int_{\vec{r}}^{\vec{r}_R} d\vec{r}' \cdot \chi_{\text{cont}}(\vec{r}'), \quad (2.25)$$

where \vec{r} is the packet position, \vec{r}_R is the position of the closest possible line interaction (resonance point) of the r-packet with matter and $\chi_{\text{cont}}(\vec{r}')$ is the continuum opacity which is equal to

$$\chi_{\text{cont}}(\vec{r}, \nu^\flat) = \chi^{\text{Th}}(\vec{r}) + \sum_{i=1}^{N_I} \chi_i^{\text{ff}}(\vec{r}, \nu^\flat) + \sum_{i=1}^{N_L} \chi_i^{\text{bf}}(\vec{r}, \nu^\flat). \quad (2.26)$$

The first term corresponds to Thomson scattering, the second one to free-free processes and the third one to bound-free processes. N_I is the total number of all atomic ions and N_L is the total number of energy levels of all ions.

The optical depth in a line for a transition from a level l to a level u integrated along a path s from 0 to s_0 is given by (see [Kromer, 2009](#), Eq. 4.15)

$$\tau_{\text{line}} = n_l \frac{B_{lu} h \nu}{4\pi} \int_{\nu^\flat(s=0)}^{\nu^\flat(s=s_0)} d\nu^\flat \left(1 - \frac{n_u g_l}{n_l g_u} \right) \phi(\nu^\flat) \frac{ds}{d\nu^\flat}, \quad (2.27)$$

where ϕ is the line profile, n_l, n_u are number densities of atoms at levels l (lower) and u (upper), respectively, g_l, g_u are corresponding statistical weights, and B_{lu} is the Einstein coefficient for a transition from a level l to a level u . If we assume the Sobolev approximation, this equation takes a form

$$\tau_{\text{line}} = n_l \frac{B_{lu} h \nu}{4\pi} \left(1 - \frac{n_u g_l}{n_l g_u} \right) \times \left(\frac{ds}{d\nu^\flat} \right)_{\nu_{lu}} \times \begin{cases} 0 & \nu_{lu} \notin [\nu^\flat(0), \nu^\flat(s_0)], \\ -1 & \nu_{lu} \in [\nu^\flat(0), \nu^\flat(s_0)]. \end{cases} \quad (2.28)$$

Here, ν_{lu} is the transition frequency, note that the derivative $(ds/d\nu^\flat)_{\nu_{lu}}$ is negative. In the case of spherically symmetric fields, the optical depth can be simplified into a form ([Castor 1974](#), see also [Noebauer & Sim 2015](#), Eq. 22) to

$$\tau_{\text{line}} = \frac{\chi c}{\nu_{lu}} \left[\mu^2 \frac{d\nu}{dr} + (1 - \mu^2) \frac{\nu}{r} \right]^{-1}, \quad (2.29)$$

where μ is angle cosine between \vec{n} and $\vec{v}(\vec{r})$, and

$$\chi = \frac{\pi e^2}{m_e c} f_{lu} n_l \left(1 - \frac{n_u g_l}{n_l g_u} \right), \quad (2.30)$$

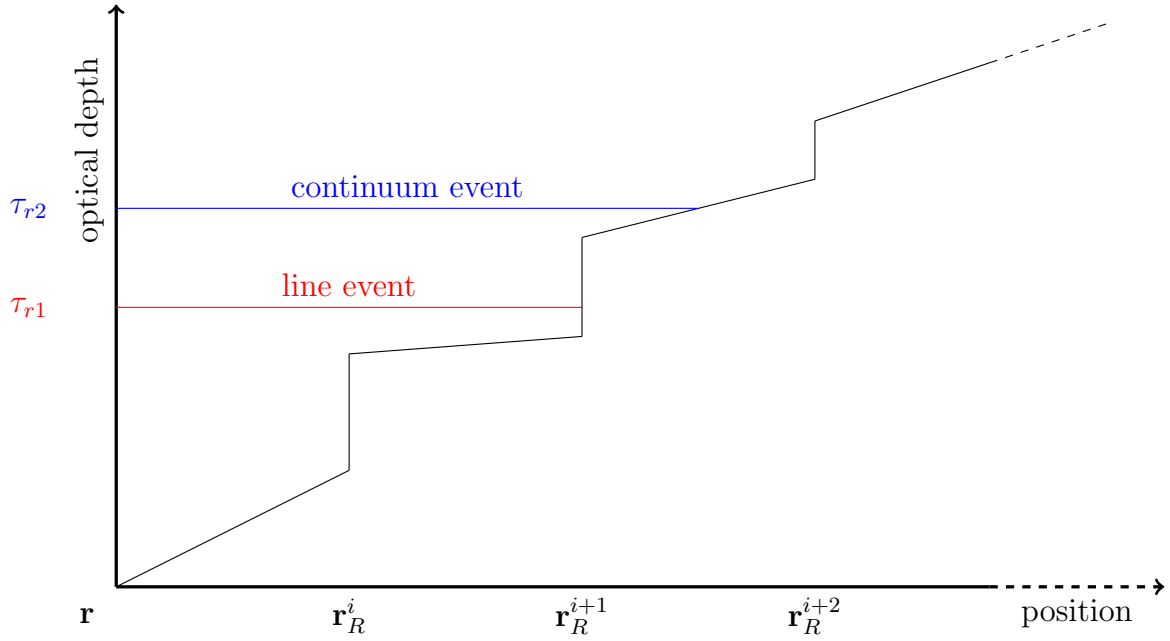


Figure 2.5: Method of selection of the next event from a randomly sampled optical depth. \bar{r}_R^* signs a resonance distances (in case of Sobolev approximation). The continuum optical depth grows constantly. Adopted from Kromer (2009).

according to Noebauer & Sim (2015, Eq. (7)). In this equation, f_{lu} is the oscillator strength for a transition from a level l to a level u .

In the Section 2.3, a probabilistic version of the radiative transfer equation was derived. According to Eq. (2.18), the packet is allowed to travel a random optical depth without an interaction (see, e.g. Whitney, 2011)

$$\tau_{\text{rand}} = -\ln(1-z) = -\ln(z'), \quad z, z' \in [0, 1], \quad (2.31)$$

where z is a random number. The generation of the random optical depth is followed by calculation of continuum optical depth τ_{cont} and line optical depth τ_{line} (Eq. 2.29), which are compared. The line and continuum optical depths are calculated by integration along the photon path. Line optical depth increases in Sobolev points only, whereas the continuum optical depth increases continuously.

In the case

$$\tau_{\text{rand}} > \tau_{\text{cont}} + \tau_{\text{line}}, \quad (2.32)$$

the photon continues its path, otherwise it interacts. The Sobolev approximation simplifies calculations; instead of integration over a path, local values in the Sobolev point can be used. According to (as in Kromer, 2009) the type of interaction: line, or continuum is chosen: the first optical depth which causes the invalidity of (2.32).

Obviously, the line and continuum interactions are treated differently. For a line interaction a specific transition must be chosen. Usually, only one transition in the specified frequency is possible. If there are more possible transitions, a line is chosen firstly, based on its optical depth:

$$w_i = \frac{\tau_i}{\sum_{j=1}^{N_{\text{lines}}} \tau_j}, \quad (2.33)$$

where N_{lines} is the number of lines with the same transition frequency. Once the transition is chosen, the macro-atom is activated; it means that the packet transmits into i-packet, which is after one,

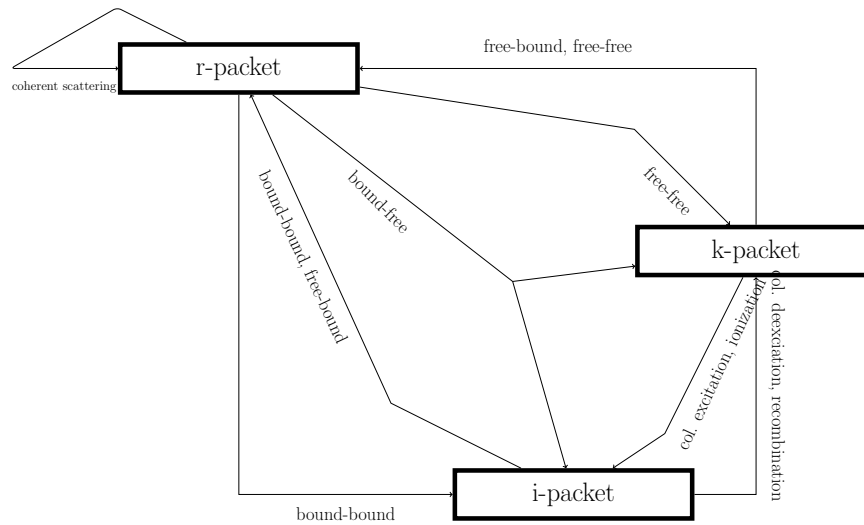


Figure 2.6: Scheme of the macro-atom interactions.

or more transitions deactivated. In the case of continuum interaction, the specific process must be determined, currently: bound-free, free-free transitions, and Thomson scattering can occur.

Thomson (and coherent) scattering causes reemission of a packet in a random direction (coherent scattering is currently assumed) with the same CMF frequency as of the original packet. The CMF quantities are recalculated to corresponding RF values. Free-free processes transform an r-packet into a k-packet and vice versa. Bound-free processes can transform r-packet to i-packet, or k-packet. Following Lucy (2003, Eq. (27)), the fraction of packets v_i/v^* ($v_i < v^*$) converts to an i-packet, where v_i is the ionization edge frequency. For each r-packet we decide (again using random numbers) whether it changes to a k-packet or to an i-packet. The described transitions are shown in Fig. 2.6. The Macro-atom mode is described in detail in the forthcoming sections.

2.4.3

Calculation of emergent spectra

Spectra are calculated due to the propagated packets through the stellar wind. Packets are stored in text files containing all important informations: the packet state (deactivated by crossing the outer boundary, destruction in photosphere, a numerical destruction because too many interactions), RF frequency and energy. Frequencies are transformed to wavelengths, since values of wavelengths are more ‘user friendly’ and memorable than the frequency values.

A spectrum is generated this way: initially, a wavelength interval ($\lambda_{\min}, \lambda_{\max}$) is defined (by user). The wavelength grid is set up. The number of intervals (bins) is defined and based on this integer the interval length is calculated

$$\Delta\lambda = \frac{\lambda_{\max} - \lambda_{\min}}{N_{\text{bin}}}, \quad (2.34)$$

where N_{bin} is the total number of bins. Calculated spectrum is a histogram technically. The boundary wavelengths of intervals are

$$\lambda_i = \lambda_{\min} + (i - 1) \cdot \Delta\lambda, \quad (2.35)$$

here, i is an integer number corresponding to the order of an interval. When the computation domain is set up, we can start creating the spectrum. We read informations packet by packet from the file.

If a packet is escaped, then we read its frequency (ν) and its energy (E). If wavelength ($\lambda = c/\nu$) satisfies the condition $\lambda_{\min} < \lambda < \lambda_{\max}$ the number of a wavelength bin is calculated

$$i = \left\lfloor \frac{\lambda - \lambda_{\min}}{\Delta\lambda} + 1 \right\rfloor. \quad (2.36)$$

An energy contribution due this packet is calculated using the equation

$$\Delta E = \frac{E}{\Delta\lambda} \frac{1}{4\pi r^2}, \quad (2.37)$$

where r denotes a distance of a detector to the central star.

The previous description is the most efficient if a spherical symmetry is assumed, hence all packets can be used to create a spectrum. If we want a spectrum measured in a specific direction, then a packet position and direction must be stored; therefore, the spectrum is more realistic. However, only a small fraction of propagated packets is used, which requires enormous number of propagated packets, at least 10^9 ; more sophisticated methods have to be used, such as peel-off method (Yusef-Zadeh et al., 1984, Wood & Reynolds, 1999, Baes et al., 2011, Steinacker et al., 2013, Lee et al., 2017), in the cases of fast mass outflows is this method referred as virtual packet scheme (Knigge et al., 1995, Long & Knigge, 2002, Kerzendorf & Sim, 2014, Bulla et al., 2015).

2.5

Macro-atom transition probabilities

We present a derivation a statistical equilibrium equation (SEE) in the ‘finite volume representation’, see Lucy (2002). Let us assume a matter with absorption and emission of radiative energy per unit volume and unit time with

$$a_i^R = n_l R_{li} \varepsilon_{il} = \sum_{l=1}^{i-1} n_l R_{li} (\varepsilon_i - \varepsilon_l), \quad (2.38a)$$

$$e_i^R = n_l R_{li} \varepsilon_{il} = \sum_{l=i+1}^{\mathcal{N}_i} n_l R_{li} (\varepsilon_i - \varepsilon_l), \quad (2.38b)$$

where R_{li} is a radiative rate for transition $l \rightarrow i$, ε is the excitation plus ionization energy of the given energy level, i denotes the current energy level, and l the lower energy level. The corresponding rates for the kinetic energy take form

$$a_i^C = n_l C_{li} \varepsilon_{il} = \sum_{l=1}^{i-1} n_l C_{li} (\varepsilon_i - \varepsilon_l), \quad (2.38c)$$

$$e_i^C = n_l C_{li} \varepsilon_{il} = \sum_{l=i+1}^{\mathcal{N}_i} n_l C_{li} (\varepsilon_i - \varepsilon_l). \quad (2.38d)$$

Then, the total amount of absorbed and emitted energy is then equal to

$$\begin{aligned} a_i^R + a_i^C - e_i^R - e_i^C &= \sum_{l=1}^{i-1} n_l R_{li} \varepsilon_{il} + \sum_{l=1}^{i-1} n_l C_{li} \varepsilon_{il} - \sum_{l=1}^{i-1} n_i R_{il} \varepsilon_{il} + \sum_{l=1}^{i-1} n_i C_{il} \varepsilon_{il} \\ &= \sum_{l=1}^{i-1} (n_l R_{li} + n_l C_{li}) \varepsilon_{il} - \sum_{l=1}^{i-1} (n_i R_{il} + n_i C_{il}) \varepsilon_{il} := \sum_{l=1}^{i-1} (\mathcal{R}_{li} - \mathcal{R}_{il}) (\varepsilon_i - \varepsilon_l), \end{aligned} \quad (2.39)$$

where we defined the total rate from/to lower levels for level i \mathcal{R}

$$\mathcal{R}_{il} = n_i R_{il} + n_i C_{il}. \quad (2.40)$$

The statistical equilibrium equation for the level i takes the form

$$\sum_{l=1}^{i-1} (\mathcal{R}_{li} - \mathcal{R}_{il}) + \sum_{u=i+1}^{\mathcal{N}_i} (\mathcal{R}_{ui} - \mathcal{R}_{iu}) = 0. \quad (2.41)$$

We can eliminate the sum containing $(\mathcal{R}_{li} - \mathcal{R}_{il})\epsilon_i$, which leads to

$$\sum_{l=1}^{i-1} (\mathcal{R}_{li} - \mathcal{R}_{il})\epsilon_i = a_i^R + a_i^C - \epsilon_i^R - \epsilon_i^C + \sum_{l=1}^{i-1} (\mathcal{R}_{li} - \mathcal{R}_{il})\epsilon_l.$$

Then

$$- \sum_{u=i+1}^{\mathcal{N}_i} (\mathcal{R}_{ui} - \mathcal{R}_{iu})\epsilon_i = a_i^R + a_i^C - \epsilon_i^R - \epsilon_i^C + \sum_{l=1}^{i-1} (\mathcal{R}_{li} - \mathcal{R}_{il})\epsilon_l,$$

while we finally get

$$e_i^R + e_i^C + \sum_{u=i+1}^{\mathcal{N}_i} \mathcal{R}_{iu}\epsilon_i + \sum_{l=1}^{i-1} \mathcal{R}_{il}\epsilon_l = a_i^R + a_i^C + \sum_{l=1}^{i-1} \mathcal{R}_{li}\epsilon_l + \sum_{u=i+1}^{\mathcal{N}_i} \mathcal{R}_{ui}\epsilon_i. \quad (2.42)$$

The equation (2.41) represents the standard form of the equation of the statistical equilibrium. It relates to nature's quantization of radiation into photons and matter into atoms. However, the equation (2.42) corresponds to energy flow in a finite volume element. It is an equivalent representation of the SEE. The flows can be quantized into indivisible e-packets. The Macro-atom (MA) approach is derived from this quantization of matter.

The Macro-atom approach is constructed to simulate the processes of interaction between radiation and matter. As described in [Noebauer & Sim \(2019\)](#) energy flows among different states approximate real matter dynamics. The whole machinery is not strictly limited by a single possible implementation; in our code, we are following the implementation of [Lucy \(2002, 2003\)](#).

Energy flow is represented by the indivisible energy packets, thus a packet cannot be divided into two, or more packets. However, the real processes do not conserve the number of packets. The fluorescence cascade is a process when an excited atom deexcites several times, while lowering its energy sequentially. Let us consider a three level atom. An atom is excited into the state three and this excitation is followed by two deexcitations, while two photons are created. In the macro-atom machinery, this energy redistribution flow should be implemented as well. Those transitions are processed via internal, non-radiative transitions. Concretely, energy packet connected with the initial frequency is allowed to undergo first transition virtually, the second transition radiatively.

2.5.1

Internal energy packets

Internal energy packets (i-packets) represent excitation energy in excited atoms. The internal packet state is characterized with the element number, ion number and level number. The ion and level numbers can change during the activation via the internal processes. To describe particular processes changing excitation and ionization state, we use following index notation: i – the current level, l – a lower level, u – an upper level, m – a level in a lower ionization state than the level i , and p an upper level ionization state. The states can be specified via the lower index (E – element, I – ion, L – level)

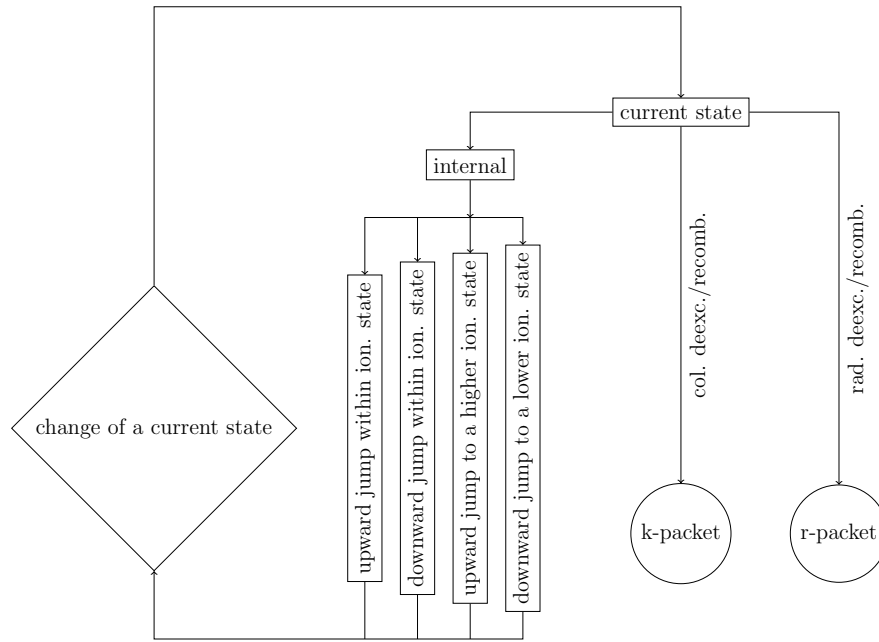


Figure 2.7: Scheme of the i-packet machinery. The internal processes cause change of the current state. Radiative or collisional deexcitation and recombination changes i-packet to r-packet or k-packet, respectively.

and its lower index classifies its concrete value, for example i_{kL_0} represents the basic level of the ion k (the state is denoted as i).

The total energy loss (left hand side of Eq. 2.42) caused by transitions from the energy level i as (see Kromer, 2009, Eq. 4.38)

$$e_i^{\text{tot}} = e_i^r + e_i^c + e_i^{\text{int, down}} + e_i^{\text{int, up}}, \quad (2.43)$$

where e_i^r and e_i^c denote radiative and collisional deactivation (introduced in Eq. 2.38b), respectively, and $e_i^{\text{int, up}}$ and $e_i^{\text{int, down}}$ mean internal upward and downward jumps, respectively. The rate e_i is a sum over all possible transitions from the state i within the current process; the transition rate is denoted p_i^{type} .

The basic scheme of i-packet machinery is shown in Fig. 2.7. This scheme is run until a non-internal process happens. Internal process can change both: ion and level index, concretely: internal downward jumps and upward jumps within an ionization state (change only level), internal downward and upward jumps changes also ionization state. Internal rates are calculated with both, radiative and collisional rates. Deactivation of a packet can be done via radiative deexcitation, or recombination. Furthermore, the energy can be transformed into kinetic energy via collisional deexcitation, or recombination. The rates for each transition are, namely: ²

- radiative deexcitation from the state i to the state l – the probability of this process is equal to

$$p_{il}^R = n_i R_{il}^S (\varepsilon_i - \varepsilon_l) / e_i^{\text{tot}}, \quad (2.44a)$$

where R_{il}^S is a rate in the Sobolev approximation, denoted with an upper index S ,

- radiative recombination (considered only from an ion ground level)

$$p_{im}^R = n_i R_{im} (\varepsilon_{i_{L_0}} - \varepsilon_m) / e_i^{\text{tot}}, \quad (2.44b)$$

²Detailed expressions for radiative (R) and collisional (C) rates can be found in the Appendix A.

R_{im} is the rate for recombination (A.11), the index i_{L_0} means the lowest level of the ion,

- collisional deexcitation – in this case

$$p_{il}^C = n_i C_{il} (\varepsilon_i - \varepsilon_l) / e_i^{\text{tot}}, \quad (2.44c)$$

C_{il} is the collisional deexcitation rate (see Section A.2),

- collisional recombination (considered only from an ion ground level)

$$p_{im}^C = n_i C_{im} (\varepsilon_{i_{L_0}} - \varepsilon_m) / e_i^{\text{tot}}, \quad (2.44d)$$

C_{im} is the collisional recombination rate (see Section A.2).

The internal processes (which do not deactivate the macro-atom) taken into account include

- internal downward jump within the current ionization state

$$p_{il}^{\text{int, down}} = n_i (R_{il}^S + C_{il}) \varepsilon_l / e_i^{\text{tot}}, \quad (2.45a)$$

- internal downward jump to the lower ionization state

$$p_{im}^{\text{int, down}} = n_i (R_{im} + C_{im}) \varepsilon_m / e_i^{\text{tot}}, \quad (2.45b)$$

- internal upward jump within the current ionization state

$$p_{iu}^{\text{int, up}} = n_i (R_{iu}^S + C_{iu}) \varepsilon_i / e_i^{\text{tot}}, \quad (2.45c)$$

where R_{iu}^S is the rate of transition to the upper level,

- internal upward jump to a higher ionization state

$$p_{ip}^{\text{int, up}} = n_i (R_{ip} + C_{ip}) \varepsilon_i / e_i^{\text{tot}}. \quad (2.45d)$$

Equations (2.44) and (2.45) contain rates which can be corrected for the stimulated emission. If we want to correct the rates for the stimulated emission, we replace rates with Eqs. (A.8) and (A.9). Otherwise, the rates are not corrected for the stimulated emission. By analogy, stimulated recombination can be treated as negative photoionization via the Eq. (A.15). The energy of states can be defined as following: the ground state is zero, following excitation energies are equal to sum of ionization potential and an excitation energy of a given ion. Therefore, no internal transitions are possible into the ground state. The total transition rate (for all transitions) from the i -th level is:

$$L_i^{\text{type}} = \sum_j p_{ij}^{\text{type}}, \quad (2.46)$$

where ‘type’ has a value from the set rad deexc., rad. recomb., col deexc, col recomb, internal: upward/downward jump within or to lower/upper ionization state. During the calculation, a cumulative probability is needed for a right choice of upcoming process. We define the cumulative probabilities for individual processes (a summation over types),

$$\mathcal{L}_k = \sum_{\text{type}=1}^k L_i^{\text{type}} \quad (2.47)$$

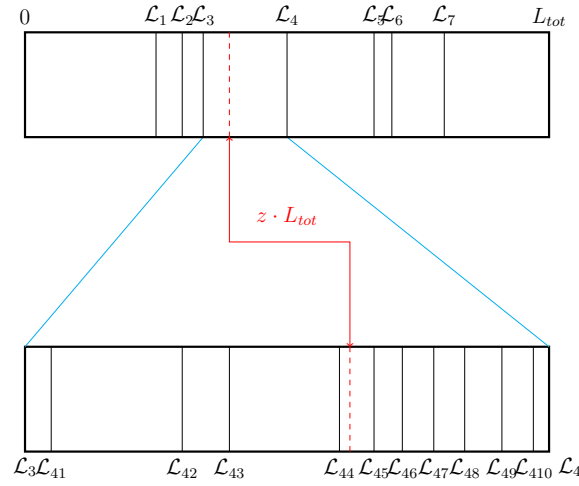


Figure 2.8: Scheme of choosing the corresponding process. There is saved information for concrete transitions (lower rectangle represents transition of a specific type with a definition (2.48)). Number of these transitions can be very high. For saving a computer time, first of all it is determined which process occurs. All total transition probabilities for specific processes are determined with Eq. (2.46). After the process determination the program goes through all possible transitions and chooses the right one which satisfies the condition described in the text. In the example here eight processes are possible, out of them the fourth process is chosen and it contains eleven transitions. The fifth transition is then selected. Adopted from Fišák et al. (2022).

used in the upper panel of Fig. 2.8. Clearly $\mathcal{L}_0 = 0$. We can also define a cumulative probability for a transition

$$\mathcal{L}_{i,x} = \mathcal{L}_{i-1} + \sum_{y=1}^x p_{iy}^{\text{type}}. \quad (2.48)$$

Here x is an index of the last transition taken into account. The choice of the transition, see Fig. 2.8 is following: a random number then it is multiplied with the total rate $L_{\text{tot}} = \mathcal{L}_N$ (N is the total number of included processes ($N \leq 8$, cf. Eqs. 2.44 and 2.45)), hence the probability is rescaled to all processes correctly. Firstly, the condition $\mathcal{L}_{q-1} \leq z \cdot L_{\text{tot}} < \mathcal{L}_q$ (where $q \in \{1, 2, \dots, N\}$ and $\mathcal{L}_0 = 0$) is tested. Then the process is known. If the transition is needed, the second step is calculated in the similar way to the previous step: $\mathcal{L}_{q,x-1} \leq z \cdot L_{\text{tot}} < \mathcal{L}_{q,x}$ is tested and the concrete transition is chosen; these indexes q and x are denoted as Q and X , respectively. This two-step process is faster than a one-step process which goes through all transitions once and determines directly a specific transition. Therefore, this algorithm is implemented into i and k packet machinery.

