HPC-techniques for modeling the transfer of polarized radiation with PRD effects

Doctoral Dissertation submitted to the
Faculty of Informatics of the Università della Svizzera Italiana
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

presented by
Simone Riva

under the supervision of
Prof. Dr. Rolf Krause and Dr. Luca Belluzzi

August 2023

Dissertation Committee

Prof. Dr. Rolf Krause	Euler Institute, Università della Svizzera italiana
Dr. Luca Belluzzi	Istituto ricerche solari Aldo e Cele Daccò (IRSOL)
Prof. Dr. Michael Multerer	Euler Institute, Università della Svizzera italiana
Prof. Dr. Svetlana Berdyugina	Istituto ricerche solari Aldo e Cele Daccò (IRSOL)
Dr. Jiří Štěpán	Astronomical Institute of the Czech Academy of Sciences

Dissertation accepted on Now August 2023

Research Advisor

Prof. Dr. Rolf Krause

Co-Advisor

Dr. Luca Belluzzi

PhD Program Director

The PhD program Director ***Prof. Dr. Walter Binder, Prof. Dr. Stefan Wolf***

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Simone Riva
Lugano, Now August 2023

Abstract

One of the key problems in astrophysical research is to investigate the physical properties of astronomical objects by deciphering the information encoded in the radiation we receive from them. This requires modeling the spectral and polarization properties of the radiation they emit by numerically solving the so-called radiative transfer (RT) problem in given models of the object under consideration. In this thesis, we consider the problem of modeling the intensity and polarization profiles of strong lines of the solar spectrum, which encode information about two particular layers of the solar atmosphere: the chromosphere and the chromosphere-corona transition region. This requires solving the RT problem for polarized radiation under non-local thermodynamic equilibrium conditions and taking into account partial frequency redistribution (PRD) effects in the scattering processes. This is a notoriously challenging problem from a computational point of view and is generally solved by introducing simplifying approximations, both on the considered atmospheric models (e.g., isothermal, optically thin, cylindrically symmetric) and in the description of scattering processes (e.g., angle-average assumption).

The thesis is carried out within the framework of a larger project that aims at developing a code for solving the RT problem for polarized radiation, taking PRD effects into account, in state-of-the-art 3D models of the solar atmosphere. A key step to achieve this goal is to develop an efficient and reliable numerical method for the calculation of the emission vector of the RT equation, taking into account PRD effects in their most general formulation. This was the first main goal of the work. Subsequently, we applied the developed methods to quantitatively assess the suitability of an approximation that is commonly applied in the numerical solution of the RT problem with PRD effects.

The core of the problem consists in integrating complex functions which highly depend on several parameters. We studied the basic functional components and analyzed a series of quadrature methods. Once the most convenient approaches have been identified we implemented them in highly optimized algorithms that take into account both the property of the integrand and the architecture of computer systems.

The developed algorithms have been extensively tested and successfully implemented in both 1D and 3D RT codes. Using the 1D code, we modeled, for the first time, the intensity and polarization of various spectral lines in semi-empirical models of the solar atmosphere, accounting for angle-dependent PRD effects. These calculations, which also included magnetic fields of arbitrary strength and direction, as well as bulk velocities, unveiled a series of artifacts induced by the angle-averaged approximation, thus highlighting the importance of considering PRD effects in the most general formulation. One of the most remarkable results was found when modeling the polarizations signals produced by the forward scattering Hanle effect. The algorithm proved to be fully suitable also for the 3D code, which is presently run on an HPC system (CSCS).

Acknowledgements

I gratefully acknowledge all the people with whom I collaborated during my PhD:

- My supervisors Prof. Rolf Krause, and Dr. Luca Belluzzi for giving me the opportunity to do this PhD and for all the advice they gave me during the project.

- Dr. Michele Bianda and all the IRSOL staff for the stimulating and positive research environment at IRSOL.

- Dr. Pietro Benedusi, for developing the RT code to solve the 3D problem, which was a key point for the success of this thesis.

- Dr. Diego Rossinelli, for giving me some necessary skills and help in developing high performance applications.

- Dr. Gioele Janett and Dr. Nuno Guerreiro, for their strong cooperation in writing the papers and for their close scientific collaboration.