The rates depend on the Sobolev optical depth via Eqs. (A.4) and (A.7), see also Klein & Castor (1978). For spherical non-homological velocity fields μ does not vanish and Eq. (2.29) depends on μ . The change to r-packet requires a new direction vector and frequency of the packet, which is randomly sampled

$$\vec{n} = (\sin(\Theta) \cdot \cos(\Phi), \sin(\Theta) \cdot \sin(\Phi), \cos(\Theta)), \quad (2.49)$$

where Θ and Φ are determined using Eqs. (2.19c) and (2.19b), respectively.

The frequency is equal to chosen transition frequency, in the case of radiative deexcitation. If the radiative recombination is chosen, the corresponding transition is chosen too, moreover, this frequency is sampled via an integral equation

$$\int_{\nu^{\downarrow}}^{\infty} d\nu (\eta_{im}^{\text{fb}}(\nu))^{\downarrow} = z \int_{\nu_i}^{\infty} d\nu (\eta_{im}^{\text{fb}}(\nu))^{\downarrow}, \quad (2.50)$$

where $z \in [0, 1]$ is chosen randomly and the emissivity is given by Eq. (A.22) and ν_i is the ionization frequency. The method of sampling of a frequency is following: first the integral on the right hand side is calculated. Second, the left hand side integral is repeatedly evaluated as a function of its lower boundary until the integral value is approximately the same as the right hand side. The lower integration boundary value ν^{\wedge} is a new r-packet frequency.

The collisional deexcitation (2.44c) and recombination (2.44d) change i-packet into k-packet. A k-packet do not posses any specific information, therefore, no additional calculations are needed.

2.5.2

Kinetic energy packets

Kinetic energy packets represent free electrons in the medium. In the Macro-atom approximation, this energy can be implemented into the scheme as an additional energy pool. The k-packet do not contain any information about status, moreover it does not propagate in the space, although it changes into a r-packet or an i-packet. Which type is chosen depends on a random choice based on cooling rates. The procedure is similar to the procedure described in the Section about internal energy packets; it is as the same algorithm as in the Fig. 2.8. The possible processes are collisional excitation, free-free emission, free-bound transition and collisional ionization.

The total collisional cooling rate is equal to (Kromer, 2009)

$$\mathcal{C} = \sum_{i=1}^{\mathcal{N}_I} n_i \mathcal{C}_i^{\text{ff}} + \sum_{i=1}^{\mathcal{N}_L} \left(n_i \mathcal{C}_{im}^{\text{fb,sp}} + n_i \mathcal{C}_{ip}^{\text{ion}} + \sum_{i'=i+1}^{\mathcal{N}_L} n_i \mathcal{C}_{ii'}^{\text{exc}} \right), \quad (2.51)$$

where \mathcal{N}_I is the total number of ions (of element k), and finally \mathcal{N}_L is the total number of levels (of ion j of element k). Individual processes leading to a change of a k-packet include

- collisional excitation

$$\mathcal{C}_{iu}^{\text{exc}} = n_i C_{iu} (\epsilon_u - \epsilon_i), \quad (2.52)$$

where C_{iu} is given by (A.17).

- free-free emission

$$\mathcal{C}_i^{\text{ff}} = C_0 q_i^2 T_e^{1/2} N_i n_e, \quad (2.53)$$

where $C_0 = 1.426 \times 10^{-27}$ in CGS units, q_i is the charge of ion i , and N_i is the ion i concentration,

- free-bound transition (radiative recombination)

$$\mathcal{C}_{mi}^{\text{fb,sp}} = N_m n_e \left(\alpha_i^{\text{E,spont}} - \alpha_i^{\text{spont}} \right) (\epsilon_{m_{L_0}} - \epsilon_i), \quad (2.54)$$

the cross section $\alpha_i^{\text{E,spont}}$ can be found in the Eq. (A.21). The first term represents thermal and ionization energy converted to radiant energy and the second term the ionization energy spontaneously converted to radiant energy in a given f-b transition.

- collisional ionization

$$\mathcal{C}_{im}^{\text{ion}} = c_{im} (\epsilon_{m_{L_0}} - \epsilon_i). \quad (2.55)$$

where c_{im} is written in the Eq. (A.20).

A k-packet transforms into an i-packet via collisional excitation (2.52) or ionization (2.55). A corresponding transition (or ion) must be chosen, because i-packet has to contain information, which element (ion, level) will be set during the initialization.

Free-free (2.53) and free-bound (2.54) transitions change k-packets to r-packets. A new frequency ν^{λ} of this packet is in the free-free case determined from the equation

$$\int_{\nu^{\lambda}}^{\infty} d\nu \left(\eta_{j_i j_j}^{\text{ff}}(\nu) \right)^{\lambda} = z \int_0^{\infty} d\nu \left(\eta_{j_i j_j}^{\text{ff}}(\nu) \right)^{\lambda}, \quad (2.56)$$

$z \in [0, 1]$ is chosen randomly. In the case of free-bound transition the equation is as the same as the Eq. (2.50).

CHAPTER 3

The general code structure

We present a Monte Carlo code for the stellar wind (and other interstellar matter) radiative models. The general code structure is shown in Fig. 3.1.

The basic information must be read: chemical composition, atomic data, input parameters, and input model (hydrodynamical model) of an atmosphere, which is a precalculated model (theoretical or a more realistic hydrodynamical model calculated with another code). The data are collected to an object called the model grid (modGrid). It is a set of discrete points where the physical characteristics are defined. Based on the modGrid, the code creates another grid called the propagation grid (propGrid). The whole propagation of packets is determined in this grid, while the physical properties like the ionization fractions, electron density, etc. are calculated in the modGrid. The propGrid can be chosen to be regular or adaptive. The adaptive propGrid takes into account the non-homogeneous distribution of modGrid points. The propGrid and modGrid are connected. Every propGrid cell has a modGrid cell associated. Currently, three additional modGrid cells are created for propGrid cells outside the computational domain, or if the closest modGrid grid is at a larger distance than allowed by a condition (depends on the definition specific for every model).

The packets are created at a defined lower boundary $r = R_*$ and propagated through the propGrid until they reach the outer boundary R_∞ . Propagation and interactions are based on random number generation. Random numbers are generated with a function initialized with a seed which is an input parameter. If the seed value is positive, the input value is not changed, otherwise the input value is recalculated based on the CPU time and the number of processor rank.

The basic parts of the code had been developed by Brankica Kubátová (Šurlan), and by Markus Kromer. The code included a simple 1-D model and basic propGrid, simple hydrogen atomic data, propagation of a packet in the basic propGrid, including resonant line scattering. This version was unpublished. Full macro-atom approach, adaptive propGrid, more various 1-D – 3-D models, including subroutines for full 3-D calculations, parallelization, were implemented by the author of this thesis. A paper (Fišák et al., 2022) has been published (accepted) together with the initial authors of this code.

3.1

Atomic data

The atomic data are divided into three files: atomic levels, atomic transitions and photoionization cross section data. The code is written in a way to enable simple implementation of another atomic data source. The atomic data are read during the initial procedures and are not changed in the following subroutines. One of the data sources is the Opacity project Delahaye et al. (2016). Data can

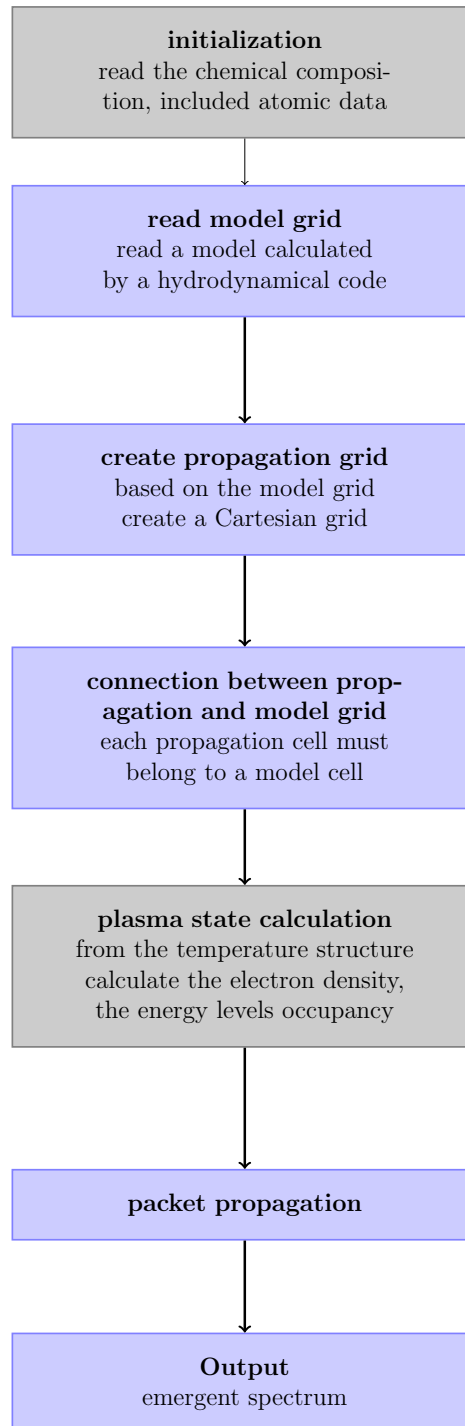


Figure 3.1: The current code structure.

be simply copied from the project web sites and saved into a simple text file. No more changes is needed.

Energy levels file contain the principal quantum number, the configuration, energy, and degeneration of a level. Energy of a level is defined like in [Lucy \(2002\)](#): the ground level energy is equal to zero and the energy is equal to excitation energy. In the case of ionized atoms the level energy is equal to the ionization energy plus the excitation energy. Thus every level has its unique number of excitation energy. The fully ionized ion contains only one energy level with the total ionization energy and statistical weight equal to one. This definition is very useful for the macro atom scheme.

Levels and transitions are read separately, therefore, transitions have to be connected with corresponding upper and lower levels. The format of the Opacity project data is strictly defined. Both files include electron configurations of lower and upper levels, therefore the identification is processed via a comparison of two arrays. In addition, the transition file contain Einstein values A_{lu} , oscillator strengths, and the transition wavelength in Angstroms.

It is not difficult to include atomic data from another source. In this terminology, difficult means that the whole code structure has to be rewritten. The Kurucz atomic data ([Kurucz & Bell, 1995](#)) source was used in the chapter 5.

3.2

Model grid

A spatial distribution of physical quantities of a continuum can be described with more less complicated analytical functions. The numerical modelling allows using much more complicated distributions, however, defined only in discrete points.

Generally, models are calculated in three dimensions. The number of dimensions can be decreased by assuming a certain symmetry. A very common symmetry is the spherical symmetry; a variable parameter is the radius. Every spherical shell represents an iso-area (all physical quantities are constant on this surface). This model is designated 1-D model.

The general computational domain structure is depicted in [Fig. 3.2](#). Three main radial zones are labeled here. The stellar zone is $r < R_*$, the main computational domain is $R_* \leq r \leq R_\infty$, and zone beyond the computational domain is $r > R_\infty$. Packets are emitted from the lower boundary independently of the propGrid cells. Packets propagate then through the computational domain until they reach one of lower, or upper boundary.

As was mentioned above, the modGrid posses information about physical structure of the model. Important physical variables, which modGrid must contain, are: temperature, density, chemical composition (abundances), ionization fractions and occupation numbers. These quantities are updated before the packet propagation. Populations and ionization fractions can be read from a file, however, usually the occupation numbers are calculated from the temperature (LTE) or temperature and radiation field (NLTE, not implemented yet).

Values are defined only for isolated modGrid points (area, etc.), although, the calculations are processed in the continuum space and a quantity in every point must be known, hence a proper approximation must be chosen. We assume that every physical quantity in the propGrid cell is constant in the whole cell volume. Although the velocity field, which is an exception; values must be interpolated from the values defined in discrete values, otherwise it will not be possible to calculate the Sobolev resonance points. In the first approximation, the velocity can be described with an analytical function. This description is independent of the propGrid.

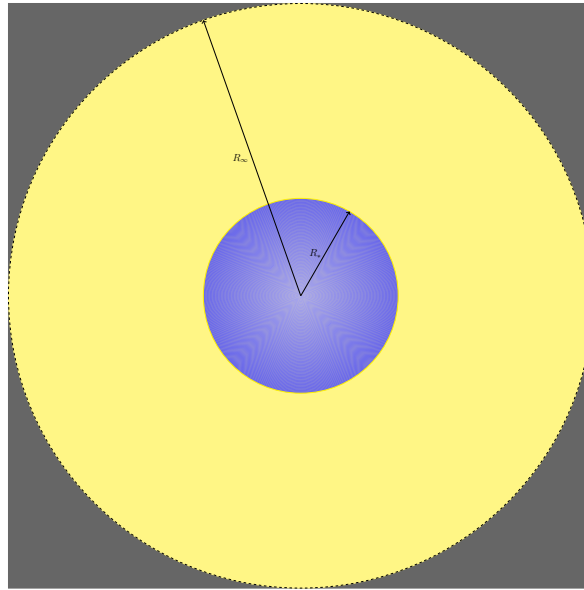


Figure 3.2: General model structure. Blue area represents the central star with radius R_* , the main computation domain is yellow coloured, $r \in (R_*, R_\infty)$ the black area represents zone beyond the model.

3.3

Propagation grid

The second grid is a propagation grid (propGrid) where the whole packet propagation is determined. It is a Cartesian grid which creation is based on the modGrid. The advantage of Cartesian grid is the independence on input model geometry during the calculations. There is only one transformation between these geometries and the general procedure can be used independently of input modGrid geometry.

The propGrid size must be large enough to include the whole modGrid. The star is placed in the coordinates origin $(0, 0, 0)$. The propGrid is defined by the interval coordinates: $x \in (x_{\min}, x_{\max})$, $y \in (y_{\min}, y_{\max})$, $z \in (z_{\min}, z_{\max})$. The physical size of rectangular parallelepiped grid is then

$$(L_x, L_y, L_z) = (x_{\max} - x_{\min}, y_{\max} - y_{\min}, z_{\max} - z_{\min}),$$

furthermore, the grid is symmetric with respect to axes, which means $x_{\min} = -x_{\max}$, $y_{\min} = -y_{\max}$, and $z_{\min} = -z_{\max}$. The propGrid is divided on (N_x^B, N_y^B, N_z^B) rectangular parallelepiped cells:

$$(w_x, w_y, w_z) = \left(\frac{L_x}{N_x^B}, \frac{L_y}{N_y^B}, \frac{L_z}{N_z^B} \right).$$

One can see in Fig. 3.3 that every propagation basic cell has its integer indexes – one vector (n_x, n_y, n_z) and one scalar index $N(n_x, n_y, n_z)$ which is calculable from the vector. The vector indexes are numbered in the interval $(1, N_i^B)$ and the scalar index is equal to

$$N = n_z + (n_y - 1) \cdot N_z^B + (n_x - 1) \cdot N_y^B \cdot N_z^B, \quad (3.1)$$

this index is defined because the cells are saved as an array of objects in the memory, N represents the position index in the array. The n_x, n_y and n_z represent the integer vector coordinates. The counting

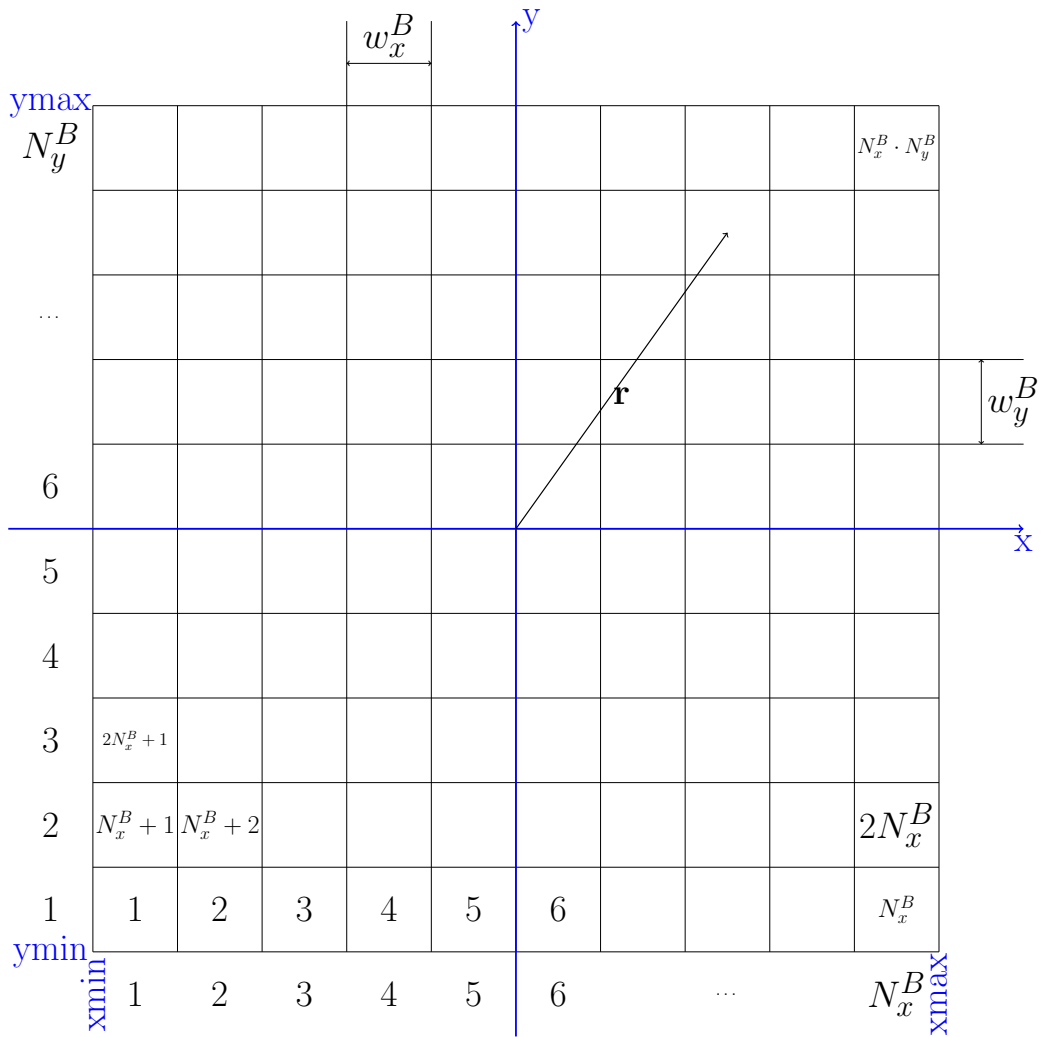


Figure 3.3: Layer of propGrid for $n_z = 1$. The physical coordinates are between points given by $x \in (x_{\min}, x_{\max})$, $y \in (y_{\min}, y_{\max})$, $z \in (z_{\min}, z_{\max})$. Every cell is given by its position and width. The basic modGrids' widths are the same and equal to w_i^B , $i \in \{x, y, z\}$. It has a scalar index and vector index $\in (N_x^B, N_y^B, N_z^B)$. The maximal scalar index is equal to $N_x^B \cdot N_y^B$.

order is obvious from the formula (3.1). One can calculate these indexes from the physical position using the formula

$$n_i = \left\lfloor \frac{r_i}{w_i} + \frac{N_i^B}{2} \right\rfloor + 1, \quad (3.2)$$

where $i \in \{x, y, z\}$, r_i is the position component and $\lfloor \cdot \rfloor$ represents the floor of the number.

Every cubic cell has defined unequivocally wall labels. The label is based on a wall direction (normal vector), i.e. (x, y, z) and a sign '+' in the case the wall is in the positive direction, or '-' the wall is in the negative direction from the propGrid cell center, see Fig. 3.4.

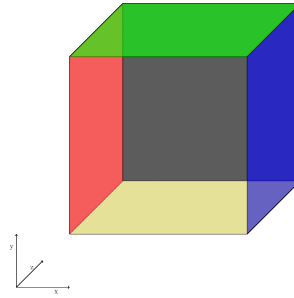


Figure 3.4: Propagation cell and its orientation to the coordinate system. The boundaries are signed as follows: the blue $x+$, the red $x-$, the green $y+$, the yellow $y-$, the front $z-$ and back $z+$. The colour cube was created using the code from <https://tex.stackexchange.com>.

3.4

Adaptive grid

The modGrid is ball-shaped object closed in a cube. The volume of a sphere with a radius r is equal to

$$V_o = \frac{4}{3}\pi r^3, \quad (3.3)$$

and the volume of a cube with a size of edge is

$$V_s = a^3 \quad (3.4)$$

Let us assume the most efficient configuration $a = 2r$, then the volume ratio is equal to

$$\frac{V_s}{V_o} = \frac{6}{\pi} \approx 1.91. \quad (3.5)$$

Therefore the $V_s \approx 2V_o$, and the third of the volume is located beyond the upper boundary. Hence a choice of too dense regular propGrid leads to a large amount of propGrid cells with no computational benefit. Moreover, the modGrid can be distributed inhomogeneously in the computation domain, as a consequence, some regions must be described with a finer grid. The size of the propGrid cell should be as small as the grid resolution. The regular grid can consume a large amount of memory. Our aim is to develop a more efficient propGrid which covers small regions with a fine grid and other regions do not consume a large amount of memory.

Therefore a development of a non-regular grid is advantageous. The non-regular grid is based on set of regular grids in a tree structure. The basic cells can be divided by regular grids onto smaller cells and these cells can be divided too, etc. There are two main grid types in the code,¹ see Fig. 3.5. The octgrid is defined in the following way:

- a cell is divided on $2 \times 2 \times 2$ subgrid
- every subcell of an arbitrary subgrid can be divided

This kind of grid has been already developed (Kurosawa & Hillier (2001), Barnes & Hut (1986)), it is very advantageous to use this kind of grid. A cell is divided into eight identical subcells (which is a reason for naming it ‘octgrid’). Originally, the implementation of this grid into the code was inspired

¹We have a plan to unify these types of grid.