- M.Sc. Giulio Mazzaglia, for the development of the AA code used in Chapter 7.

- Dr. Franziska Zeuner, for carrying out some of the observations reported in chapter 7.

- Dr. Ernest Alsina Ballester, for helping me at the beginning of this project to understand the main theory and the problems and challenges to be faced.

- Dr. Fabio Riva, for a final collaboration in testing the results and preparing the defense of the thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we present the scientific context of this PhD dissertation and introduce the specific problem we aim to solve, highlighting its relevance and complexity from a computational point of view. After a brief review of the state of the art, we describe the first main goals of the thesis.

1.1 Scientific context

Most of the available information about astrophysical objects, including the Sun, is encoded in the electromagnetic radiation that we receive from them. This information can be deciphered by comparing the spectral and polarization properties of the observed radiation with the results of theoretical calculations, based on the numerical solution of the so-called radiative transfer (RT) problem. The RT problem consists of modeling how the properties of a beam of electromagnetic radiation are modified as it propagates in a medium (for example, a stellar atmosphere) and interacts with the atoms, molecules, and other particles present therein, through *absorption*, *emission*, and *scattering processes* (e.g., Mihalas, 1978).

In the solar physics community, there is presently great interest in investigating two thin layers of the solar atmosphere: the chromosphere and the transition region between the chromosphere and the corona. These regions are at the core of some enduring problems, such as understanding the physical mechanisms that drive the heating of the solar corona, the acceleration of the solar wind, and the triggering of eruptive phenomena (e.g., flares and coronal mass ejections). It is clear that the magnetic field plays a key role in all of these phenomena, but our knowledge of its intensity and topology is still very limited. The magnetic fields

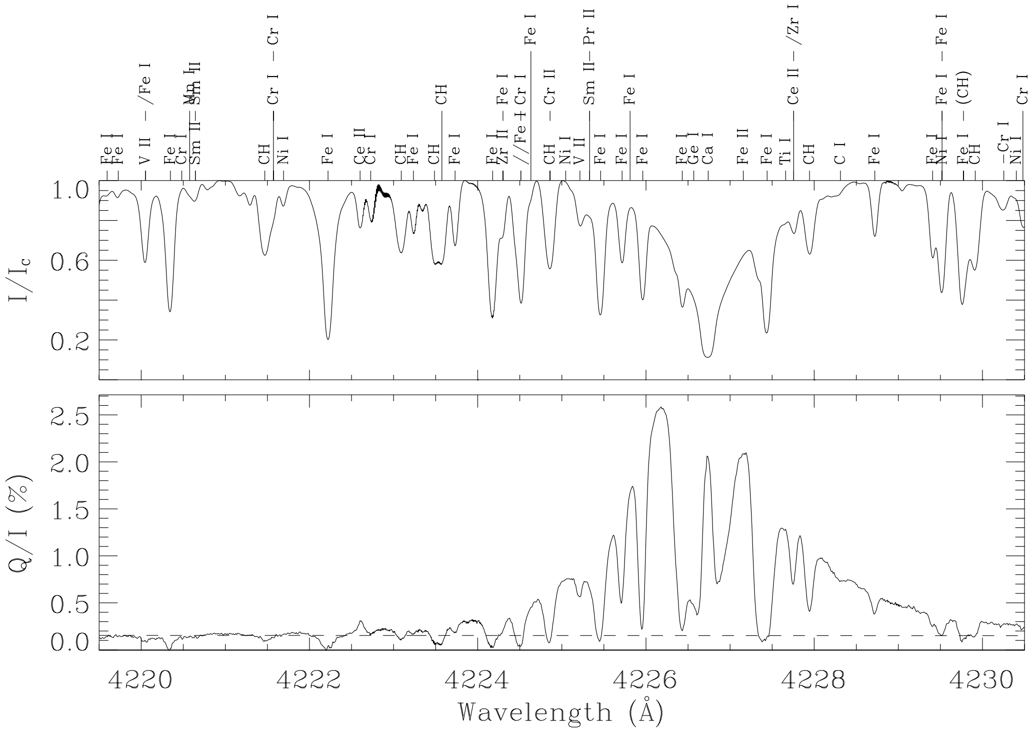

Figure 1.1. *Intensity spectrum (upper panel) and Second Solar Spectrum (lower panel) across an interval including the Ca* I *4227 Å line, measured at the north pole of the Sun, close to the limb, in a condition of quiet Sun with the ZIMPOL polarimeter (Ramelli et al., 2010). The Second Solar Spectrum is represented in terms of the fractional linear polarization* Q/I*, taking the reference direction for positive* Q *parallel to the limb. Credits: Gandorfer (2002).*

present in the solar atmosphere can be investigated by exploiting the fingerprints they leave in the polarization of spectral line radiation through various physical mechanisms, such as the Zeeman effect, the Hanle effect, and magneto-optical effects (e.g., Stenflo, 1994; Landi Degl'Innocenti and Landolfi, 2004). Particularly promising for diagnosing the weak magnetic fields present in the chromosphere and transition region is the Hanle effect (Hanle, 1924). This is the modification, induced by the magnetic field, of the linear polarization produced by the scattering of anisotropic radiation (scattering polarization) (e.g., Stenflo, 1994; Trujillo Bueno, 2001; Landi Degl'Innocenti and Landolfi, 2004).

The clearest manifestation of scattering polarization is the so-called Second

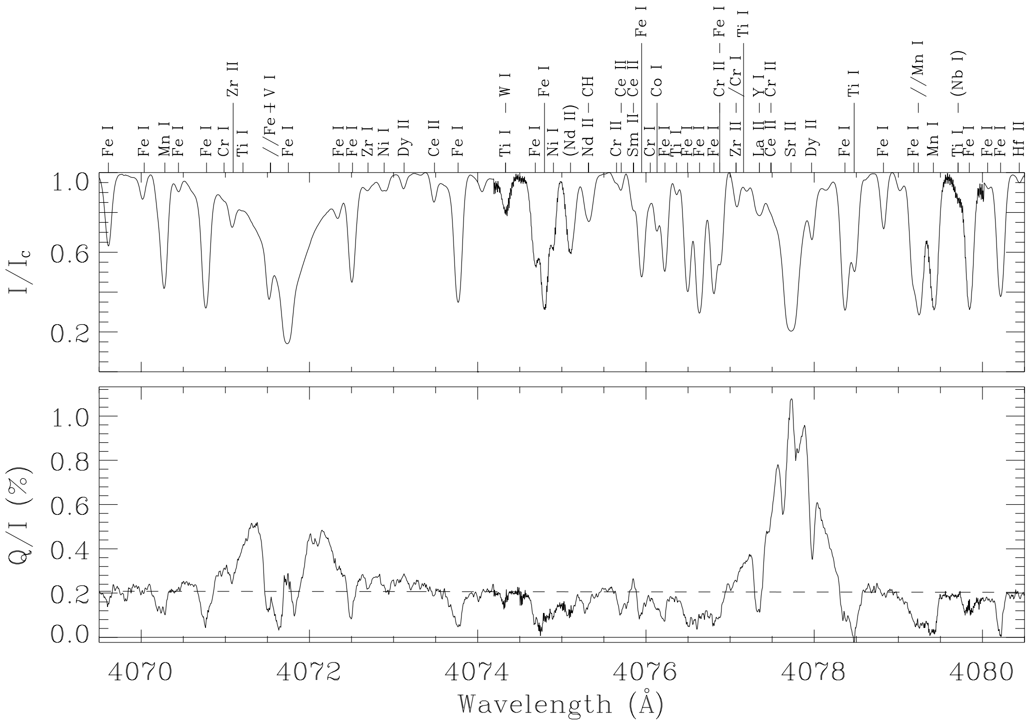

Figure 1.2. *Same as 1.1 but in a spectral interval including the Sr* II *4078 Å line. Credits: Gandorfer (2002).*

Solar Spectrum (e.g. Gandorfer, 2000, 2002, 2005), namely the linearly polarized spectrum of the solar radiation coming from quiet regions close to the edge of the solar disk (limb). Examples of the intensity spectrum and of the Second Solar Spectrum are given in Figures 1.1 to 1.3.