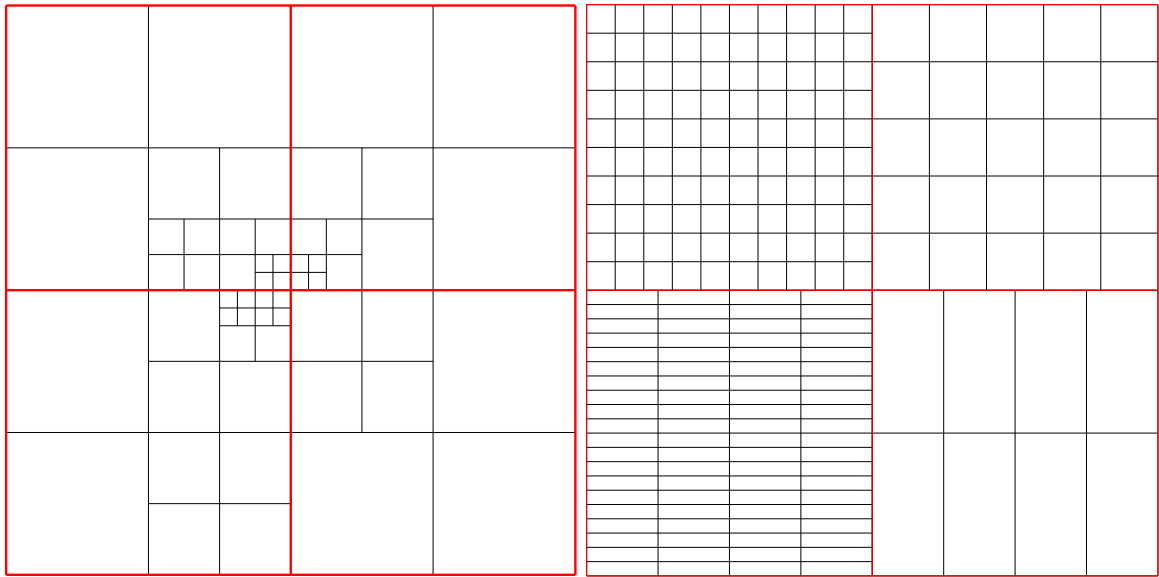


Figure 3.5: *Left:* adaptive propGrid 'octgrid'. Every cell can be recursively divided into four (in a 2-D case) or eight (in the 3-D case) cells until the predefined minimum cell limit is reached. *Right:* apg 'one-level subgrid' type. In every basic cell (red color) a subcell with dimensions N_x, N_y, N_z and variable widths w_x, w_y, w_z is created.

by a lecture during the spring school at Les Houches in the May 2016. The principle is as the same as in the cited papers, we did not copy any part of another code. The second adaptive propGrid type is one-level subgrid, which is defined:

- a cell is divided to a subgrid $N_x \times N_y \times N_z$,
- a subcell cannot be divided.

This adaptive propGrid is depicted in the right panel of Fig. 3.5. The algebra connected with these grids is similar to the basic grid. The calculation of a propGrid cell index is a simple application of the Eq. (3.2). In the case of the octgrid, a simpler algorithm can be applied – only calculation of a quadrant, which the point occupies.

A tricky part lies in the creation of the adaptive propGrid for a specific modGrid, because the number of cells in a zone must be proportional to the number of modGrid points. Therefore so-called virtual points are created. It is a representation of a modGrid; it is more easy to count number of virtual points in a propGrid cell, than to measure an area of a modGrid cell. Every virtual point possesses an information about its position and the basic propGrid cell including this point. In addition, a statistical weight can be defined, and a modGrid cell index can be calculated.

The simplest case is the 3-D model. The modGrid cells are represented by dimensionless points. 2-D and 1-D models are more complicated. Usually calculated 1-D cases are defined with spherical shells, see Fig. 3.6. The virtual points are distributed among shells. The total number of virtual points is defined by user. A local number of virtual points for a selected shell is proportional to the area of the shell. The azimuthal and polar coordinates are calculated randomly.

Once the virtual points are distributed, the adaptive propGrid can be created. The octgrid is created as follows: a number of virtual points is counted in every basic propGrid cell. If a number of virtual points is larger than a predefined number, \mathfrak{N} , the cell will be divided. The created subcells are tested for the same condition and, again, the cell is divided, if the number $> \mathfrak{N}$, the division is repeated again. Theoretically, the level of division is not defined, however, technically, a minimal propGrid cell

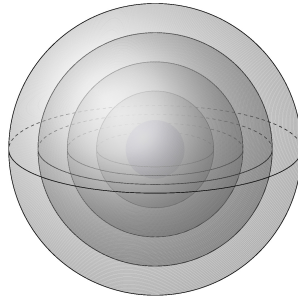


Figure 3.6: Scheme of 1-D model consisted of spherical shells.

width is defined; if the divided cells would be smaller than this variable, then no division is processed. The creation of one-level grid is less complicated. A subgrid division is proportional to a number of virtual points located inside the current basic propGrid cell

$$(N_x, N_y, N_z) = (\lfloor N_{vp}^\alpha \rfloor, \lfloor N_{vp}^\alpha \rfloor, \lfloor N_{vp}^\alpha \rfloor), \alpha \in \mathbb{R}, \quad (3.6)$$

N_{vp} denotes a number of virtual points in a propGrid cell. The choice of concrete α is more complicated task and depends on concrete models. In the current version of code, every mpi process creates a unique propGrid; this property may be changed in future.

3.5

The connection between the modGrid and the propGrid

The propGrid is created alongside the modGrid, however, each propGrid cell does not contain any information about the physical conditions. During the propagation, the program solves propagation only in the propGrid, therefore, each propagation cell must be connected to some modGrid cell. Every propGrid cell has saved an index of a modGrid cell and we can access important information about physical conditions in this cell during the calculation.

The first task is to map the appropriate modGrid cell to a given propagation cell. Let us consider several cases. The simplest case is the regular modGrid – every point is on the regular net point of intersection. Thus a regular propGrid can be created in the way that each modGrid point is in the centre of the propagation cell. The problem is larger in the case of spherically symmetric modGrid. Each modGrid cell is represented by its radial distance from the origin. We have to find the closest modGrid cell for each propagation cell. Generally the distance between two sets is calculated using the formula

$$l = \min \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}, \quad (3.7)$$

where (x_0, y_0, z_0) represents the centre of the propagation cell and $\{x, y, z \in \mathbb{R}^3 : (x, y, z) \in \text{MC}\}$. In the case of spherically symmetric model, the point on the sphere with the shortest distance between the propagation cell centre lies on the line connecting the origin and the propagation cell centre. Thus the radial distance of the propagation cell center must be calculated and the distance between two cells is equal to difference between its radial distances,

$$l = \min\{r - r_I | I = 1, \dots, n_{\text{mg}}\}. \quad (3.8)$$

Here, n_{mg} (number of modGrid cells) distances.

It is obvious that the spherical shell is transformed to a different shape. One can imagine this process as a pixel representation of a picture on a screen. Thus circle is not an accurate circle. With increasing density of propagation cells the shape is more circular. All physical variables are constant in the whole volume of the propGrid cell except velocity, which must be continuous vector field, since we must calculate resonance points of transitions.

A connection between propGrid and modGrid grid can lead to a following event: a propagation cell is far away from any modGrid cell but still, it is connected to the cell – the closest cell. Another condition must be added – a maximal distance between these two cells. We set up this distance as

$$l_{\text{max}} \leq k \cdot \sqrt{w_x^2 + w_y^2 + w_z^2}, \quad (3.9)$$

where w_i is the width of a propagation cell, and $k > 1$ is a model dependent constant defined by a user.

The adaptive propGrid creation and its connection with the modGrid can be a numerically expensive task. The code offers an option to save modGrid and propGrid in a file with all information including connection between these grids. This file can be read in another run of the program and the calculations are proceed without modGrid reading and propGrid construction, therefore, packet propagation can be studied for different set of parameters. Furthermore, the saved data are useful source for the creating of figures of the grids.

3.6

The packet propagation through the propGrid

The packet propagation through the grid is a crucial part of the calculations. One task is to find out which propGrid cell wall will be crossed by a packet. Since the packet trajectory is a straight line, the solution is simple. The packet position and its direction is known and the current propGrid cell is known too. Therefore, simple equations describing the path (a parametrization of a line) can be written

$$\vec{r}_i = \vec{r}_0 + \vec{n}t, \quad (3.10a)$$

$$\vec{r}_i + \vec{w}_i = \vec{r}_0 + \vec{n}t, \quad (3.10b)$$

where \vec{r}_i is the position of a propGrid cell corner, \vec{w}_i is the width of the current cell, \vec{r}_0 is the current position of the packet, \vec{n} is a propagation direction of the packet, and, finally, t is a curve parameter. These six equations should be solved separately, we get six possible distances from the calculation of the parameter t

$$t_j = \frac{(r_i)_j - (r_0)_j}{n_j}, \quad j = 1, 2, 3, \quad (3.11a)$$

$$t_{j+3} = \frac{(r_i)_j + (w_i)_j - (r_0)_j}{n_j}, \quad j = 1, 2, 3. \quad (3.11b)$$

j represents the vector index. If a packet is located inside the propGrid cell, three values of t are positive, and three values are negative. The smallest positive number is a distance to the nearest cell.

The next crossing cell is usually calculated for a packet located in the propGrid cell boundary. One of calculated t_j is equal to zero. However, because of numerical uncertainties, a position can be slightly in front of or beyond a wall in specific cases. This causes problems during propagations, hence, a packet must have saved its last (formally) crossed boundary to avoid a propagation to the

boundary, which should have been already crossed. This case leads to an error in calculation of a current propGrid cell index.

After a cross boundary calculation, a neighbour number must be obtained. In the case of regular propGrid is the algebra simple. If a cell number is known, then the neighbour numbers are very easy to determine from Eq. (3.1) because of strictly defined cell numbering. On the other hand the adaptive propGrid is more complicated. Based on the adaption propGrid creation rules, the next propGrid cell index can be obtained in the following way:

1. find next crossing with the propagation cell,
2. if the neighbour cell number is known we go to a step 4,
3. if the neighbour cell is not known, we will look to a lower cell level and its neighbour in the given direction, we repeat this point until we find a neighbour cell (the lowest level is of course the basic grid),
4. if the upper cell is non-zero, we will find out which cell it is, if the cell is divided, this procedure is repeated until a top of the tree is reached.

This procedure is demonstrated in Fig. 3.7. The presented scheme represents a relative way to calculate an index of a packet. It is possible to calculate a position independently of this scheme. The equation (3.2) can be applied in all levels of grid, however, the corner and a width of the propGrid cell, where is a subgrid located, must be taken into account.

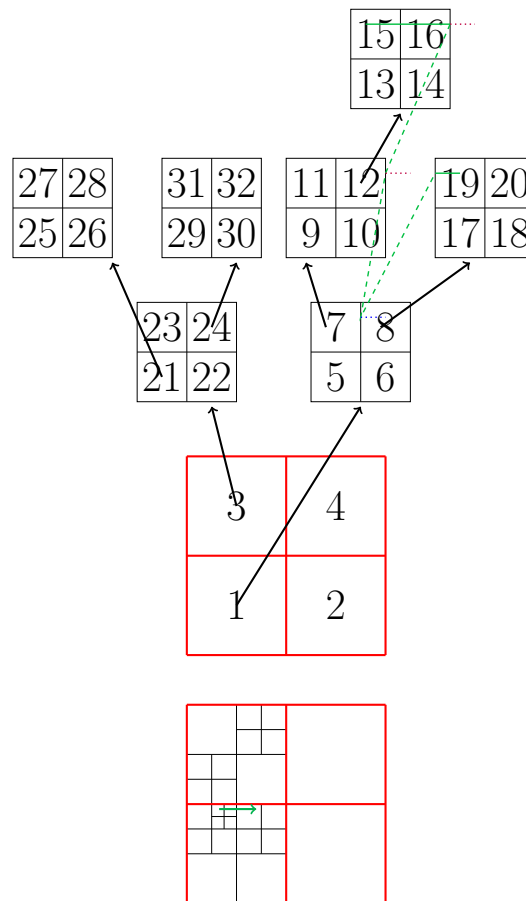


Figure 3.7: Scheme of the packet propagation through the grid. The bottom scheme shows a propagation grid. The upper scheme shows the same grid, but expressed in a tree structure. The green line is an example of packet's path. It is simple to show this path on the lower graph. The upper graph explicitly shows the adaptive propagation grid level structure including the illustration of cell indexing. The numbers in the figure describe the logic of cell indexing in the code. First, the basic cells are indexed. Then the subcell of the basic cell with the lowest number. First all cells of a given subgrid are numbered and after that the numbering continues with subgrids of next cells (with an increasing index). Adopted from [Fišák et al. \(2022\)](#).

CHAPTER 4

Tests of the propGrid

The calculation of the spectra via the propGrid should give the consistent results. We show several tests run with the different propGrid to show the difference among the output results based on the propGrid properties.

4.1

The model setup

The model is assumed to be spherically symmetric and described with the homologous approximation. We add a spherical clump – a steep increase of density which increases the opacity rapidly. The matter is composed of the ions in the Tab. 4.1.

We created two different density profiles: single and double-peak-density spherical clump. The density structure is the following

$$\rho(r) = \begin{cases} \rho_0 \left(\frac{r}{R_*}\right)^{-2} & R_1 \geq r > R_*, \\ \rho_0 \left(\frac{r}{R_*}\right)^{-2} + \rho_{\text{cl}}(r) & R_2 > r \geq R_1, \\ \rho_0 \left(\frac{r}{R_*}\right)^{-2} & R_\infty > r \geq R_2, \\ 0 & r \geq R_\infty, \end{cases} \quad (4.1)$$

where R_1 , and R_2 is predefined interval of a clump, ρ_0 density at $r = R_*$, and the function $\rho_{\text{cl}}(r)$ takes the form

$$\rho_{\text{cl}}(r) = \rho_{\text{cl}} \exp \left[-\frac{\left(r - \frac{R_1 + R_2}{2}\right)^2}{(R_2 - R_1)^2} \right] \quad (4.2)$$

for a single-peak-density clump, and

$$\rho_{\text{cl}}(r) = \rho_{\text{cl}} \left[\exp \left(-\frac{(r - r_{01})^2}{\sigma^2} \right) + \exp \left(-\frac{(r - r_{02})^2}{\sigma^2} \right) \right] \quad (4.3)$$

for a double-peak-density clump. Here, σ is a half-width of the Gaussian profile, and r_{01} and r_{02} are the centre of the Gaussian profiles. Parameters described in the equations are listed in Tab. 4.2.

Ion	Max. principal num.	N of levels
HI	10	55
HII	1	1
HeI	6	53
HeII	10	55
HeIII	1	1

Table 4.1: The numbers of included levels

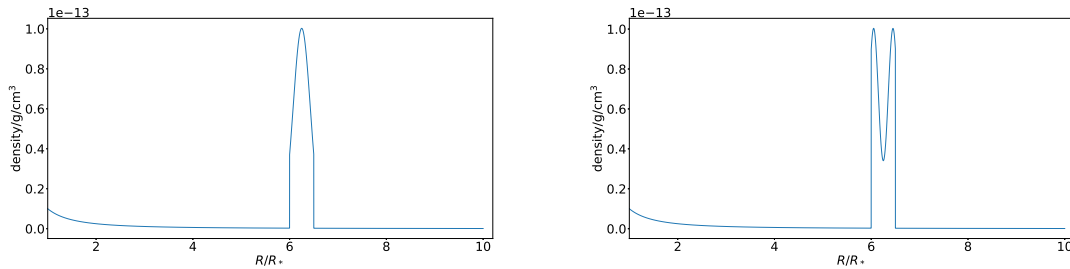


Figure 4.1: Density structure of the test wind model with a spherical single-peak-density clump (upper panel) and double-peak-density clump (lower panel).

effective temperature T_{eff}	14734 K
stellar radius R_*	$16145R_{\odot}$
parameter R_1	$6.0R_*$
parameter R_2	$6.5R_*$
outer boundary R_{∞}	$10R_*$
outer boundary velocity V_{∞}	$2.875 \times 10^9 \text{ cm} \cdot \text{s}^{-1}$
density in an initial time ρ_0	$10^{-14} \text{ g} \cdot \text{cm}^{-3}$
clump density ρ_{cl}	$10^{-13} \text{ g} \cdot \text{cm}^{-3}$
double-peak-density clump	
clump inner boundary r_{01}	$6.05R_*$
clump inner boundary r_{02}	$6.45R_*$
clump width σ	$0.15R_*$

Table 4.2: Parameters used in Eqs. (4.1), (4.2) and (4.3) as a testcase.

4.2

The tested properties

The propGrid has to be created during the computational domain setup, which includes the allocation and reallocation of arrays. This procedure can be very time consuming. We compare times needed for the grid creation and propagation of packets through the grid. Furthermore, the consumed memory is important as well, we show the number of created basic propGrid cells and the adaptive propGrid cells. Moreover, we define two variables expressing an accuracy of propGrid: the first variable com-

compares the total volume of the propagation cells belonging to a model cell and volume of the model cell itself. This quantity is defined

$$\chi_I = \frac{\sum_{\text{PC} \rightarrow I} V_{\text{PC}}}{V_I}, \quad (4.4)$$

where V_{PC} is volume of a propagation cell belonging to the modGrid cell I , and V_I is a volume of the model cell. The χ_I is equal to one in the ideal case, which corresponds to the case of equality of propagation cells volume and the corresponding model cell. For a better characterization the whole propagation grid accuracy, we define the mean value of χ

$$\sqrt{\langle \chi^2 \rangle} = \sqrt{\frac{1}{N_{\text{modGrid}}} \sum_{I=1}^{N_{\text{modGrid}}} (\chi_I - 1)^2}, \quad (4.5)$$

where N_{modGrid} denotes the total number of modGrid points. The value should be as close to zero as possible.

The second parameter is a variable called ‘relative covering’. Every propGrid cell is associated with a modGrid cell, however, not every modGrid cell can be associated to any propGrid cell (even to one). This phenomena is described by a number of modGrid cells having associated at least one propGrid cell and it is equal to

$$\text{rc} = \frac{N_c}{N_{\text{modGrid}}}, \quad (4.6)$$

here, N_c is a number of modGrid cells associated to at least one propGrid cell and N_{modGrid} is the total number of modGrid cells.

The main output of the calculation is a spectrum. The generation of spectra from packets is described in the subsection 2.4.3. All spectra in this section are calculated from all packets from a simulation. Spectra are very sensitive to the input model settings and they can be used as an indicator of the propGrid accuracy. We show comparison of spectra for several cases of propGrid parametrization and we expect that the spectra converge to a final spectrum if the number propGrid cells is large enough.

4.3

The basic propagation grid

The basic propGrid is a regular grid containing propGrid cells with identical size. The first simple test was done with the single-peak-density clump density profile Eq. (4.2). The results of calculations of relative covering for different propagation grids are listed in the Tab. 4.3. The variable $\langle \chi^2 \rangle^{1/2}$ is decreasing for the increasing number of cells and the relative covering is equal to one hundred percent for the case of 100^3 propGrid cells. The calculated spectra are very intriguing. The plot of $\text{Ly}\alpha$ in the upper panel Fig. 4.2 shows a profile belonging to a single-peak-density line. The red-shifted part is a classical P Cyg profile created in an outflow, the blue-part absorption line around 1140 \AA is created in the spherical clump and its shift is caused by strong Doppler effect. Other lines are formed only in the clump. Individual line profiles are shown in the Fig. 4.2 in the lower plots. The spectra with clump can be compared with the spectrum with no clump added ($\rho_{\text{cl}}(r) = 0$ in Eq. (4.2)). The spectra calculated from a single-peak-density profile for low propGrid resolution do not differ significantly from a spectra calculated from a model with no clump. The reason of this phenomenon lies in the propGrid, which does not include the clump density profile in a required detail. While increasing the number of propagation cells, the number of cells connected to the density clump is increasing, and

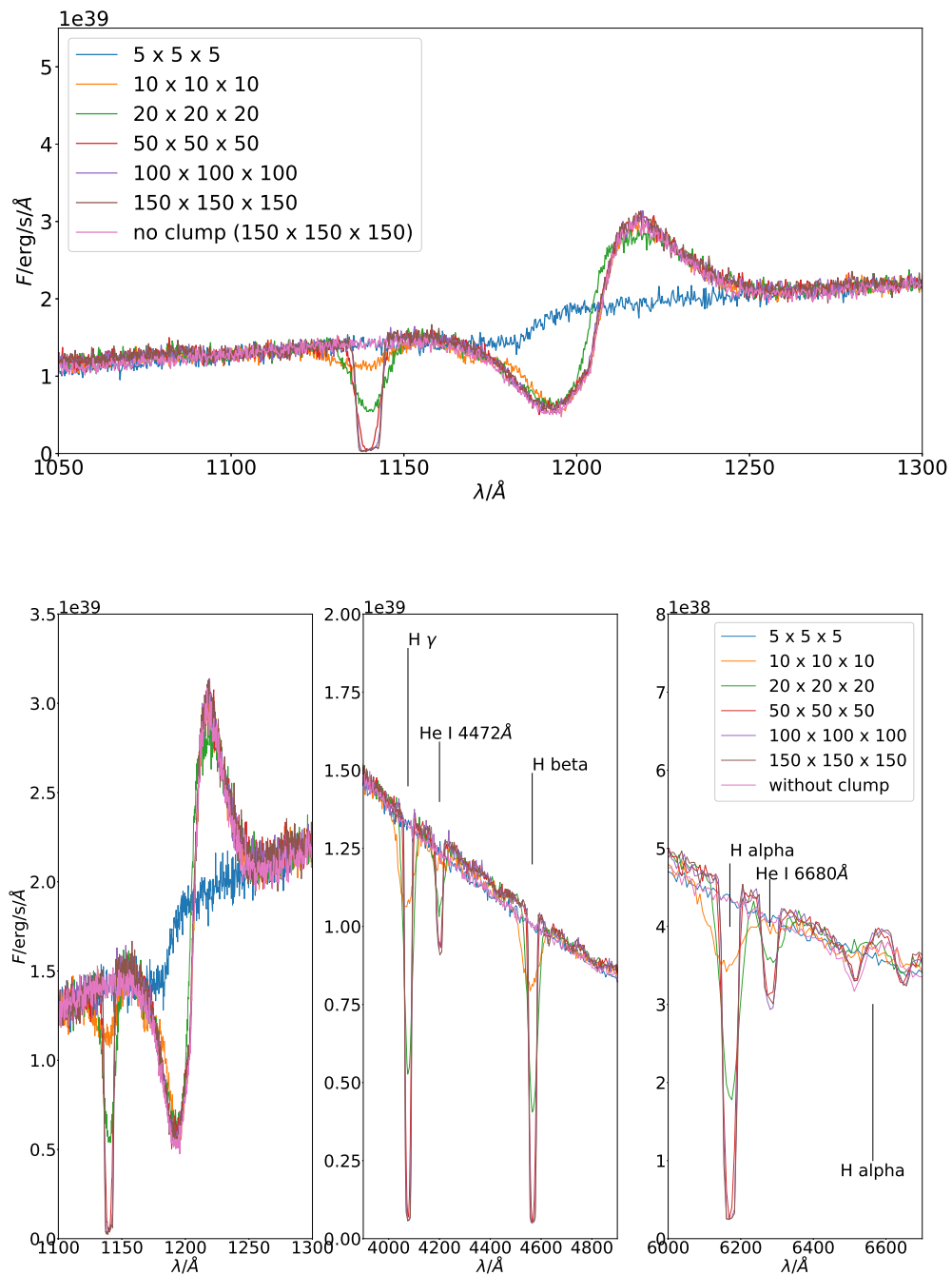


Figure 4.2: Comparison of spectral line profiles calculated with different basic propGrids in the case of the single-peak-density clump model. *Upper panel:* hydrogen Ly α line with a P Cyg profile and the clump absorption component, *lower panels:* Hydrogen Balmer lines and neutral helium absorption lines. Line labels correspond to their vacuum wavelengths.

number of cells	$\langle \chi^2 \rangle$	rel. covering %
5^3	5.77	8.2
10^3	3.86	15.3
20^3	4.07	40.0
50^3	1.29	63.5
100^3	0.33	100.0
150^3	0.15	100.0

Table 4.3: Parameters of the regular grids for the single-peak tests.

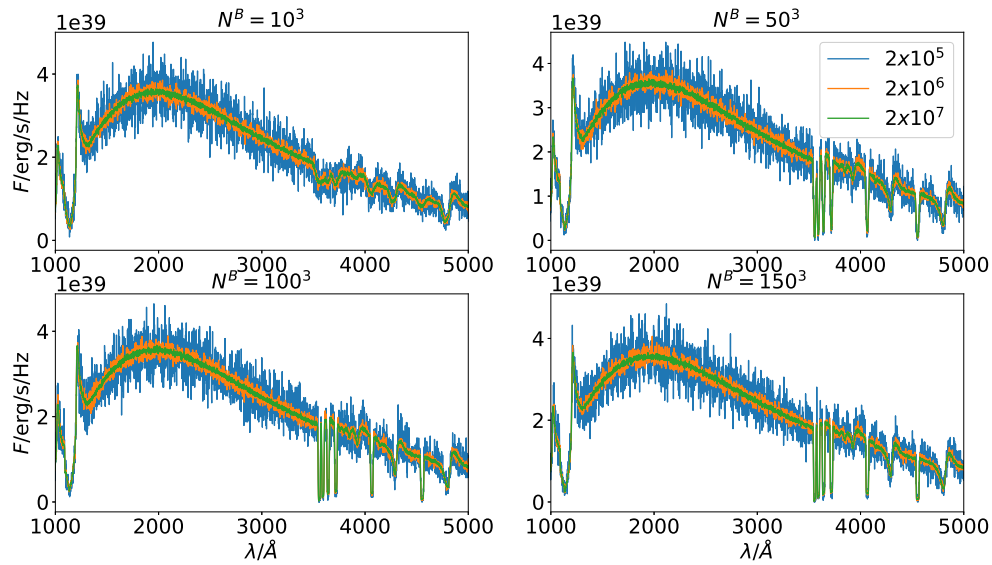


Figure 4.3: Comparison of calculated spectra of models with different propGrid parameters and different number of packets. Four different models, differ with $N_B = \{10^3, 50^3, 100^3, 150^3\}$

also the intensity of the absorption line profiles. The spectra seem to converge for the grids larger than 100^3 cells.

propGrid resolution and number of packets

The code creates the propGrid based on the modGrid, although many parameters are set before the calculation. As a consequence, results can be affected significantly by a wrong choice of parameters. We calculated several spectra with different parameters – total number of packets and size of basic propGrid. The total numbers of packets were set to 2×10^5 , 2×10^6 , 2×10^7 , and basic propGrid sizes 10^3 , 50^3 , 100^3 and 150^3 . Spectra are plotted in the Fig. 4.3, moreover, the case calculated with 10^6 packets and four different propGrids is plotted in Fig. 4.4. The propGrid parameters affect spectra significantly. The spectra calculated with a low resolution propGrid, such as $N^B = 10^3$, include shallow spectral lines. It is caused by insufficient ‘pixelization’ of the modGrid. The fit of propGrid is described by parameters introduced in Eqs. (4.5) and (4.6). Other propGrids possessing larger number of cells describe the modGrid much more sufficiently. The spectra are well converged for the case $N^B = 50^3$.

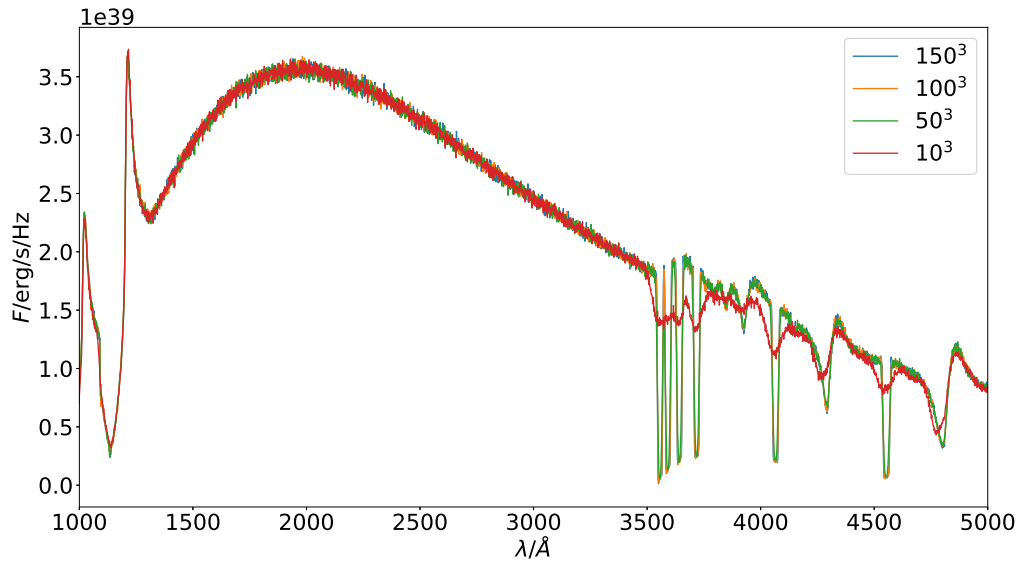


Figure 4.4: Comparison of calculated spectra of models with different propGrid parameters and 10^6 packets.

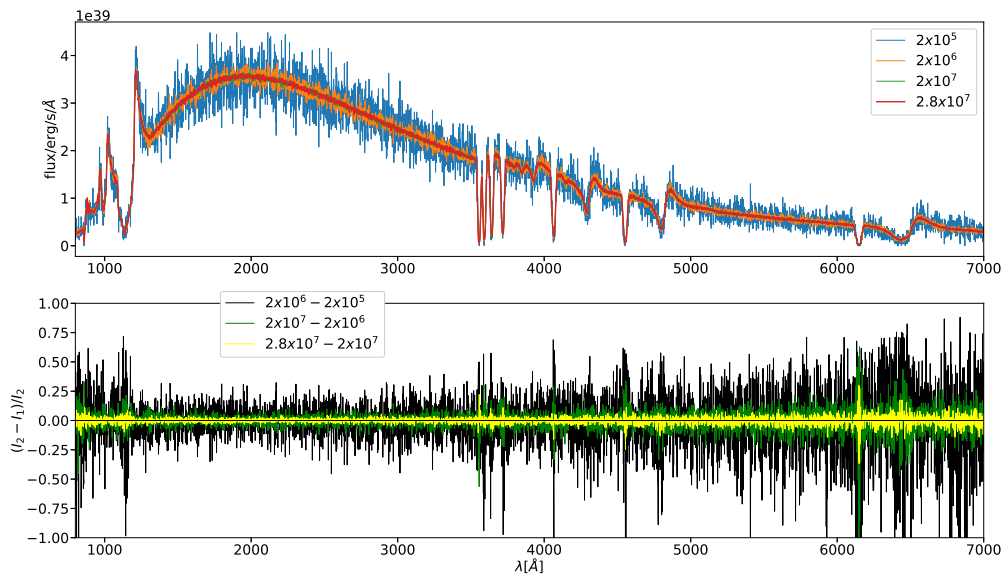


Figure 4.5: Spectra calculated for different number of packets (upper panel). The noise is clearly decreasing with the increasing number of packets. A relative change of spectra among the spectra in the upper panel are plotted in the lower panel.

Number of packets affects the noise in the spectra. It is shown in the Fig. 4.5. The noise decreases in a case with larger number of packets. Clearly, a larger number of packets do not create a better spectrum if propGrid parameters are not chosen well. The line profiles fit better even for smaller number of packets.

num. of cells (N^B) (basic grid)	num. of subcells	num. of virtual points	time of calculation		$\langle \chi^2 \rangle$	rel. covering %
			creation	pack. prop.		
			relative to t_1 $t_1 = 0.617$ s	relative to t_2 $t_2 = 267.18$ s		
regular, see Fig. 4.7						
10^3	–	–	0.001	0.150	3.87	15.3
50^3	–	–	0.1	0.380	1.03	63.5
100^3	–	–	0.7	0.680	0.33	100.0
150^3	–	–	1	1.00	0.15	100.0
octgrid						
10^3	3760	10^3	0.06	0.178	1.97	92.9
10^3	40768	10^4	0.3	0.256	0.61	100
10^3	404788	10^5	3	0.452	0.11	100
10^3	4223624	10^6	231	0.908	0.06	100
50^3	424	10^3	1	0.396	1.02	100
50^3	18496	10^4	6	0.401	0.68	100
50^3	328720	10^5	60	0.484	0.17	100
50^3	899904	10^6	770	0.897	0.06	100
one-level subgrid						
10^3	965	10^3	0.03	0.174	2.23	100
10^3	45956	10^4	0.2	0.289	0.45	100
10^3	1631479	10^5	10	0.809	0.07	100
10^3	54463204	10^6	483	2.426	0.06	100
50^3	0	10^3	0.5	0.393	1.30	63.5
50^3	1400	10^4	5	0.403	0.98	100
50^3	149804	10^5	47	0.443	0.15	100
50^3	6357511	10^6	3767	1.002	0.04	100
100^3	0	10^3	4	0.749	0.33	100
100^3	8	10^4	38	0.750	0.33	100
100^3	27902	10^5	402	0.767	0.29	100
100^3	1907557	10^6	3431	0.830	0.05	100

Table 4.4: Computational statistic of the propagation grids in several cases. We show the number of cells in the basic grid (second column), the total number of subcells (if subgrids are present, third column), the relative CPU time needed for grid creation (fourth column) and the relative CPU time of the packet propagation (fifth column). The reference values are corresponding CPU times needed to create the regular grid 150^3 . For each grid the standard deviation $\langle \chi^2 \rangle$ (4.5) is shown. This quantity describes the quality of the modGrid cell covering.

4.4

The adaptive propagation grid

The next test contained double-peak-density clump density profile, see Eq. (4.3). Three types of the propGrid were tested: basic grid, octgrid and one-level subgrid. The results of the regular grid are similar to the previous part. The octgrid is very efficient in the case of smaller propGrid cells. In the Tab. 4.4, one can see that the octGrid with $N^B = 10^3$ fits the modGrid better than the basic grid with $N^B = 50^3$; the total number of created cells is much larger in the second case and the values of $\langle \chi^2 \rangle^{1/2}$ are equal. One-level subgrid is less effective because of long time needed for its creation, which decreases with the increasing division of the basic grid. A slice of the octgrid and the one-level subgrid for the double-peak-density clump case is depicted in the Fig. 4.6. One can see that the grid is really denser in the area surrounding the double-peak-density clump. The one-level grid is less

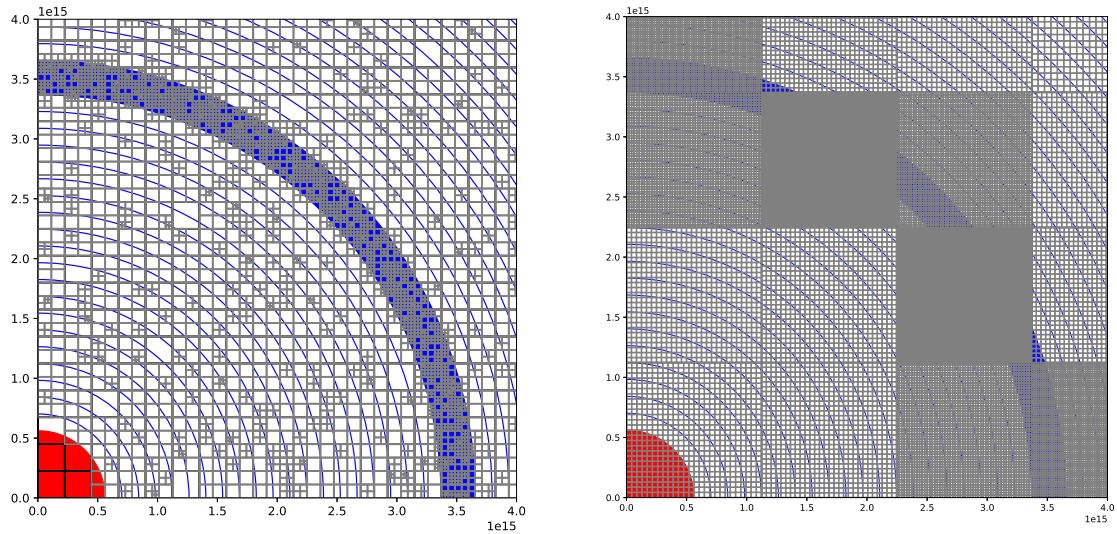


Figure 4.6: Adaptive propagation grids calculated for the double-peak-density clump spherically symmetrical model. We have plotted only one quadrant in the zero plane. The red “circle” is the central star, blue circles represent the model grid shells and the grey and the blue Cartesian grid represent propagation grid. It is clear that in the area of the clump the density of the propGrid is much larger. *Left:* Octgrid, basic grid: 50^3 , 10^6 virtual points, *right:* one-level subgrid, basic grid: 10^3 , 10^6 virtual points.

efficient, since the size of a propGrid subcell is constant in the whole propGrid basic cell. Therefore, a large volume is covered with a dense propGrid.

The spectra are shown in the Figs. 4.7 and 4.8. The characteristic hydrogen double-peak-density absorption line can be seen. It is clear that the shape is caused by Doppler shift in the spherical double-peak-density clump. The propagation cell input parameters must be chosen wisely. This can be seen in the Fig. 4.6. The right figure shows the one-level subgrid based on the $10 \times 10 \times 10$ regular propagation grid. The high density of the subgrid is also located in the large area outside the spherical clump. For the one-level subgrid the choice of denser basic grid is more appropriate. One can see that in both cases the basic grid cells are divided even outside the clump, which is caused by the virtual points distribution, see Section 3.4. The virtual points are not distributed in the clump only but also among the other model cells.

The adaptive grid provides a more suitable approach for the modGrid representation. It does not slow the packet propagation, however its creation is a time consuming task for some cases. This problem can be partially eliminated by parallelization of the grid creation, which is a very difficult task, or enable reading already created propagation grid, if the old one is sufficient enough.

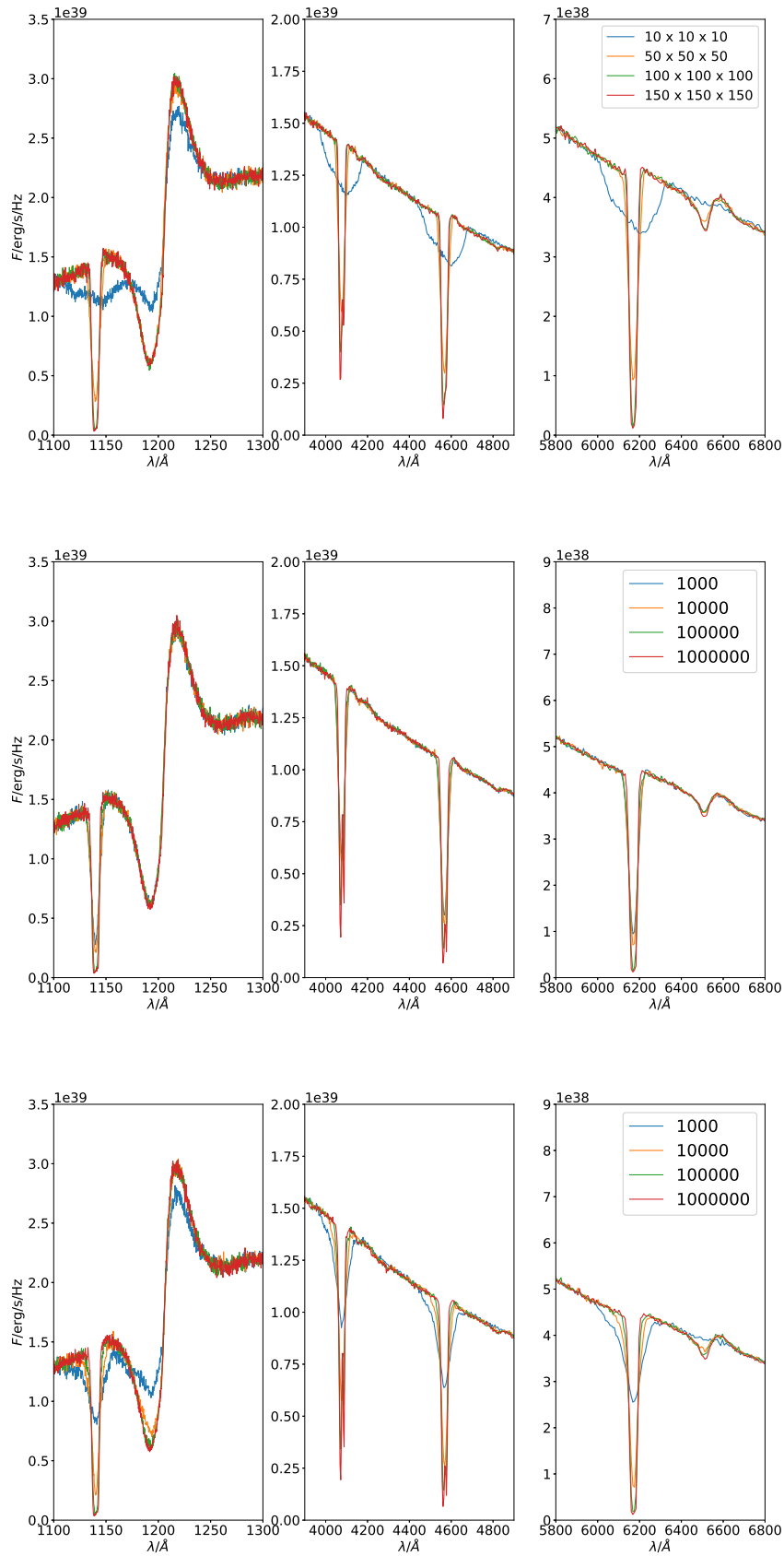


Figure 4.7: Comparison of spectra calculated with different propGrids in the case of the double-peak-density clump density profile. *Upper panels:* with different number of cells in the basic propGrid. *Middle panels:* the octgrid: with different number of the virtual points, basic grid size is $10 \times 10 \times 10$. *Lower panels:* the octgrid: with different number of virtual points, basic grid size is $50 \times 50 \times 50$.

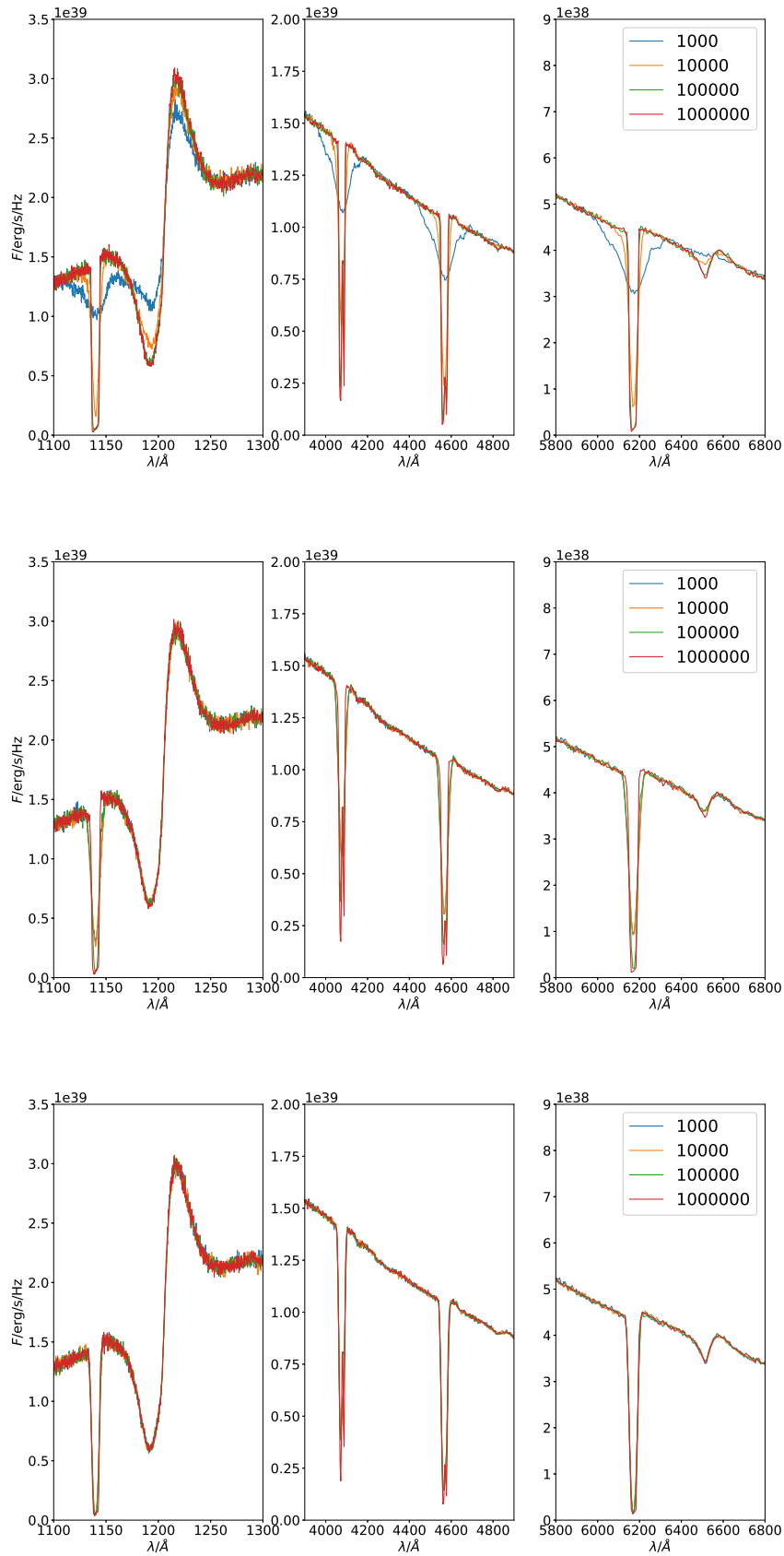


Figure 4.8: Comparison of spectra calculated with different propGrids in the case of the double-peak-density profile. *Upper panels:* the one-level type of the propagation grid with the number of the virtual points as a parameter, $10 \times 10 \times 10$ basic grid. *Middle panels:* one-level grid type, $50 \times 50 \times 50$ basic grid. *Lower panels:* one-level grid type, $100 \times 100 \times 100$ basic grid.

CHAPTER 5

A comparison with the Tardis code

The interactions of packets and matter are crucial for the radiative transfer calculations. Therefore a verification of implemented rates must be done. The most acceptable way is a comparison with another code; assuming the same input model of an atmosphere, to calculate emergent spectra and compare results by our MC code and results by the Tardis code.

The Tardis code (Kerzendorf & Sim, 2014, Vogl et al., 2019, Kerzendorf et al., 2019) is a supernova spectrum synthesis code. Several simple test cases were chosen. The physical properties of the model can be simply specified in a configuration file: the lower boundary conditions: stellar luminosity (in $\log L_*$); the structure of the model (velocity which give the model physical size, density structure, the chemical composition, matter properties (LTE, NLTE), line interaction type. The characteristics of models are described in the following paragraphs.

The outflow properties are established via the standard Tardis input. It enables to choose physical characteristics defined with simple analytical formulae. The density profile is chosen in a form

$$\rho(v, t_0, t_{\text{expl}}) = \rho_0 \left(\frac{t_0}{t_{\text{expl}}} \right) \left(\frac{v}{v_0} \right)^\zeta. \quad (5.1)$$

In this equation, ρ_0 is density in the lowest boundary shell (in a point $v = v_0$) at an initial time t_0 , t_{expl} is a time parameter and the exponent ζ has to be chosen by a Tardis user. The Eq. (5.1) is time dependent, but both Tardis and our code calculate a single time frame. We selected $t_0 = 1$ day, $t_{\text{expl}} = 13$ days, and $\zeta = -2$. The Tardis code assumes the homologous velocity field (i.e., Röpke, 2005, Kurfürst et al., 2020, Kurfürst & Krtićka, 2019, Pejcha et al., 2022) which takes a form

$$v(r) = \frac{r}{R_\infty} V_\infty. \quad (5.2)$$

Here, R_∞ is the outer boundary, V_∞ the terminal velocity and r an arbitrary radius. The excitation and ionization equilibrium are calculated using the LTE approximation, see Appendix. B.

The Tardis atomic data for hydrogen and helium are sourced from the Kurucz database. They differ from the atomic data in the Chapter 4. This database contains hydrogen and helium atoms. Table 5.1 provides basic informations about number of included levels.

The calculated input models are based on the standard model properties of the Tardis code. Both the Tardis and our code assume the Sobolev approximation, therefore, all line profiles are assumed to be delta functions

$$\phi_i(v) \propto \delta(v - v_i), \quad (5.3)$$

where i denotes the i -th line and v_i is the transition frequency.

We focused on the line interaction firstly. There are three possible approximations of the line interactions in the Tardis code. The most simple approximation is denoted as ‘scattering’; coherent

Ion	Max. principal quant. num.	N of levels	N of transitions
HI	10	55	75
HII	1	1	0
HeI	6	53	175
HeII	10	55	67
HeIII	1	1	0

Table 5.1: Numbers of atomic levels and atomic transitions included.

number of packets	4×10^7
propagation grid	regular, 150^3
input luminosity	$10^{9.44} L_{\odot}$
velocity start (v_0)	5000 km/s
velocity end (v_{∞})	30000 km/s
inner boundary	$8072.4 R_{\odot}$
outer boundary	$48434.6 R_{\odot}$
uniform abundances	ab.(H) = 0.89, ab.(He) = 0.11

Table 5.2: Set of parameters used for the test case spectra calculation.

scattering is enabled, a packet is re-emitted without any other inter-processes included. More complex approximation, ‘the downbranch’ mode, enables fluorescence. Re-emission in the same frequency is enabled, in addition any other radiative transition from an upper level to lower level is possible as well. The last but not least approximation is the ‘macro atom’ mode, enabling internal transitions among levels.

The concrete values of parameters are listed in the Tab. 5.2. It was chosen to create as simple model as possible. The procedure of calculation is following. First of all a Tardis model was calculated with the parameters in Tab. 5.2. The Tardis code provides two main output files: the calculated model and the emergent spectrum. The model should contain as many common input parameters as possible. Our case include following informations for each radial shell: radius, velocity, temperature, density and electron density. Those data are loaded by our code and a spectrum is calculated. Then, the output spectrum from our code is compared with the output spectrum from the Tardis code. The compared spectra are plotted in Fig. 5.1, 5.2, and Fig. 5.3.

Analyzing differences between our code and the Tardis code one can see, that strong Lyman lines and weaker Balmer lines are present. Flux in continuum does not differ, however, flux in lines is more pronounced, which is visible in the plots of relative differences (lower panels of Figs. 5.1, 5.2, and 5.3).

Spectra of our code and the Tardis code are relatively shifted, which is visible in the blue wing of the hydrogen $\text{Ly}\alpha$, therefore a large difference peak in fluxes is visible close to the wing wavelength, however, this difference disappear whilst our spectrum is corrected. Spectra before shift are plotted in the lower panel of Fig. 5.3. $\text{H}\alpha$ changes due to velocity shift as well, furthermore the differences in flux increases with more complex line treatment, whilst differences are small in the case of resonance scattering included, the macro atom case provides larger differences in the calculated line profiles. Several possibilities can explain these differences. The Tardis code calculates the propagation of packets in spherical coordinates, whilst our code makes calculations in the Cartesian grid which ‘pixelizes’ the input model of an atmosphere. To check this sensitivity of our calculations to a grid

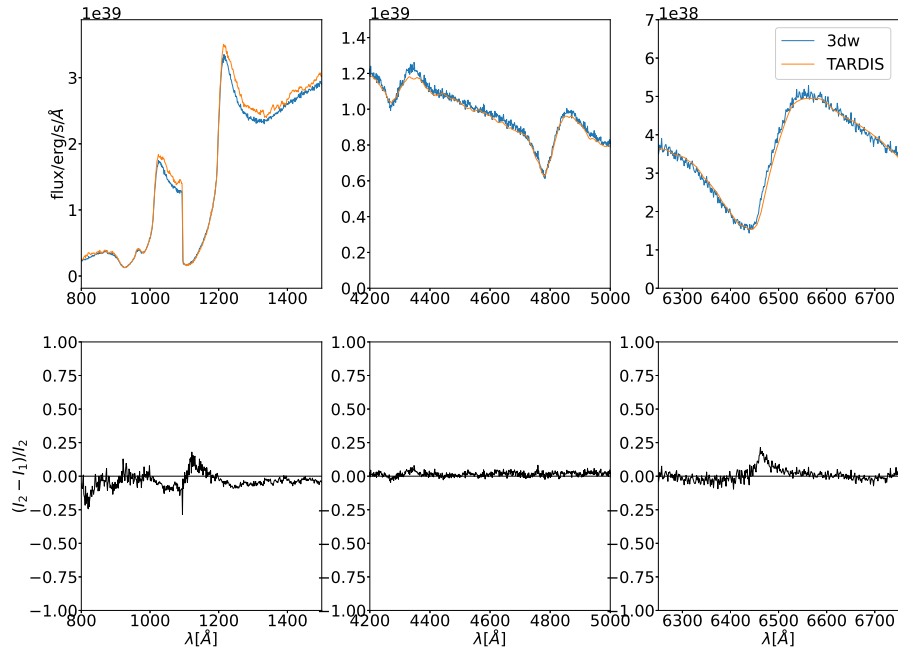


Figure 5.1: Comparison of spectra calculated by our MC code and by Tardis code. *Upper panel:* spectra comparison, *Lower panel:* relative difference between spectra and fluxes. Model in the ‘scattering’ mode.

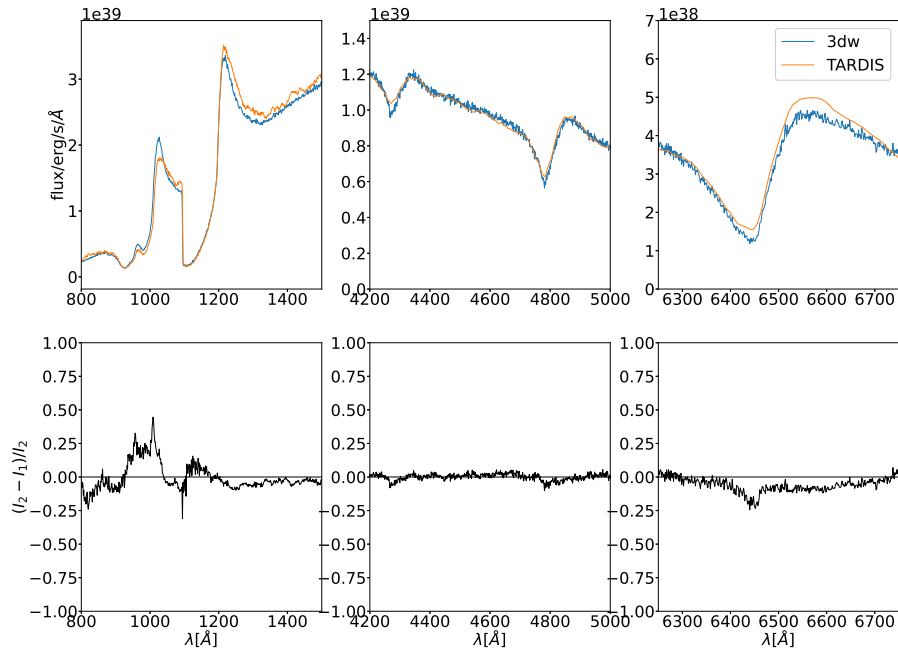


Figure 5.2: Comparison of spectra calculated by our MC code and by Tardis code. *Upper panel:* spectra comparison, *Lower panel:* relative difference between spectra and fluxes. Model in the ‘downbranch’ mode.

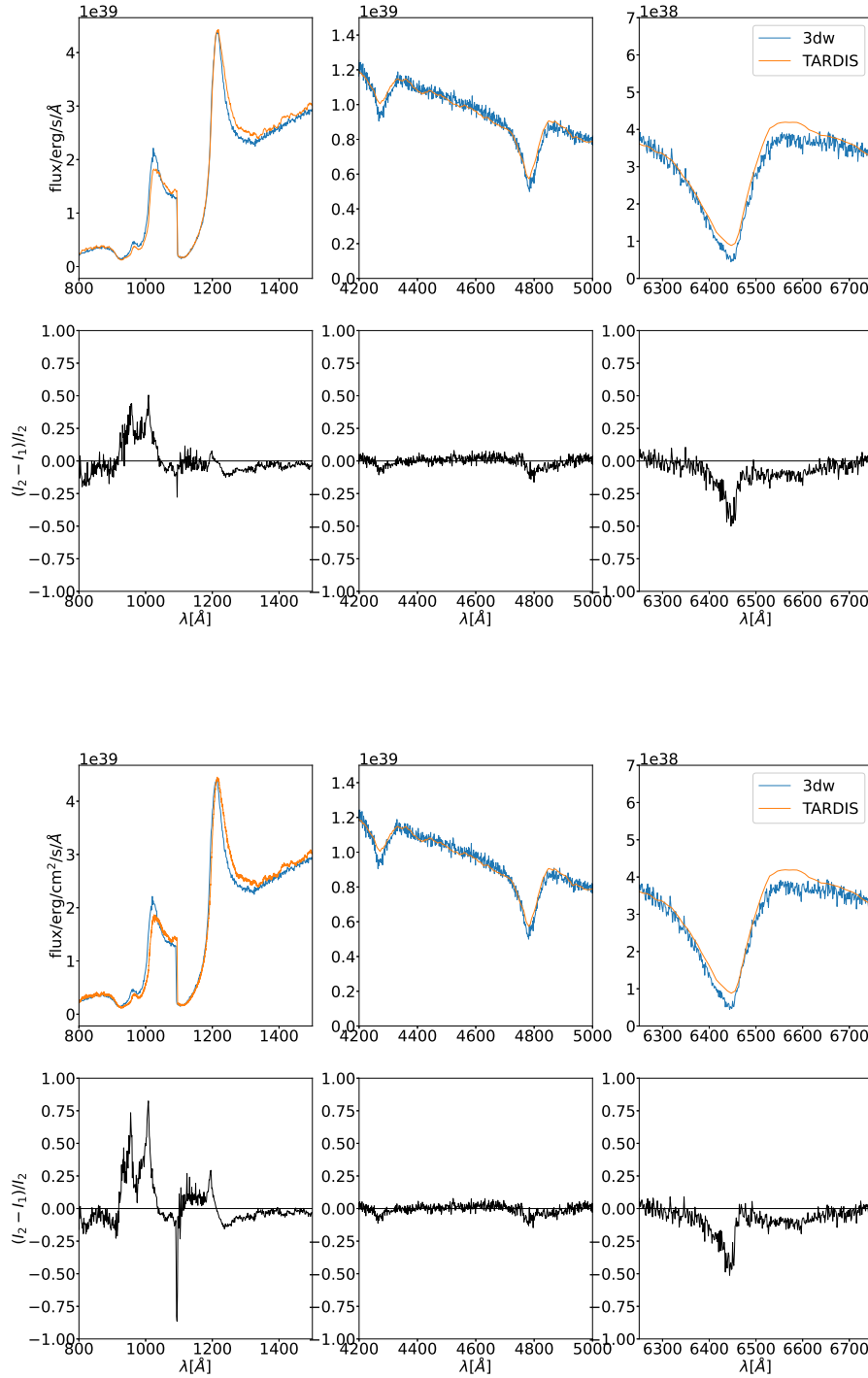


Figure 5.3: Comparison of spectra calculated by our MC code and by Tardis code. *Upper panel:* spectra comparison, *Lower panel:* relative difference between spectra and fluxes. Model in the ‘macro-atom’ mode.

resolution, we calculated several spectra with different propGrids. The spectra do not significantly change for basic propGrid division above $N_B > 50$.

We calculated additional tests to understand the differences better, along with runs a sensitivity of spectra changes on several parameters was investigated. The largest changes were observed whilst parameters V_∞ and R_∞ were changed causing large shifts of $\text{Ly}\alpha$ and the other Lyman lines. Other lines, such as Balmer lines were not that sensitive, which means their shifts were not as significant as the Lyman series lines. The lower boundary condition was tested, concretely the temperature of Planck radiation of the lower boundary condition, therefore the whole continuum was changed and no better agreement was observed. Another lower boundary condition is a stellar radius which affects the effective temperature and did not explain any differences. The tests including the input supernova model change did not bring a significant change.

There are observed differences between matter physical properties, such as electron densities, which differ by 0.7% in maximum, and ionization factions which differ about 1%. This affects the opacity structure and, consequently, the Monte Carlo rates.

CHAPTER 6

Basic applications in multidimensional cases

This chapter presents basic applications of our code in the 3-D input hydrodynamical models. A calculation of radiative transfer through a 3-D environment is the main goal of the whole code development. We will be able to figure out importance of 3-D structures and their contribution to the calculated emergent spectra.

Although an adaption of our MC code into multidimensional models is not as difficult as in the case of classical case of radiative transfer methods, 3-D model calculations bring new complications and challenges. The main challenge lies in a large amount of modGrid points, consequently, longer time needed to read input file and more memory to save modGrid. The most consuming part is a connection between modGrid and propGrid; the closest modGrid cell must be found for every propagation cell which means to check every modGrid cell and find out the one cell with the shortest distance. The number of cycles is enormous even for smaller sized grids. Furthermore, the calculation of ionization fractions and electron density is time consuming too. This problem can be partially solved with parallelization or by using mentioned propGrid, with well defined associated cells, which means that indexes of cells are the same in both cases, this is applied in the first test case.

The first section of this chapter describes an artificial 3-D model. This model is created to be numerical efficient and to find out any numerical uncertainties which could have been developed because of transformation into the 3-D. The second section describes a calculation of radiative transfer in a 3-D hydrodynamical model. Problems connected to implementations are included as well.

6.1

A simple testing 3-D model

A step between 1-D (input) models and 3-D input models is a pseudo 3-D model. It is a 1-D model defined in a 3-D grid input file. This arrangement simulates some parts of 3-D input models: very large input files which prolong the time needed for a connection between these two grids, however our test grid is simple enough since our input modGrid points are regularly distributed in space. This fact prevents long calculations of associated modGrid cells for each propGrid cell because it is created and numbered with same indexes in both cases.

Physical properties of the model are following. Velocity field is described with homologous expansion (linear function), see Eq. (5.2) and density profile is $\rho(r) \propto \rho_0 r^{-2}$. The ejecta are isothermal with temperature $T = 15500\text{K}$ in every modGrid cell. Ejecta are composed of hydrogen (number

number of packets	2×10^7
grid size N_B	$50 \times 50 \times 50$
effective temperature T_{eff}	14734 K
stellar radius R_*	$8R_\odot$
outer boundary R_∞	$10R_*$
outer boundary velocity V_∞	$2.875 \times 10^9 \text{ cm} \cdot \text{s}^{-1}$
density ρ_0	$10^{-14} \text{ g} \cdot \text{cm}^{-3}$
isothermal matter temperature	15500 K

Table 6.1: Numerical parameters of calculated testing model.

fraction 0.89037) and helium (number fraction 0.10963) with constant abundance in the whole computational domain.

6.1.1

Velocity calculation

Velocity field should be described by a continuous function, since the Sobolev resonance points are calculated. If the field was constant in a propGrid cell, it would be impossible to calculate whether a packet can interact in a line or not. More complicated 3-D models supply complicated velocity fields without a possibility of use of an analytical formula. However, the velocity in every point is needed to calculate based on the closest modGrid points with known velocity.

One method is called the Trilinear interpolation, see [Press et al. \(1992\)](#). Let us consider an arbitrary point inside the propGrid. Eight points with defined velocity vectors are chosen; it creates a hexahedron, in general case, non-regular shaped. The point must be located inside this hexahedron. Linear interpolation is calculated three times between doublets of points in each step. Interpolation points are chosen to be equal to one of the point coordinates each step. The first step will create four points with a common $x = x_0$ coordinate. Two points with $x = x_0$ and $y = y_0$ are created and the last step the resulting velocity for the point is finally calculated. This procedure is displayed in [Fig. 6.1](#).

A non-trivial task is a choice of eight points for the trilinear interpolation. There are two possibilities in our code. The first possibility is the regular propGrid. The points are located in geometrical centers of eight propGrid cells. The propGrid cells are chosen with a respect to the current cell. The first point lies in the centre of the current cell. Others are chosen in respect to a relative position of the point – a corresponding quadrant which contains the point. Another case – adaptive propGrid is not as simple as the previous case. The points are created respectless on neighboring propGrid cells. The points lie in corners of a cube which width is the same as the width of the current cell. One corner is in the center of the current propGrid cell, other corners are located in a quadrant which contains the point. Velocities in the points are the modGrid velocities, basically, velocities of associated modGrid cells for a propGrid cell. If the cell is located outside the modGrid the velocity is set to zero.

As a first step to test an implementation of this method, we calculated a 1-D¹ model with a velocity field generated with an analytical function. In the positions of interpolation points velocity vectors were calculated:

$$\vec{v}(r) = v(I) \frac{\vec{r}}{|r|}, \quad (6.1)$$

¹The physical parameters are as the same as described above, except points are defined as a function of radius, instead of a general coordinates x , y , and z .

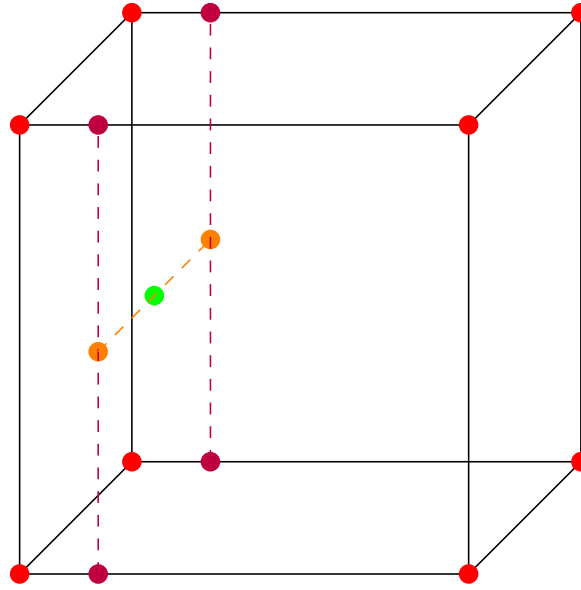


Figure 6.1: Procedure of the trilinear interpolation based on known velocity vectors in eight points. We are calculating a velocity in the green point. The first step is a calculation of linear interpolation between each two red points to obtain the purple points. The second step is calculation of interpolation between the purple points, which calculate values in the orange points, and, finally, the last linear interpolation between the orange points, which calculates a vector in the green point.

where $v(I)$ is the velocity magnitude in the I -th modGrid cell (in this specific case, sinus is the source function), \vec{r} position of the point. Compared values are plotted in the Fig. 6.2 for the regular propGrid case. A calculation was done with fifty packets.² Every point corresponds a velocity determination, generally, in a position \vec{r} ; the points were plotted as a function of radius, since the model is spherically symmetric. There are two cases differing in a number of modGrid points. The first (upper two panels) model consists of 36 model points, the second one consists of 475 model points. The radial distance between neighbour shells is equal and it is equal to $0.25 R_*$ (the first case) and $0.02 R_*$ (the second case). The modGrid is spherically symmetric, however the procedure of interpolation of velocity is as the same as in general 3-D case. The propGrid consists of cells with widths: 50^3 : $w = 0.4 R_*$, 100^3 : $w = 0.2 R_*$ and 200^3 : $w = 0.1 R_*$. Spectra are plotted in Fig. 6.3. Spectra do not differ significantly, especially the cases $N^B = 100^3$ and $N^B = 200^3$ are almost identical.

A good agreement between numerical and analytical magnitude of velocities can be clearly seen. The largest differences are close to both modGrid boundaries, lower and upper. The velocities outside the modGrid grid are set to zero by default and clearly, if an interpolation is calculated among points outside the modGrid domain, at least at one point velocity is equal to zero. As a consequence, a neighbouring zone to the lower or upper boundary converges to the analytical value gradually. The size of the zone depends on the propGrid cells width. The convergence towards the analytical function is steeper in the case of smaller propGrid cells, since the propGrid cells beyond the numerical domain are used for the trilinear interpolation in significantly smaller area. Another effect can be seen in Fig. 6.2. In the case of too dense propGrid, a relative error seems to be staircase shaped. This effect is caused by several propGrid cells belonging to the same modGrid cell, therefore with the same magnitude of velocity. The velocity field seems to be constant among those cells. It is not better to calculate with a finer propGrid in general, it is limited by the modGrid as well.

²During every packet propagation is velocity calculated many times alongside the whole packet trajectory; as a result, even for fifty packets the total number of velocity points is sufficient.

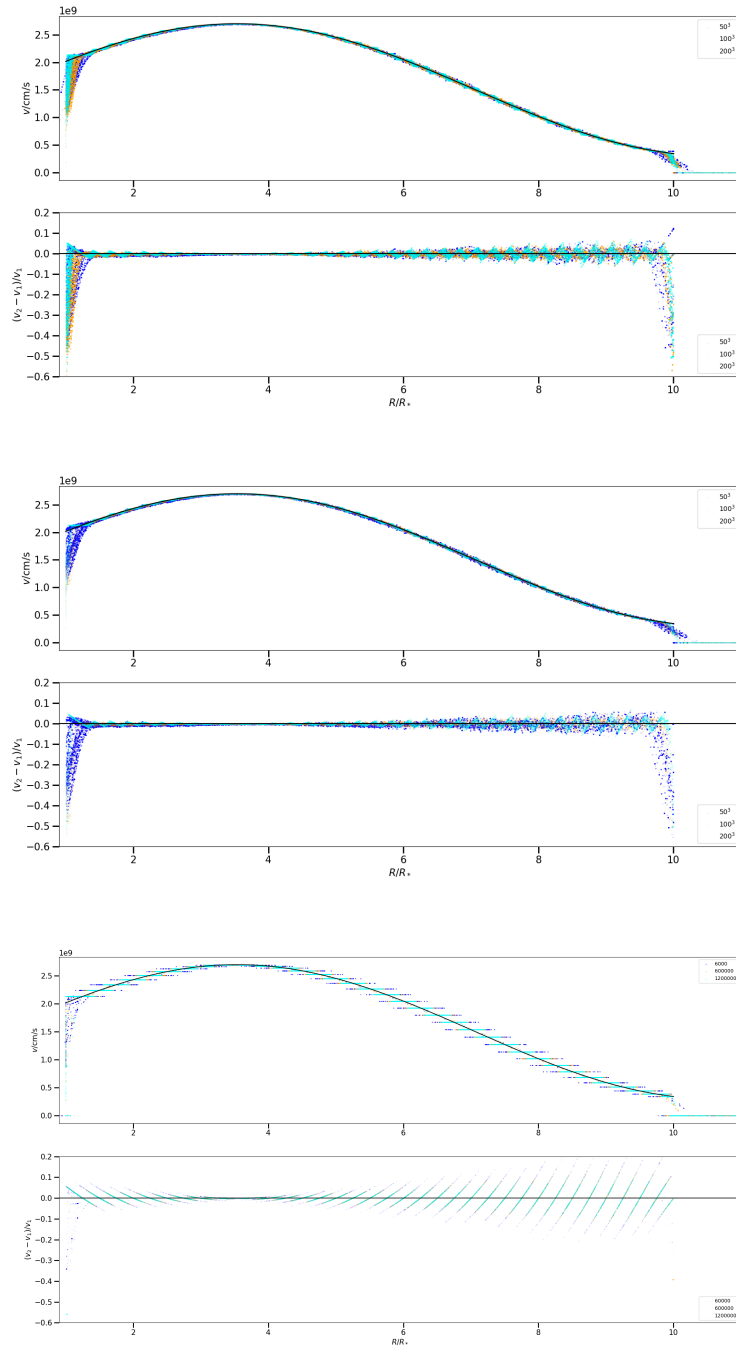


Figure 6.2: A comparison of velocity calculated with the trilinear interpolation and an analytical function for several basic propGrid resolutions. Two upper panels: modGrid with 36 model points, Two lower panels: modGrid with 475 model points.

The staircase profile of velocity magnitudes is more clear in the case of one-level adaptive propGrid, the bottom panel in Fig. 6.2. The main reason is the method of the velocity interpolation depicted in Fig. 6.4. An error is reduced with a higher number of virtual points; this type of adaptive propGrid seems does not to be efficient in the case of spherically symmetric models of winds.

6.1.2

Emergent spectra

We calculated spectra for two cases: the first one homological velocity field is calculated as an analytical function of radius, see Eq. (5.2). In the second case, the same model with velocity interpolated from discrete points was calculated, therefore, the same spectra should be produced. The results are plotted in the Fig. 6.5. The spectra fit perfectly, which implies that procedures containing velocity interpolation seem to work properly. We can test more complex models.

We can test spectra sensitivity based on the density of modGrid points (cells). Calculated spectra are plotted in the Fig. 6.6. The line profiles are changing similarly to the tests in the Chapter 4; shallow line profiles occur for a low resolution grid, deeper profiles occur for increasing resolution of modGrid. In comparison to the referenced chapter, the modGrid parameters are changed and a finer resolution in the propGrid is not capable to achieve emergent spectra as good as in the cases of better modGrid.

6.1.3

Computational details

3-D models are very computational expensive – both time and memory consuming. It is a considerable problem for every method and the Monte Carlo method is not an exception.³ The physical variables must be saved for every grid point (modGrid cell).

This section is assuming the most simple case of a 3-D model – grid points create a regular grid. Furthermore it is created in the same order as the propGrid and the connection between modGrid and propGrid is quite simple – an index of modGrid cell is equal to the propGrid index. A number of data depends on the third power of a grid resolution (in a direction x , y or z) therefore the amount of consumed memory increased rapidly in case of large grid resolution.

6.1.4

Results

This model was created to detect any problem resulting from the transition from 1-D model to multidimensional models. The efficiency was achieved by using a regular modGrid. This modGrid was very simply fitted by the propGrid which reduced the computational time significantly.

Non-trivial part of the transition is the trilinear interpolation of a velocity field. Velocity fields are complicated and it is not possible to use an analytical formula. Consequently, we must use velocity vectors saved in discrete points. Furthermore we still need semi-continuous velocity field, since we calculate Doppler shift in an arbitrary point of the propGrid cell to determine whether a line interaction is possible or not. The test case was chosen to be spherically symmetric but velocity values were determined by using the trilinear interpolation among eight points. The accuracy of this method is satisfactory, a relative error is not larger than 5% in the inner parts of computational domain. The edge parts are affected with boundary conditions – velocity in zones $r < R_*$ and $r > R_\infty$ is set to zero

³I can compare possible calculated models on a laptop with AMD processor (3.5 GHz) and RAM 5.7 GB. It is possible to calculate 1-D model with $N^B = 300^3$. In contrast to this case, a 3-D model was possible to calculate only up to $N^B = 75^3$.

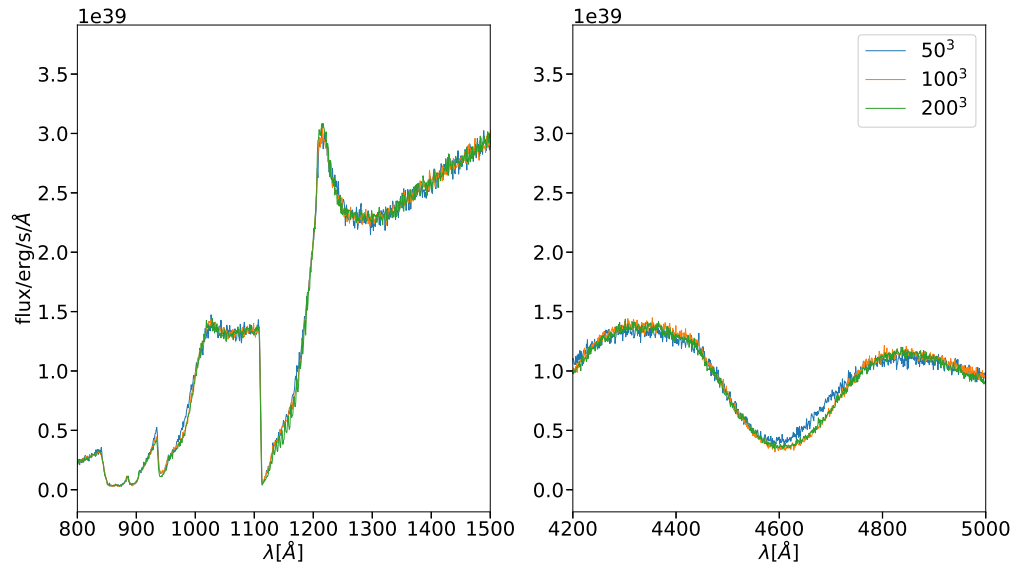


Figure 6.3: Comparison among three spectra in a case of three different basic propGrid resolutions.

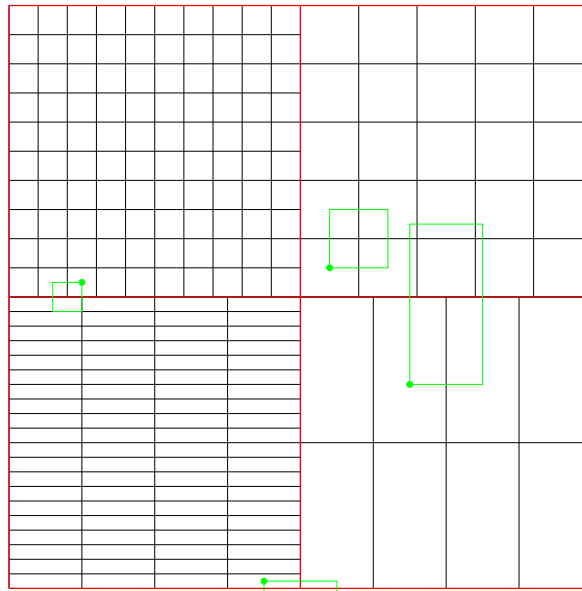


Figure 6.4: Illustration of a choice of the eight points for the trilinear interpolation, in a case of the nested-propGrid type. Velocities assumed in the corner points are considered to be modGrid velocities. In the case, the propGrid subcells do not differ substantially, no large difference occurs. If an interpolation starts in a large propGrid subcell and other corners are placed in a different propGrid subgrid with significantly finer division, a difference may occur. However, it is not an usual case, see the next section.

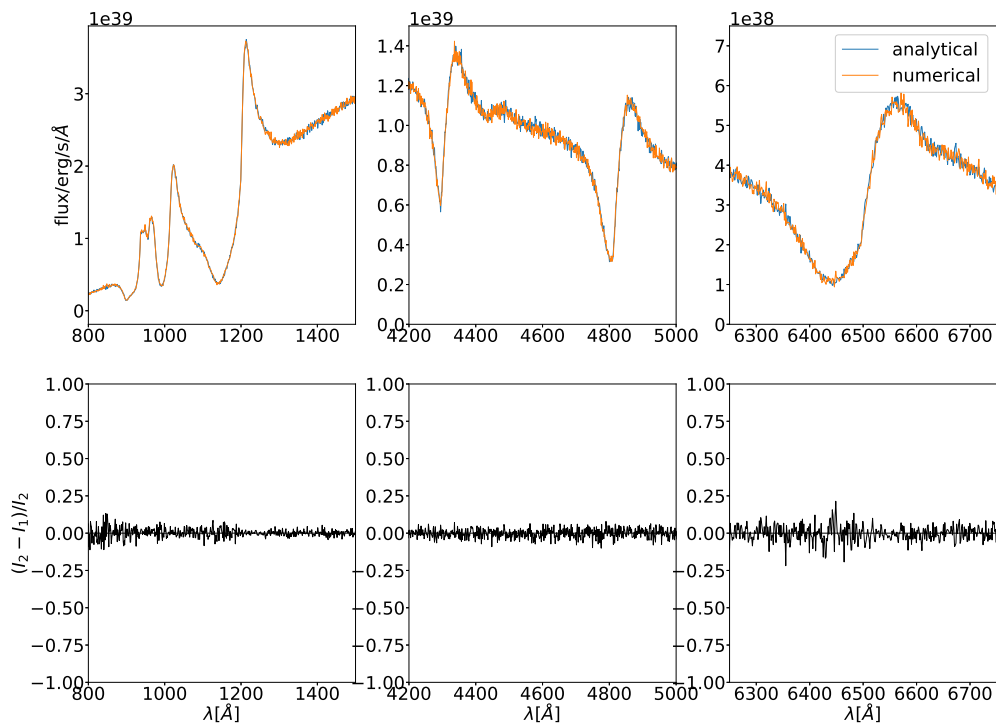


Figure 6.5: A simple test to compare spectra calculated with pure analytical velocity field and velocity field read from a file.

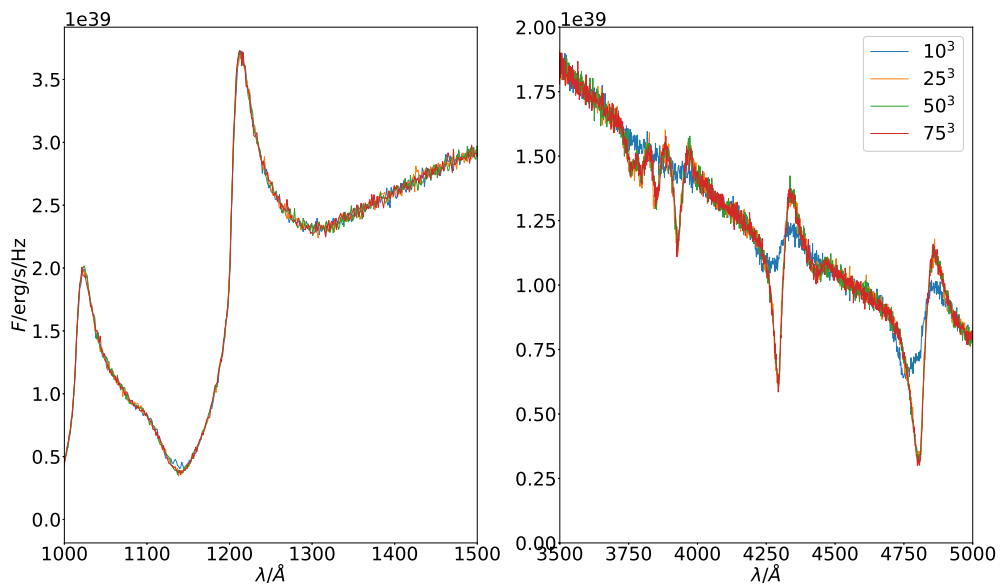


Figure 6.6: A comparison of spectra for several modGrid sizes (also propGrid sizes, since the strict one-to-one correspondence is valid).

and as a result of this choice the trilinear interpolation starts from/converges to zero in neighboring propGrid cells. This effect can be minimized using denser propGrid, on the other hand too dense propGrid creates zones with constant velocity fields, since the values are not interpolated, only taken over from the closest modGrid cell. Too dense propGrid creates a step function, which is visible in Fig. 6.2.

The emergent spectra of 1-D model and pseudo 3-D model were compared. The agreement between spectra is excellent; it seems that the only differences are caused by random noise.

This pseudo model is a bridge between 1-D models and full 3-D models. It is usable to quickly trace problems occurring during calculations with much more complicated 3-D models and quickly debug incriminated parts of the code, if necessary.

6.2

Hydrodynamical model MPI-AMRVAC

The following application is a solution of radiative transfer with an input model calculated with a code MPI-AMRVAC: a flexible (magneto-) hydrodynamics code (Xia et al., 2018, Keppens et al., 2021, Moens et al., 2022a,b). The computation domain is a block-shaped volume named ‘box-inwind’ approach (e.g., Sundqvist et al., 2018, Moens et al., 2022b). The lower boundary is located deeply inside the stellar atmosphere; matter is optically thick – mainly because of free electron scattering opacity. The computation domain upper boundary is located in the zone of a supersonic wind.

The model couples hydrodynamics with a simple radiation model. The hydrodynamical equations: the conservation of mass (1.4), the conservation of momentum (1.5) and the conservation of energy (1.6). This set of equations is closed by equation

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho v^2, \quad (6.2)$$

where $\gamma = c_p/c_v$ is an adiabatic coefficient defined via heat capacities. The gravitational force per a mass unit is assumed in a form

$$\vec{g} = G \frac{\rho M_*}{r^3} \hat{r}, \quad (6.3)$$

ρ is mass density, G is the gravitational constant, M_* the core mass, and \hat{r} a vector, with a magnitude r . It is assumed that the mass is concentrated in the stellar core, while the mass inside the atmosphere and wind is several orders lower. The coupling of gas and radiation is described by the equation

$$\frac{\partial E}{\partial t} + \nabla \cdot (E\vec{v} + \vec{F}) = -\dot{q} - \nabla\vec{v} \odot \mathcal{P}. \quad (6.4)$$

Here, E is the frequency-integrated radiation energy density, \mathcal{P} is the frequency-integrated radiation pressure tensor, \dot{q} includes heat and cooling terms, and, finally, \vec{F} is the frequency-integrated radiation flux. The operator ‘ \odot ’ is a scalar dot of two tensors ($\nabla\vec{v} \odot \mathcal{P} = \partial_i v_j p^{ij}$). Since the environment is optically thick, the nonequilibrium flux-limited diffusion approach is used, see Moens et al. (2022a), the vector \vec{F} becomes:

$$\vec{F} = -\frac{c\lambda}{\chi} \nabla E. \quad (6.5)$$

A variable λ is the flux limiter. The flux limiter is a leading parameter for a radiative transfer treatment: standard Monte Carlo, or a ‘semi-diffusion’ method. Opacity χ is assumed in the form

$$\chi = \chi^{\text{OPAL}} + \chi^{\text{line}}, \quad (6.6)$$

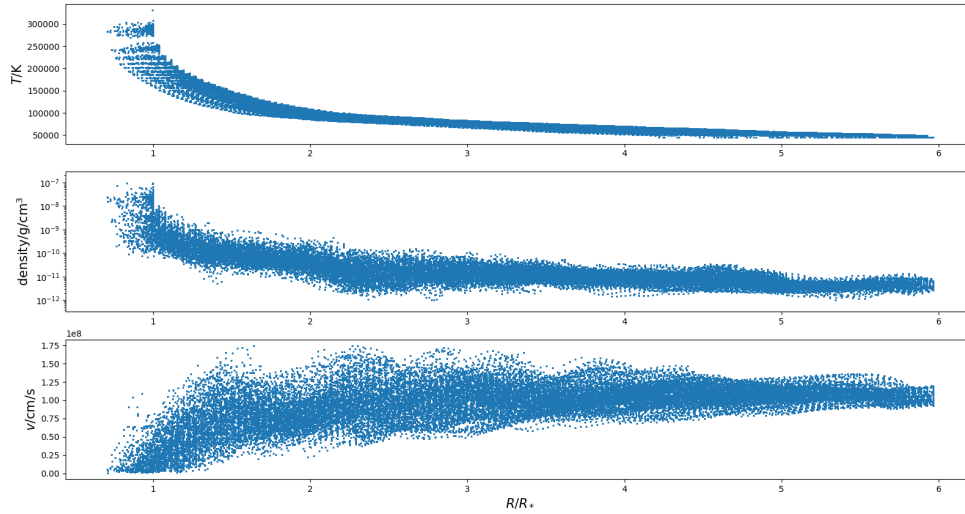


Figure 6.7: Input physical quantities calculated with the hydro-model. From the top to below: temperature, density and velocity magnitude as a function of radius.

χ^{OPAL} is opacity obtained from the OPAL tables (Iglesias & Rogers, 1996); since this term does not describe opacity of a supersonic medium accurately, (e.g., Poniatowski et al., 2021), the term χ^{line} is the total contribution from all lines computed for a supersonic medium, (e.g., Poniatowski et al., 2022, Pauldrach et al., 1998, 2001). However, it is still not full radiative transfer treatment. This is the main reason to use our code to calculate radiative transfer and emergent spectra for this hydro-model.

6.2.1

Set up of model

A model was set up according to Moens et al. (2022a); since an original version of the hydrodynamical model is defined inside a box, an input must be defined in the whole space between R_* and R_∞ .

The input temperature structure is depicted in Fig. 6.7 and 6.8, where the temperature range starts at the values around 300000 K in deep layers; decreases to temperatures 50000 K. The density values fit into a range starting at $10^{-7} \text{ g} \cdot \text{cm}^{-3}$ and reaching $10^{-12} \text{ g} \cdot \text{cm}^{-3}$ at the upper boundary. Finally, velocity magnitude is significantly dispersed, large values can be found inside the photosphere; the upper limit is $1.75 \times 10^8 \text{ cm} \cdot \text{s}^{-1}$, however, the terminal velocity is around $1.15 \times 10^8 \text{ cm} \cdot \text{s}^{-1}$. Furthermore, all important values are listed in the Table 6.2 The atmosphere is composed of helium and carbon, concretely, a helium abundance is assumed to be 0.98 and a carbon abundance 0.02. Atomic data are implemented from the Opacity project (Delahaye et al. (2016)) with detailed informations in Tab. 6.3.

6.2.2

Numerical grids

Mapping of the modGrid onto propGrid is not straightforward, since the modGrid points are distributed irregularly in the computational domain. Not every propGrid cell contains any modGrid

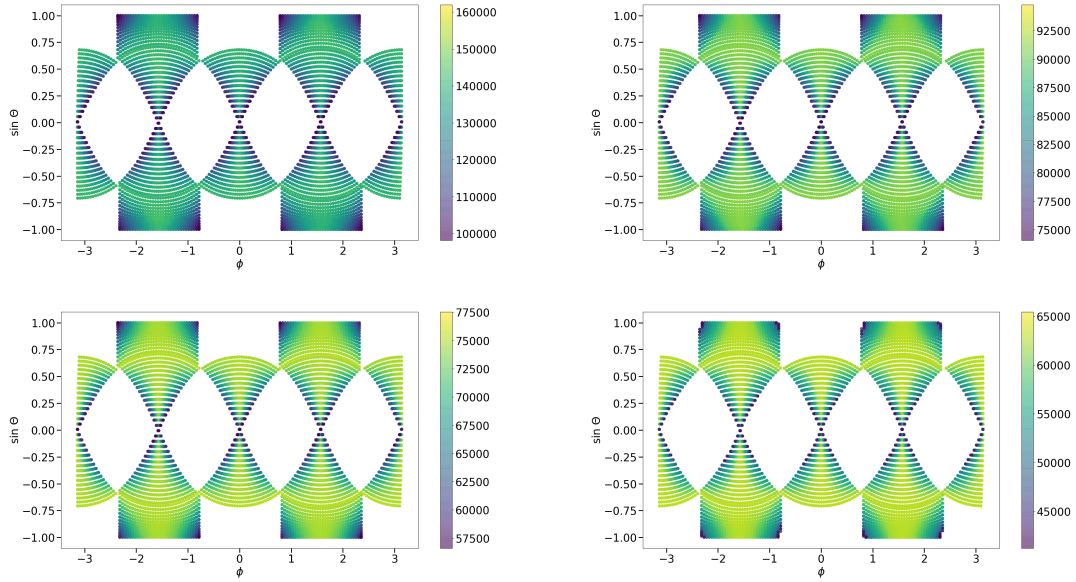


Figure 6.8: Temperature structure in radial slices from the input model.

number of packets	1×10^7
grid size N_B	$100 \times 100 \times 100$
size of computation domain	$6.085R_\odot$
effective temperature T_{eff}	44891 K
stellar radius R_*	$0.707R_\odot$
outer boundary R_∞	$5.966R_*$
outer boundary velocity V_∞	$1.74413586 \times 10^8 \text{ cm} \cdot \text{s}^{-1}$
temperature range	(50000, 300000) K
density range	$(10^{-12}, 10^{-8}) \text{ g} \cdot \text{cm}^{-3}$

Table 6.2: Numerical parameters of calculated testing model.

Ion	Max. principal quant. num.	N of levels	N of transitions
HeI	10	54	315
HeII	10	55	285
HeIII	1	1	0
CI	10	361	4158
CII	10	184	396
CIII	10	137	830
CIV	10	25	153
CV	10	55	315
CVI	10	54	285
CVII	1	1	0

Table 6.3: Numbers of atomic levels and atomic transitions included.

point. A direct connection between modGrid and propGrid leads to a situation shown in the Fig. 6.9. propGrid cell index is calculated for every propGrid cell using the equation (3.2).

A tricky problem lies in the velocity interpolation. The interpolation is tightly connected to the propGrid: the velocity vector connected to the current propGrid centre. It is clear, that in the slice shown in Fig. 6.9 (a velocity vector is defined only for yellow labelled cells) a large amount of non-connected cells occurs, furthermore, a large resolution of the grid leads to more isolated islands of connected propGrid cells, however the propGrid choice cannot affect the propagation and interaction of packets. The whole machinery must be tested to discover all possible disadvantages.

The main problem of vacuum cells (a propGrid cell $R_* < r < R_\infty$ without a connected modGrid cell). is a destroying the velocity structure – the neighbouring cell do not possess an information about velocity, therefore a linear interpolation is not possible. If the velocity in the neighbouring cell is set to zero, then an artificial velocity gradient is created, see Fig. 6.10. The number of those cells must be reduced. A dummy solution in 1-D was simple: finding the closest model point to every propGrid point. 3-D model provides significantly larger number of modGrid points, hence this algorithm is numerically expensive, in the 3-D practically inapplicable. A different approach must be developed in this case, hence a copying of associated cells to the neighbour cells is processed until all propGrid cells have associated cells, see, lower panel of Fig. 6.9.

The adaptive grid offers more efficient solution. The basic propGrid can be chosen significantly less denser in comparison to a case without adaptive grid. Therefore every propGrid basic cell includes modGrid points which are used for the connection the correct modGrid cell to propGrid cell. If the basic propGrid cell do not possess any modGrid point, all cells will be treated as vacuum. However, this method still depends on the choice of the basic propGrid. Another approach can be a simple copy of associated cell from a neighbour propGrid cell. Another improvements are possible due to the velocity interpolation.

6.2.3

A treatment of high-opacity domains

Since the lower parts of model (points with lower radii) are placed deeply in the stellar photosphere, the continuum opacity is dominant in comparison to other opacities. It leads to a large number of scattering processes during the MC calculations, furthermore, the propagation of packets is computationally expensive and inefficient. There are several methods to avoid this problem. One of possible method is a diffusion approximation combined with the MC method, (i.e., Noebauer & Sim, 2019, Fleck & Cummings, 1971, Fleck & Canfield, 1984, Densmore et al., 2007).

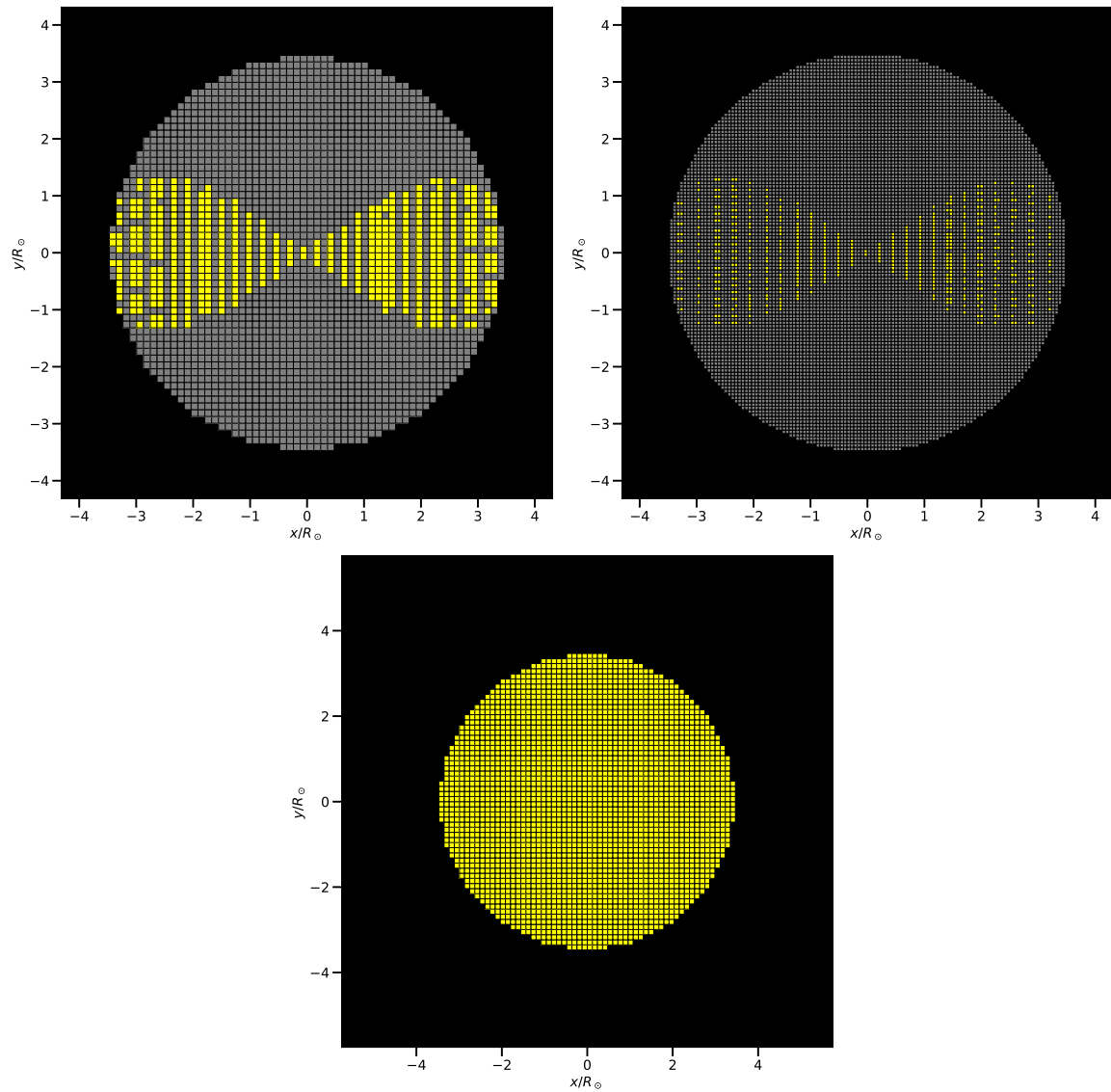


Figure 6.9: Mapping between modGrid and propGrid. A slice in z -plane in a distance $4.78R_\odot$, hence the propGrid cells corresponding to the central star are not located here. Black cells represent propGrid cells beyond the computation domain $r > R_\infty$, grey cells represent vacuum, only yellow cells represent associated propGrid cells, *Upper left panel:* $N^B = 100^3$, *upper right panel:* $N^B = 200^3$. *Lower panel:* full connected propGrid.

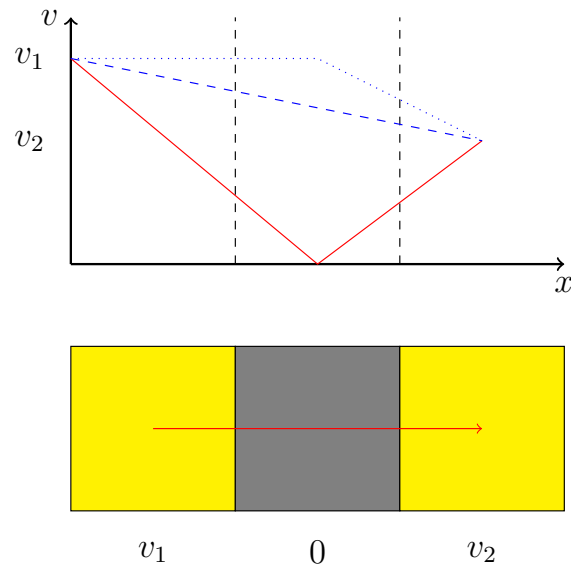


Figure 6.10: Scheme of non-valid velocity interpolation. A packet is propagating from the left propGrid cell to the right, while the middle cell is the vacuum cell. If a velocity in the vacuum cell were treated as zero, an artificial velocity gradient is created (red line). Hence two possible solutions are possible: remember the last velocity vector and calculate the interpolation only between second and third propGrid cell, or interpolate straightly between the right and left cell, which causes less steep velocity function.

In our code, we have implemented the simplest method, which is an acceptable basement for later improvements described in the citations above. The main idea is to avoid the MC calculations inside high-opacity cells with moving the lower boundary upwards. This boundary radial position is dependent on azimuth and declination, hence it is not efficient to set a global lower boundary to be an arbitrary $r > R_*$. First of all, the conditions are chosen appropriately; based on those conditions (which are connected to the modGrid cells) is every modGrid cell labelled as ‘diffusive’⁴ or ‘non-diffusive’.

The ‘non-diffusive’ modGrid cells are treated as normal cells. Otherwise, the ‘diffusive’ process is activated. A d-packet is created. The d-packet possesses information about a current propGrid cell it is located in, current energy which does not change. It does not possess a frequency, nor a precise position. The only process allowed is a change of a propGrid cell. The following procedure depends on the next propGrid cell:

- the next cell is also diffusive – the same procedure is repeated again,
- the next cell is not diffusive – a packet changes into an r-packet with a frequency randomly generated from the Planck’s law and a random direction outwardly from the current propGrid cell,
- the next cell is below lower boundary – a choice is repeated again.

The main parameter is the λ parameter introduced in the Eq. (6.5). Another parameter may be the absorption coefficient. Based on the values, a modGrid cell is labelled before the packet propagation. This parameter does not change during a single iteration. An edge value can be chosen freely and tests are shown in the section below.

⁴In the later context, the title ‘diffusive’ is misleading. This title is held because of future improvements which will lead to a real diffusive approximation.

6.2.4

Preliminary results

We present preliminary results. Many computational aspects will be tested and improved, see the forthcoming section. The presented results are calculated in the LTE approximation and the spectrum is created from all propagated packets. First of all, we present calculated matter properties, and the calculated spectra for simplified model parameters.

Calculated ionization fractions are plotted in Fig. 6.11. The largest ionization fraction is fully ionized. Carbon ionization fraction is more complicated. Fully ionized carbon and five-times ionized carbon dominates in the deepest layers, its value decreases rapidly with radius. The most dominant ion is CIV, other fractions of the ions increase slightly.

We present very first spectra obtained with our code. The first spectra on the Fig. 6.12 are calculated for several values of Planck distribution effective temperatures defined at the lower boundary of the model. The only resonant scattering was allowed, hence every line is P Cygni type. The values of effective temperatures were chosen to be in a standard range of Wolf-Rayet stars. The spectrum consists of several lines, mainly HeII, weaker lines CIV. Line profiles are deeper in the case of large effective temperatures.

The second case includes a propagation of packet allowing a d-packet machinery. The d-packet is activated for modGrid cells with $\lambda > 0.3$. A propagation of a packet is plotted in Fig. 6.13 and the spectrum in Fig. 6.14. The path is not yet efficient, however, no Monte Carlo propagation is being calculated in the d-packet cells. The effective temperature is not constant as well as lower boundary radius depends on the azimuthal and polar angle. The temperature of emitted r-packet (in Planck distribution) depends on the current cell the packet is emitted from. The energy fraction lost in photosphere is zero, furthermore, the energy loss via Thomson scattering is not significant. This approach is a reliable starting point to further improvements, see forthcoming section.

6.2.5

Further improvements

The presented results are preliminary. Many improvements are planned to be done and published in the next paper. The spectra will be calculated in the adaptive propGrid, since its potential to cover the modGrid is larger than the case of regular propGrid.

Two possible ways to prevent unwanted velocity gradients are possible, see Fig. 6.10. The first one is to save the last velocity vector of non-vacuum cell; a velocity in next vacuum cell is treated as a transfer into the propGrid cell with the identical associated modGrid cell (dotted line). A transfer from vacuum cell into the non-vacuum cell: the velocity interpolation can be done between the saved velocity vector and a velocity vector inside the next cell (and a vector from the other propGrid cells needed to calculate interpolation).

The d-packet dynamics will be improved. The diffusive approximation is much more appropriate approach, since the current solution lies in changing of lower boundary condition. Therefore the combination of diffusive and Monte Carlo approach is more efficient than the Monte Carlo approach lone standing. Other improvements are described in the chapter 7.

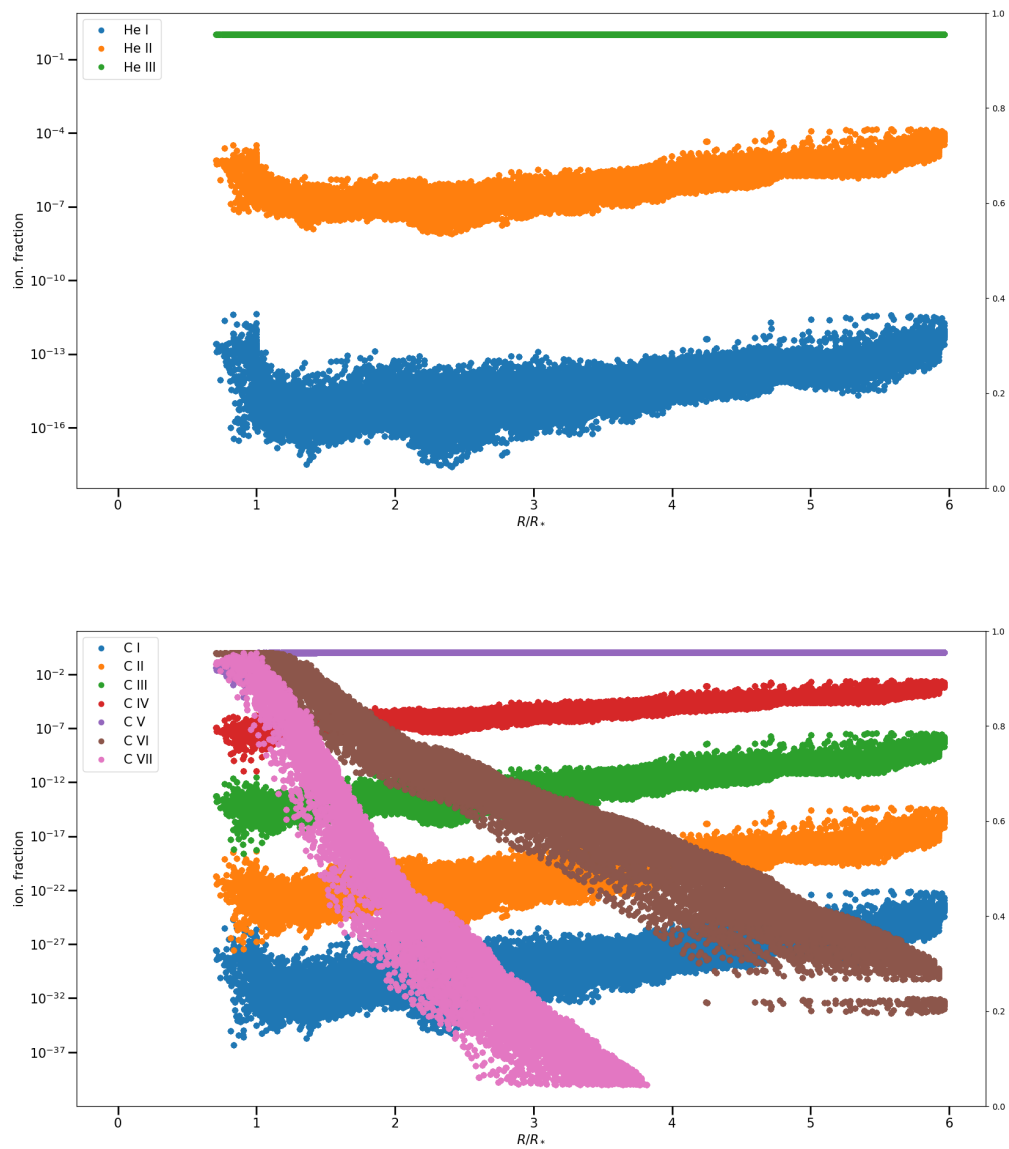


Figure 6.11: Ionization fractions calculated in LTE approximation from the input model. *Upper panel:* Helium ions, *Lower panel:* Carbon ion.

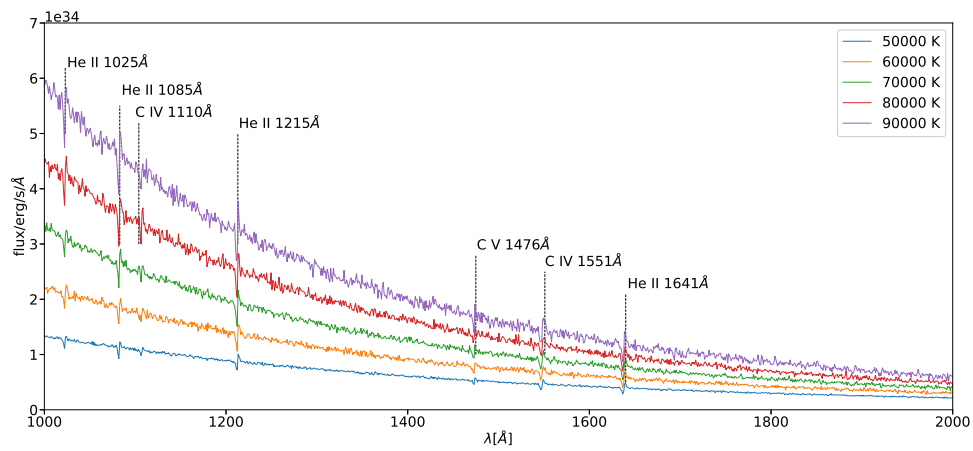


Figure 6.12: Calculated spectra for several effective temperatures of star (lower boundary equilibrium temperature of radiation field).

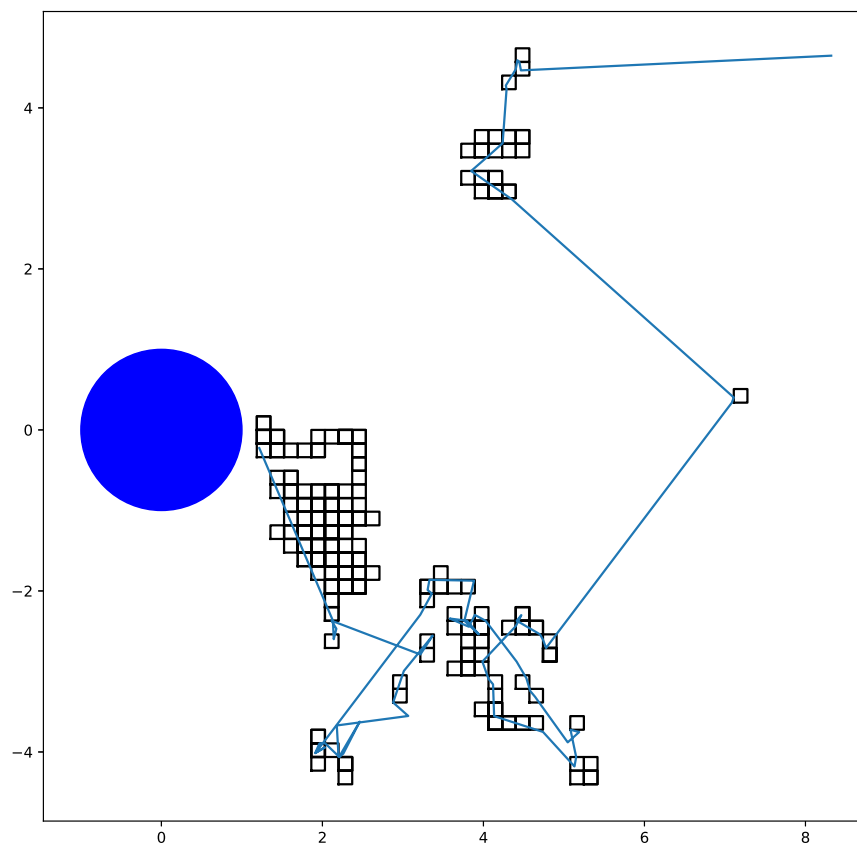


Figure 6.13: Path of a packet consisted of d-packet parts and Monte Carlo parts. The cells represent d-packet cells and blue line represents a path of r-packets.

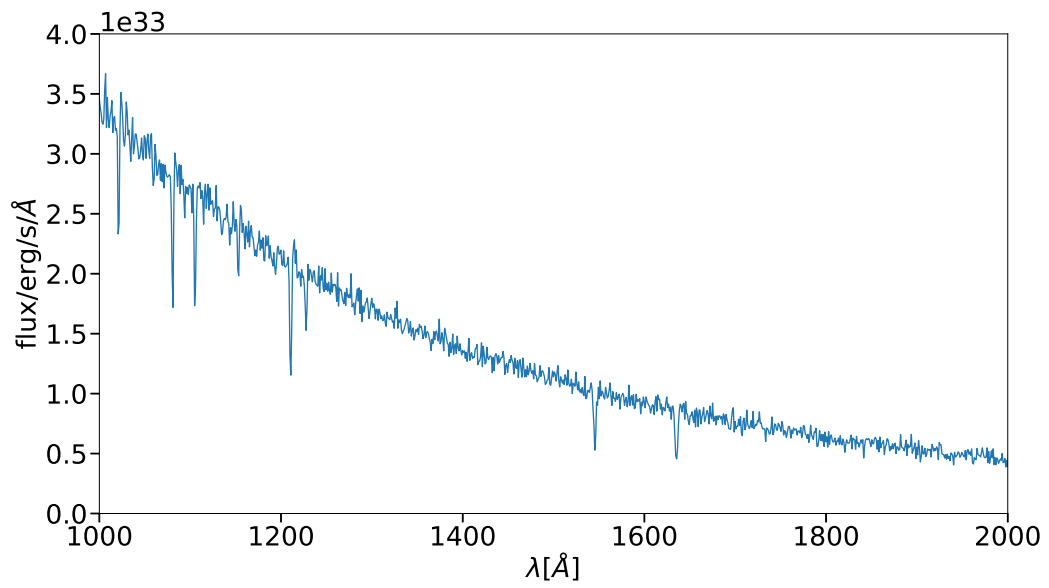


Figure 6.14: Spectrum calculated of a case with d-packets machinery enabled.

CHAPTER 7

Summary

Firstly, we briefly described our code, a method of solving radiative transfer equation, and numerical grids. Several tests were made to figure out the precision of our numerical implementations. To test interaction between packets and matter, we compared our results with results calculated with `Tardis` code. Finally, first calculations of 3-D models were introduced.

The code starts with reading the input model and creates the model grid (`modGrid`), which contains physical properties of the circumstellar outflow. Currently we use a precalculated hydrodynamical (or pure analytical) model of stellar outflow. Then a computation domain is created; the propagation of radiative energy packets (which describe radiation) is processed. This domain is called `propGrid`. Every packet represents an energy which propagates and changes its state until it reaches the outer boundary. The inner and outer boundaries are set up to be consistent with the `modGrid`.

The `modGrid` is the input model of atmosphere. It consists of necessary physical data: size of computation domain (lower and upper boundary); boundary conditions: effective temperature; physical values for `modGrid` points: velocity, density, temperature. The code is not restricted to one input geometry; an implementation of reading a datafile is not a difficult task. The code supports 1-D models, 3-D models (which are not difficult to read) and some 2-D models (not fully implemented yet). The `propGrid` has to be constructed alongside the `modGrid`.

The `propGrid` is a Cartesian grid, where the whole propagation of radiation is processed. The main advantage of using a Cartesian grid is working with one version of geometry implemented into subroutines, therefore, conversion has to be done only during the reading of `modGrid` data. The `propGrid` can be regular or adaptive. The numerics in regular grid is easy to implement, however, it is not efficient. To describe the spatial changes, the `propGrid` cells have to be small enough; including the zones beyond the upper boundary and below the lower boundary. Therefore a large amount of memory is wasted. The efficiency problem is solved with adaptive grid, see 3.4. Two types of the adaptive `propGrid` are created: `octgrid` and `one-level grid`. Both of them are advantageous in different input models.

The `propGrid` is created on a base of `modGrid`, however, individual `propGrid` cells have to be connected with `modGrid` cells. A connection in 1-D model is quite simple, the 3-D case is more complicated, since the number of `modGrid` points can be significantly larger, therefore the consumed time increases rapidly. On the other hand, the `propGrid` and `modGrid` can be saved and loaded for another run again, including their connection.

Plasma is still treated in LTE approximation, it means that ionization and excitation balance is treated with Saha and Boltzmann equation, see B. It is much more simple; more accurate solution–statistical equilibrium equations are planned to be implemented in the nearest future. A simple recalculation of the temperature structure is already implemented into the code; the calculation is based on

a simple model of radiation field and depends on a number of propagated packets. This improvement will be tested and published in forthcoming publications.

The radiation field is quantized into packets, the matter is quantized into macro-atoms, according to [Lucy \(2002, 2003\)](#). The interactions are probabilistic and depend on implemented rates. The verification of implementation was tested via a comparison of emergent spectra calculated with the same input models with another program, the *Tardis* code, which is a supernova spectrum synthesis code. Three basic modes of line interactions were tested: pure scattering (scattering mode), scattering and fluorescence (downbranch mode), and scattering, fluorescence, and internal transitions within a ionization state (macro-atom mode). The agreement between emergent spectra is satisfactory. Differences in continua are negligible, although the differences in lines are more significant.

The Monte Carlo method yields a large advantage for 3-D models calculations. Our code is currently capable to calculate radiative transfer through full 3-D models. These models were presented in the Chapter 6. In the first part, pseudo 3-D models were presented. A transformation from 1-D to 3-D is not straightforward, therefore we discussed several aspect of this transformation. The second section, 6.2, describes the very first implementation of 3-D hydrodynamical model. The brief description of the code is followed by parameters. The input model of atmosphere contains very optical thick domains, which should be avoided with the Monte Carlo solution.

7.1

Future work

The code provides many options to calculate radiative transfer in wide range of input models of atmospheres. In the following list, the major the future improvements are listed. Some of these improvements are crucial for good results, others are optional.

Full Macro-atom approximation

The Macro-atom approximation includes a large amount of transitions. Line interactions were tested, including radiative deexcitation and internal downward and upward jumps in the internal packet mode. A large number of processes is included, however, no tests have been done yet. The tests will be done comparing emergent spectra against another radiation transfer code, similarly to the Chapter 5.

Bound-free processes The bound-free processes are not straightforward to be implemented into the code. The photoionization cross sections are read from files from *Opacity* project and saved into a table including the cross sections for ion levels as a function of frequency. The bound-free processes are both time and memory consuming. The cross section is calculated with the linear interpolation, furthermore, the tables including cross sections have to be stored during the whole calculation.

k-packets The kinetic energy packets represent the collisional processes in the matter. These packets contribute to the matter emissivity, therefore it is important to calculate the cooling rates. In addition, cooling (and heating) rates are necessary to calculate the thermal balance, therefore the gas temperature.

The calculation of rates have to be proceed for every included transitions in each *modGrid* cell. This procedure becomes very computational expensive for the case of large number of included transitions. There are methods to save computation time. One possibility is to save rates of the last *modGrid* cell to save computing time, for a case, whether a packet returns back.

i-packets Several i-packet processes have been already tested, such as radiative excitation, internal upward and downward jumps within a ionization state. Nevertheless, there is a large amount of rates which have not been tested yet: internal upward and downward jumps to higher and lower ionization state, collisional deexcitation, and radiative deexcitation. The main challenge lies in consistent¹ inclusion of van Regemorter approximation.

Calculation of recombination rates are time consuming. Mainly because of linear interpolation of the photoionization cross section. CPU time is partially reduced by precalculation of integrals and creating tables; the integral values are function of temperature. The temperature grid is regular (in a form $T(i) = T_0 + i\Delta T$, i is an index of an interval, T_0 initial temperature and ΔT a difference between temperatures), therefore, for a given temperature is index i calculable directly. When the corresponding interval is determined a linear interpolation can be calculated. The tables are calculated before the packet propagation, thus the calculations of integral (over an exponential function) is not calculated during the i-packet machinery, which saves time.

More precise physical approximations

Several approximations are used during the code development. It is a natural phenomenon, since it is more advantageous to test implemented code on simple case. The reality is usually much more complicated, consequently, laws of physics are equally complicated. We present several aspects of our code, which will be developed in the nearest future.

Ionization and excitation balance Ionization and excitation balance is currently treated in the LTE approximation. It is assumed that certain level of thermodynamic equilibrium is satisfied in short distances, therefore the ionization balance can be calculated only with the Saha equation and a concrete level populations with the Boltzmann equation. This condition is more-less satisfied in the large optical depths. The accuracy of this approximation decreases rapidly in the case of stellar winds, hence NLTE approximation should be used instead. However, we were focused mainly on the implementation of radiative transfer and packet interactions, hence this approximation is acceptable. The NLTE is partially assumed via the Macro-atom approximation implementation, however, the ionization fractions and level populations play crucial role for the rates calculations.

The main task is to implement the NLTE rates calculation, furthermore, new Monte Carlo estimators have to be implemented into the code. The NLTE approximation will enable us to calculate more precise models comparable with observations.

Beyond the Sobolev approximation The Sobolev approximation is held in the current equations implemented into the code. This approximation is valid in the case of large gradient outflows, such as supernova ejecta. It is less valid in the case of stellar winds.

The Sobolev approximation is connected to the profiles of lines. Line profiles are assumed to be delta functions, which is satisfactory condition in the Sobolev approximation. In further developments, more realistic line profiles inclusion is inevitable, since the Sobolev approximation is not valid in every case.

¹Rates are dependent on the change orbital quantum number, which is not straightforward to find whether this number change or from a comparison of two arrays of electronic configuration.

Full diffusion approximation The Monte Carlo method is inefficient in the optically thick zones. A kind of semi-diffusive approximation was already implemented into the code, however full diffusive approximation is capable to calculate more accurate electron temperature and other radiation and matter parameters.

More various boundary conditions The current boundary conditions assume homogeneous surface. These conditions can be changed; the initial distribution function can depend on the surface position.

Spectra of full 3-D models

The current emergent spectra are created out of all propagated packets (through the outer boundary). This procedure is acceptable in the case of spherically symmetric models. In future, this method will not be satisfying enough and we will need spectra calculated for a given direction. Mainly in the case of non-symmetric objects. The backward-ray tracing method will be implemented into the code, the current method is inefficient – much larger number of packets (by several orders) would be needed in the current method.

More various types of objects

It is clear that our code is not bound only to the case of stellar winds. Many tests were processed on supernova ejecta. Other circumstellar objects can be included too. Circumstellar discs are great candidates for an implementation into our code. The implementation has already been done and very first spectra were calculated. This is the main promising part of our project.

Numerical efficiency

The code is developed to be efficient, however, the efficiency does not reach its maximum. Several procedures can be run more optimally, other processes should be parallelized.

Adaptive propGrid properties The results in the Chapter 4 show that the adaptive propGrid is not effective enough; mainly its creation is time consuming. Several steps have been already done: both, propGrid and modGrid can be stored into a file and loaded in another calculation without a calculation, which saves a plenty of time. The connection between the grids has been accelerated. However it is still not efficient enough and more work has to be done in this aspect of the code.

Plasma state calculation The calculation of ionization fractions has to be done for every included modGrid cell. Since the program is parallelized, every core can calculate a different part of model. While LTE approximation is the only possibility, the efficiency would not large, although it increases significantly in the case of NLTE approximation – the ionization balance is solved with the statistical equilibrium equations.

Conclusions

We presented our new MCRT code. This code is being developed to calculate the radiative transfer in 3-D circumstellar outflows and it is still a work in process. We summarize implemented properties and tests which were done. This code is capable to solve multiline radiative transfer in a precalculated hydrodynamical model, which can be 1-D, 2-D, or 3-D. The implemented utilities are: opacity project atomic data, adaptive propGrid, full Lucy Macro atom approach. Furthermore, 3-D models can be calculated with full general velocity field saved in an input file.

The propGrid and its connection to the modGrid was tested in the 4; a simple model with a spherical (single and double) clump was created. A comparison of regular grid, octgrid, and one-level subgrid was done. We tested efficiency of covering modGrid by propGrid, a relative covering, timing and calculated spectra. Firstly the regular propGrid was tested on single spherical clump. Even very dense propGrid does not cover modGrid better than $\langle \chi^2 \rangle = 0.15$ (definition is in the equation (4.5)). Secondly the double clump model was tested. The octgrid seems to be most advantageous in this case. If the basic propGrid is divided sparsely, the creation times are reasonable. The one-level subgrid is efficient too, however, the creation times exceed the octgrid creation times significantly. There is still a large space to improve both of the adaptive propGrids used, which is planned to be done in future.

The main goal of this thesis was to present radiative transfer through the first full 3-D models, therefore, a testing of procedures for the 3-D case had to be done, especially an interpolation of the general velocity field via the trilinear interpolation. Accuracy of this implemented method was discussed; we show that errors are not significant in the case of regular propGrid, and the trilinear interpolation is implemented correctly. Algorithms for the case of adaptive grid have to be further developed.

Radiative transfer through the hydrodynamical model was discussed. The main problem lies in high continuum opacity domains in the lower parts of the model. Since it is not efficient to calculate Monte Carlo radiative transfer in these zones, a method to avoid Monte Carlo calculations was developed. A simple solution was done using d-packets, which move a lower boundary condition to the optically thin zones, which is better to solve with Monte Carlo method.

Two simple test case were calculated. The first case includes different lower boundary effective temperature, the second test case was done with the electron scattering and diffusion approximation included. There are plenty of helium and carbon lines in the spectra. These first results are quite promising, furthermore, a wide range of the future applications of this code are possible, which creates many opportunities to use it.

APPENDIX A

Radiative and collisional rates used in the code

Here we specify detailed expressions for rates used in our calculations. Copied from [Fiřák et al. \(2022\)](#).

A.1

Radiative rates

Bound-bound rates The radiative rate for an upward bound-bound transition from the level i to the level u (excitation) is equal to

$$R_{iu} = B_{iu} \int d\nu \phi_{iu}(\nu) J(\nu). \quad (\text{A.1})$$

The inverse rate (deexcitation from the level i to the level l) is

$$R_{il} = A_{il} + B_{il} \int d\nu \phi_{il}(\nu) J(\nu), \quad (\text{A.2})$$

In these equations, A_{il} , B_{il} and B_{iu} represent the Einstein coefficients, $\phi_{iu}(\nu)$ is the line profile of a transition between levels i and u , and $J(\nu)$ the mean radiation field intensity.

For the case of the Sobolev approximation we approximate the line profile by a delta function,

$$\phi(\nu) = \delta(\nu - \nu_{lu}), \quad (\text{A.3})$$

where ν_{lu} denotes the line centre frequency. Then the absorption rate for transitions from the level i to the level u simplifies to (for derivation see [Klein & Castor, 1978](#))

$$R_{iu}^S = \beta_{iu} B_{iu} J_{lu}, \quad (\text{A.4})$$

where the upper index S stands for Sobolev,

$$J_{lu} = \int d\nu \cdot \phi(\nu) J(\nu) = \int d\nu \cdot \delta(\nu - \nu_{lu}) J(\nu) = J(\nu_{lu}), \quad (\text{A.5})$$

and

$$\beta_{iu} = \frac{1}{\tau_{iu}^S} [1 - \exp(-\tau_{iu}^S)]. \quad (\text{A.6})$$

denotes a probability that a scattered photon escapes. The quantity τ_{iu}^s is the Sobolev optical depth for the transition between levels i and u . Similarly, the emission rate from the level i to the level l is

$$R_{il}^S = \beta_{il} (A_{il} + B_{il}J_{il}). \quad (\text{A.7})$$

where β_{il} is defined similarly to the Eq. (A.6).

If we treat stimulated emission as negative absorption the absorption and emission rates become

$$\tilde{R}_{iu}^S = \left(B_{iu} - \frac{n_u}{n_i} B_{ui} \right) \beta_{iu} J_{iu} \quad (\text{A.8})$$

and

$$\tilde{R}_{il}^S = \beta_{li} A_{il}, \quad (\text{A.9})$$

respectively.

A.1.1

Bound-free and free-bound rates

The radiative ionization rate from the level i to the level p (here the ground level of the next higher ion)

$$R_{ip} = 4\pi \int_{\nu_i}^{\infty} d\nu \frac{\alpha_{ip}(\nu)}{h\nu} J(\nu), \quad (\text{A.10})$$

and the radiative recombination rate (i is the ground level of an ion here, the corresponding population is denoted as n_{i_0}), m corresponds to a level in the next lower ion:

$$R_{im} = \left(\frac{n_m}{n_{i_0}} \right)^* \times 4\pi \int_{\nu_m}^{\infty} d\nu \frac{\alpha_{mi}(\nu)}{h\nu} \left(\frac{2h\nu^3}{c^2} + J(\nu) \right) \exp\left(-\frac{h\nu}{k_B T_e}\right), \quad (\text{A.11})$$

where $\alpha_{mi}(\nu)$ is the photoionization cross section from the level m to the level i . The equation consists of two terms: spontaneous (R_{im}^{spont}) and stimulated (R_{im}^{stim}) recombination rate,

$$R_{im}^{\text{spont}} = \frac{n_{i_0}}{n_i} n_e \phi_i(T_e) 4\pi \int_{\nu_i}^{\infty} d\nu \frac{\alpha_{mi}(\nu)}{h\nu} \frac{2h\nu^3}{c^2} \exp\left(-\frac{h\nu}{k_B T_e}\right), \quad (\text{A.12})$$

and

$$R_{im}^{\text{stim}} = \frac{n_{m_{L_0}}}{n_i} n_e \phi_i(T_e) 4\pi \int_{\nu_i}^{\infty} d\nu \frac{\alpha_{mi}(\nu)}{h\nu} J(\nu) \exp\left(-\frac{h\nu}{k_B T_e}\right). \quad (\text{A.13})$$

Here

$$\phi_i(T_e) = \frac{n_i}{n_{m(0)}^* n_e}. \quad (\text{A.14})$$

is the Saha-Boltzmann factor.

If we assume that the stimulated recombination is negative photoionization, the photoionization rate is given by

$$\tilde{R}_{ip} = \gamma_i - \frac{n_{p_{L_0}}}{n_i} n_e \alpha_i^{\text{stim}}, \quad (\text{A.15})$$

where

$$\gamma_i - \frac{n_{p_{L_0}}}{n_i} n_e \alpha_i^{\text{stim}} = 4\pi \int_{\nu_i}^{\infty} d\nu \left(1 - \frac{n_{p_{L_0}}}{n_i} \frac{n_i^*}{n_{p_{L_0}}^*} \exp\left(-\frac{h\nu}{k_B T_e}\right) \right) J(\nu). \quad (\text{A.16})$$

A.2

Collisional rates

For collisional rates from the level i to the level e (which may be l , u , m , or p , see section 2.5.2) the expressions of C_{ie} for bound-bound and bound-free transitions are given below. The inverse rates are calculated using $n_i^* C_{ie} = n_e^* C_{ei}$ (see Hubeny & Mihalas, 2015, Eq. 9.52).

A.2.1

Bound-bound transitions

The bound-bound collisional rates are calculated using the van Regemorter (1962) approximation

$$C_{iu} = 14.5 \cdot n_e c_0 T^{\frac{1}{2}} \left(\frac{I_H}{h\nu_{iu}} \right) f_{iu} \frac{h\nu_{iu}}{k_B T_e} \exp\left(-\frac{h\nu_{iu}}{k_B T_e}\right) \Gamma\left(\frac{h\nu_{iu}}{k_B T_e}\right), \quad (\text{A.17})$$

(see also Hubeny & Mihalas, 2015, Eq. 9.58), here $c_0 = 5.46510 \cdot 10^{-11}$ is a constant, $I_H = 13.6\text{eV}$ is ionization potential of hydrogen and $h\nu_{iu} = \varepsilon_u - \varepsilon_i$ is the transition energy between levels i and u . The function Γ is defined as

$$\Gamma(x) = \max(\bar{g}, 0.276 \cdot \exp(x) E_1(x)), \quad (\text{A.18})$$

\bar{g} is given by (here n, n' denote principal quantum number, l, l' the orbital quantum number)

$$\bar{g} \sim \begin{cases} 0.7, & \text{for transitions } \langle n, l \rangle \rightarrow \langle n, l' \rangle, \\ 0.2, & \text{for transitions } \langle n, l \rangle \rightarrow \langle n', l' \rangle, \end{cases} \quad (\text{A.19})$$

and $E_1(x)$ is the exponential integral function.

A.2.2

Bound-free transitions

The collisional bound-free rates are calculated using an approximate formula (see Hubeny & Mihalas, 2015, Eq. 9.60)

$$C_{ip} = n_e \frac{1.55 \cdot 10^{13}}{T_e^{\frac{1}{2}}} \bar{g}_i \frac{\alpha_{ip}}{k_B T_e} \exp\left(-\frac{h\nu_i}{k_B T_e}\right), \quad (\text{A.20})$$

where \bar{g}_i is equal to

$$\bar{g}_i \sim \begin{cases} 1 & \text{charge of ionization state of level } i \text{ is } 0, \\ 2 & \text{charge of ionization state of level } i \text{ is } 1, \\ 3 & \text{charge of ionization state of level } i \text{ is } \geq 2. \end{cases}$$

A.2.3

Free-bound transitions

The cross section for the spontaneous free-bound transition from the level i to the level m is

$$\alpha_i^{\text{E, spont}} = \phi_i(T_e) 4\pi \int_{\nu_i}^{\infty} d\nu \frac{\alpha_{im}(\nu)}{h\nu_i} \frac{2h\nu^3}{c^2} \exp\left(-\frac{h\nu}{k_B T_e}\right) \quad (\text{A.21})$$

where the Saha-Boltzmann factor is defined in (A.14).

A.3

Emissivities

The free-bound emissivity

$$\eta_{ip}^{\text{bf}}(\nu) = n_i^* \alpha_{ip}(\nu) \frac{2h\nu^3}{c^2} \exp\left(-\frac{h\nu}{k_B T_e}\right). \quad (\text{A.22})$$

where $\alpha_{ip}(\nu)$ is the photoionization cross-section for a transition from the level i to the level p .

The free-free emissivity

$$\eta_i^{\text{ff}} = n_e N_j \alpha_i^{\text{ff}}(\nu, T_e) \frac{2h\nu^3}{c^2} \exp\left(-\frac{h\nu}{k_B T_e}\right), \quad (\text{A.23})$$

N_j represents the population of the ion j , the free-free cross-section, see (Kromer, 2009, Eq. (3.46))

$$\begin{aligned} \alpha_{i_k}^{\text{ff}}(\nu, T) &= \frac{4e^6}{3hc} \left(\frac{2\pi}{3m_e^2 k_B}\right)^{1/2} \frac{(k-1)^2 g_{\text{ff}}(j)}{T^{1/2} \nu^3} \\ &= 3.69255 \times 10^8 \cdot (k-1)^2 g_{\text{ff}}(j) \cdot T^{-1/2} \times \nu^{-3}. \end{aligned} \quad (\text{A.24})$$

If we put the Eq. (A.23) into Eq. (2.56) we can find easily an analytic formula

$$\nu = \frac{k_B T}{h} \log(z). \quad (\text{A.25})$$

where $z \in (0, 1)$ is a random number.

APPENDIX B

Ionisation and excitation equilibrium

The occupation numbers are in the LTE approximation calculated via the Boltzmann excitation formula, which we use in the form

$$\frac{n_u}{n_l} = \frac{g_u}{g_l} \exp\left(-\frac{\varepsilon_u - \varepsilon_l}{k_B T_e}\right), \quad (\text{B.1})$$

where n denotes the atomic level number densities, g is the statistical weight, ε is the level energy, and the indexes l and u denote lower and upper level of the transition, respectively. The ionization balance follows the Saha equation, here in the form

$$\frac{N_j}{N_p n_e} = \frac{U_j}{U_p} \frac{C}{T_e^{3/2}} \exp\left(\frac{\varepsilon_{p_{l_0}} - \varepsilon_{j_{l_0}}}{k_B T_e}\right), \quad (\text{B.2})$$

where C is the Saha constant

$$C = \frac{1}{2} \left(\frac{2\pi\hbar^2}{m_e k_B}\right)^{3/2}, \quad (\text{B.3})$$

N denotes ion number densities, U denotes the partition function, ε denotes energies, and the indexes p and j denote atomic states. The partition function of the ion i is

$$U_i(T_e) = \sum_{j=1}^{\vec{N}_L} g_j \exp\left(-\frac{\varepsilon_j}{k_B T_e}\right), \quad (\text{B.4})$$

where \vec{N}_L is the total number of included levels for ion i . The sum runs only over the levels included in the calculation.

Bibliography

- Adam, J. 1990, *A&A*, 240, 541
- Baes, M., Verstappen, J., De Looze, I., et al. 2011, *ApJS*, 196, 22
- Barlow, M. J. & Cohen, M. 1977, *ApJ*, 213, 737
- Barnes, J. & Hut, P. 1986, *Nature*, 324, 446
- Biegging, J. H., Abbott, D. C., & Churchwell, E. B. 1989, *ApJ*, 340, 518
- Bjorkman, J. E. & Cassinelli, J. P. 1993, *ApJ*, 409, 429
- Blondin, J. M., Kallman, T. R., Fryxell, B. A., & Taam, R. E. 1990, *ApJ*, 356, 591
- Bulla, M., Sim, S. A., & Kromer, M. 2015, *MNRAS*, 450, 967
- Carciofi, A. C. & Bjorkman, J. E. 2006, *ApJ*, 639, 1081
- Carciofi, A. C., Bjorkman, J. E., & Magalhães, A. M. 2004, *ApJ*, 604, 238
- Carciofi, A. C., Bjorkman, J. E., & Zsargó, J. 2017, in *The Lives and Death-Throes of Massive Stars*, ed. J. J. Eldridge, J. C. Bray, L. A. S. McClelland, & L. Xiao, Vol. 329, 390–390
- Carlberg, R. G. 1980, *ApJ*, 241, 1131
- Carroll, B. W. & Ostlie, D. A. 2017, *An introduction to modern astrophysics, Second Edition* (Weber State University)
- Castor, J. I. 1974, *ApJ*, 189, 273
- Castor, J. I., Abbott, D. C., & Klein, R. I. 1975, *ApJ*, 195, 157
- Chandrasekhar, S. 1934, *MNRAS*, 94, 522
- Delahaye, F., Zwölf, C. M., Zeippen, C. J., & Mendoza, C. 2016, *Journal of Quantitative Spectroscopy and Radiative Transfer*, 171, 66
- Densmore, J. D., Urbatsch, T. J., Evans, T. M., & Buksas, M. W. 2007, *Journal of Computational Physics*, 222, 485
- Dessart, L. & Owocki, S. P. 2003, *A&A*, 406, L1
- Dessart, L. & Owocki, S. P. 2005, *A&A*, 437, 657
- Driessen, F. A., Kee, N. D., & Sundqvist, J. O. 2021, *A&A*, 656, A131

- Feldmeier, A., Puls, J., & Pauldrach, A. W. A. 1997, *A&A*, 322, 878
- Feldmeier, A., Puls, J., Reile, C., et al. 233, *Astrophysics and Space Science*, 233, 293
- Feldmeier, A., Shlosman, I., & Vitello, P. 1999, *ApJ*, 526, 357
- Fišák, J., Kubát, J., Kubátová, B., Kromer, M., & Krtička, J. 2022, *A&A*, in press
- Fleck, J. A., J. & Canfield, E. H. 1984, *Journal of Computational Physics*, 54, 508
- Fleck, J. A., J. & Cummings, J. D. 1971, *Journal of Computational Physics*, 8, 313
- Freytag, B., Höfner, S., & Liljegren, S. 2019, *IAU Symposium*, 343, 9
- Gayley, K. G. 1995, *ApJ*, 454, 410
- Georgiev, L. N., Hillier, D. J., & Zsargó, J. 2006, *A&A*, 458, 597
- Griffiths, D. J. 1999, *Introduction to electrodynamics*, 3rd edn. (Upper Saddle River, N.J.: Prentice Hall)
- Hamann, W. R. & Gräfener, G. 2003, *A&A*, 410, 993
- Hamann, W. R. & Gräfener, G. 2004, *A&A*, 427, 697
- Hennicker, L., Puls, J., Kee, N. D., & Sundqvist, J. O. 2018, *A&A*, 616, A140
- Hennicker, L., Puls, J., Kee, N. D., & Sundqvist, J. O. 2020, *A&A*, 633, A16
- Higginbottom, N., Knigge, C., Long, K. S., Sim, S. A., & Matthews, J. H. 2013, *MNRAS*, 436, 1390
- Hillier, D. J. 1987, *ApJS*, 63, 969
- Hillier, D. J. 1990, *A&A*, 23, 116
- Hillier, D. J. & Miller, D. L. 1998, *ApJ*, 496, 407
- Hubeny, I. 1988, *Computer Physics Communications*, 52, 103
- Hubeny, I. & Mihalas, D. 2015, *Theory of stellar atmospheres* (Princeton, N.J.: Princeton University Press)
- Ibgui, L., Hubeny, I., Lanz, T., & Stehlé, C. 2013, *A&A*, 549, A126
- Iglesias, C. A. & Rogers, F. J. 1996, *ApJ*, 464, 943
- Keppens, R., Teunissen, J., Xia, C., & Porth, O. 2021, *Computers & Mathematics with Applications*, 81, 316, development and Application of Open-source Software for Problems with Numerical PDEs
- Kerzendorf, W., Nöbauer, U., & Sim, S. 2019, *tardis-sn/tardis: TARDIS v3.0 alpha2*, Zenodo, doi: 10.5281/zenodo.2590539
- Kerzendorf, W. E. & Sim, S. A. 2014, *MNRAS*, 440, 387
- Klein, R. I. & Castor, J. I. 1978, *ApJ*, 220, 902
- Knigge, C., Woods, J. A., & Drew, J. E. 1995, *MNRAS*, 273, 225

- Korčáková, D. & Kubát, J. 2005, *A&A*, 440, 715
- Kromer, M. 2009, Doctoral thesis, Technischen Universität München, München
- Kromer, M. & Sim, S. A. 2009, *MNRAS*, 398, 1809
- Krtička, J. 2006, *MNRAS*, 367, 1282
- Krtička, J. 2011, fyzika horkých hvězd II, skripta
- Krtička, J. & Kubát, J. 2004, *A&A*, 417, 1003
- Krtička, J. & Kubát, J. 2009, *MNRAS*, 394, 2065
- Kudritzki, R. P., Puls, J., Lennon, D. J., et al. 1999, *A&A*, 350, 970
- Kurfürst, P. & Krtička, J. 2019, *A&A*, 625, A24
- Kurfürst, P., Pejcha, O., & Krtička, J. 2020, *A&A*, 642, A214
- Kurosawa, R. & Hillier, D. J. 2001, *A&A*, 379, 336
- Kurucz, R. & Bell, B. 1995, Atomic Line Data (R.L. Kurucz and B. Bell) Kurucz CD-ROM No. 23. Cambridge, 23
- Lamers, H. J. G. L. M. & Cassinelli, J. P. 1999, Introduction to Stellar Winds (Cambridge University Press)
- Lamers, H. J. G. L. M. & Morton, D. C. 1976, *ApJS*, 32, 715
- Lee, E. K. H., Wood, K., Dobbs-Dixon, I., Rice, A., & Helling, C. 2017, *A&A*, 601, A22
- Leenaarts, J. & Carlsson, M. 2009, in Astronomical Society of the Pacific Conference Series, Vol. 415, The Second Hinode Science Meeting: Beyond Discovery-Toward Understanding, ed. B. Lites, M. Cheung, T. Magara, J. Mariska, & K. Reeves, 87
- Leitherer, C., Chapman, J. M., & Koribalski, B. 1995, *ApJ*, 450, 289
- Leitherer, C., Chapman, J. M., & Koribalski, B. 1997, *ApJ*, 481, 898
- Lobel, A. & Blomme, R. 2008, *ApJ*, 678, 408
- Long, K. S. & Knigge, C. 2002, *ApJ*, 579, 725
- Long, K. S. & Knigge, C. 2002, in Astronomical Society of the Pacific Conference Series, Vol. 261, The Physics of Cataclysmic Variables and Related Objects, ed. B. T. Gänsicke, K. Beuermann, & K. Reinsch, 327
- Lucy, L. B. 2002, *A&A*, 384, 725
- Lucy, L. B. 2003, *A&A*, 403, 261
- Lucy, L. B. 2005, *A&A*, 429, 19
- Ludwig, H. G. & Steffen, M. 2016, *Astronomische Nachrichten*, 337, 844
- Metropolis, N. 1985, in Lecture Notes in Physics, Berlin Springer Verlag, ed. R. Alcouffe, R. Dautray, A. Forster, G. Ledonois, & B. Mercier, Vol. 240, 62

- Moens, N., Poniatoski, L. G., Hennicker, L., et al. 2022a, *A&A*, 665, A42
- Moens, N., Sundqvist, J. O., El Mellah, I., et al. 2022b, *A&A*, 657, A81
- Noebauer, U. M. & Sim, S. A. 2015, *MNRAS*, 453, 3120
- Noebauer, U. M. & Sim, S. A. 2019, *Living Reviews in Computational Astrophysics*, 5, 1
- Nordlund, Å. & Stein, R. F. 2009, in *AIP Conf. Ser.*, Vol. 1171, *Recent Directions in Astrophysical Quantitative Spectroscopy and Radiation Hydrodynamics*, ed. I. Hubeny, J. M. Stone, K. MacGregor, & K. Werner, 242–259
- Oskinova, L. M., Feldmeier, A., & Hamann, W. R. 2004, *A&A*, 422, 675
- Oskinova, L. M., Feldmeier, A., & Hamann, W. R. 2006, *MNRAS*, 372, 313
- Owocki, S. P., Castor, J. I., & Rybicki, G. B. 1988, *ApJ*, 335, 914
- Owocki, S. P. & Cramner, S. R. 1994, *ApJ*, 424, 887
- Owocki, S. P. & Puls, J. 2002, *ApJ*, 568, 965
- Owocki, S. P. & Rybicki, G. B. 1984, *ApJ*, 1984, 337
- Papkalla, R. 1995, *A&A*, 295, 551
- Pauldrach, A. W. A., Hoffmann, T. L., & Lennon, M. 2001, *A&A*, 375, 161
- Pauldrach, A. W. A., Hoffmann, T. L., & Méndez, R. H. 2004, *A&A*, 419, 1111
- Pauldrach, A. W. A., Lennon, M., Hoffmann, T. L., et al. 1998, in *Astronomical Society of the Pacific Conference Series*, Vol. 131, *Properties of Hot Luminous Stars*, ed. I. Howarth, 258
- Pejcha, O., Calderón, D., & Kurfürst, P. 2022, *MNRAS*, 510, 3276
- Petrenz, P. & Puls, J. 2000, *A&A*, 358, 956
- Poniatoski, L. G., Kee, N. D., Sundqvist, J. O., et al. 2022, *A&A*, 667, A113
- Poniatoski, L. G., Sundqvist, J. O., Kee, N. D., et al. 2021, *A&A*, 647, A151
- Porter, J. M. 1997, *A&A*, 324, 597
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, *Numerical recipes in FORTRAN. The art of scientific computing* (Press Syndicate of the University of Cambridge)
- Puls, J., Najarro, F., Sundqvist, J. O., & Sen, K. 2020, *A&A*, 642, A172
- Puls, J., Owocki, S. P., & Fullerton, A. W. 1993, *A&A*, 279, 457
- Puls, J., Urbaneja, M. A., Venero, R., et al. 2005, *A&A*, 435, 669
- Röpke, F. K. 2005, *A&A*, 432, 969
- Runacres, M. C. & Owocki, S. P. 2002, *A&A*, 381, 1015

- Sander, A., Hamann, W.-R., Hainich, R., Shenar, T., & Todt, H. 2015, in *Wolf-Rayet Stars: Proceedings of an International Workshop held in Potsdam, Germany, 1-5 June 2015*. (Potsdam, Germany: Universitätsverlag Potsdam), 139–142
- Santolaya-Rey, A. E., Puls, J., & Herrero, A. 1997, *A&A*, 323, 488
- Snow, T. P., J. & Morton, D. C. 1976, *ApJS*, 32, 429
- Snow, T. P., Lamers, H. J. G. L. M., Lindholm, D. M., & Odell, A. P. 1994, *ApJS*, 95, 163
- Steinacker, J., Baes, M., & Gordon, K. D. 2013, *ARA&A*, 51, 63
- Sundqvist, J. O., Owocki, S. P., & Puls, J. 2018, *A&A*, 611, A17
- Sundqvist, J. O., Puls, J., & Feldmeier, A. 2010, *A&A*, 510, A11
- Sundqvist, J. O., Puls, J., Feldmeier, A., & Owocki, S. P. 2011a, *A&A*, 528, A64
- Sundqvist, J. O., Puls, J., Feldmeier, A., & Owocki, S. P. 2011b, *A&A*, 528, A64
- Townsend, R. H. D., Owocki, S. P., & Howarth, I. D. 2004, *MNRAS*, 350, 189
- ud Doula, A. 2014, *Proceedings of the International Astronomical Union*, 9, 321
- van Regemorter, H. 1962, *ApJ*, 136, 906
- Vogl, C., Sim, S. A., Noebauer, U. M., Kerzendorf, W. E., & Hillebrandt, W. 2019, *A&A*, 621, A29
- Štěpán, J. & Trujillo Bueno, J. 2013, *A&A*, 557, A143
- Šurlan, B., Hamann, W. R., Aret, A., et al. 2013, *A&A*, 559, A130
- Šurlan, B., Hamann, W. R., Kubát, J., Oskinova, L. M., & Feldmeier, A. 2012, *A&A*, 541, A37
- Waters, L. B. F. M., Cote, J., & Lamers, H. J. G. L. M. 1987, *A&A*, 185, 206
- Whitney, B. A. 2011, *Bulletin of the Astronomical Society of India*, 39, 101
- Wood, K. & Reynolds, R. J. 1999, *ApJ*, 525, 799
- Xia, C., Teunissen, J., El Mellah, I., Chané, E., & Keppens, R. 2018, *ApJS*, 234, 30
- Yusef-Zadeh, F., Morris, M., & White, R. L. 1984, *ApJ*, 278, 186
- Zsargó, J., Hillier, D. J., & Georgiev, L. N. 2006, *A&A*, 447, 1093