In the visible range, the intensity spectrum shows a plethora of absorption profiles (i.e., regions of reduced intensity) induced by the atoms and molecules composing the solar atmosphere. These profiles can have different shapes, some are broad and deep (strong lines), and others narrower and with smaller depressions (weak lines). In mass, the solar atmosphere is about 73% hydrogen, 25% helium, and 2% other elements. The most frequent absorption profiles are those induced by iron (Fe), which occur across the entire spectrum (e.g. Moore et al., 1966).

Most of the spectral lines appearing in the intensity spectrum do not produce significant linear polarization signals in the Second Solar Spectrum. However, a

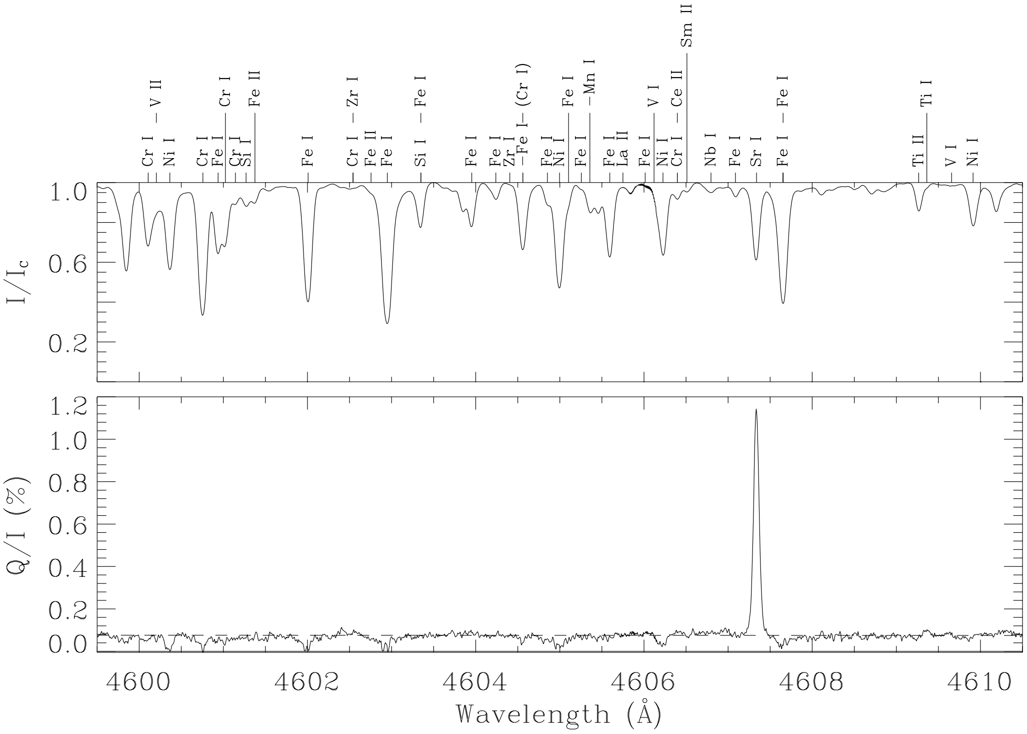

Figure 1.3. *Same as 1.1 but in a spectral interval including the Sr* I *4607 Å line. Credits: Gandorfer (2002).*

number of them show appreciable scattering polarization signals that today can be detected with instruments like ZIMPOL (Ramelli et al., 2010). The largest signals of the Second Solar Spectrum usually appear in resonance lines (Belluzzi and Landi Degl'Innocenti, 2009), that is spectral lines produced by atomic transitions where the lower level corresponds to the *ground* state (i.e., the state with lowest energy) or to *metastable* states (i.e., states that are not connected to lower energy ones via permitted radiative transitions). Examples of three resonance lines showing strong scattering polarization signals are the Ca I 4227 Å, Sr II 4078 Å, and Sr I 4607 Å lines. Measured intensity and Q/I profiles of these lines are shown in Figures 1.1 to 1.3.

The information about the chromosphere and transition region is encoded in strong resonance lines, like H I Ly-α at 1215 Å, Mg II *h* and *k* at 2800 Å, Ca II H and K at 3950 Å, or the Ca I line at 4227 Å. Modeling the intensity and polarization profiles of such lines is difficult because it requires solving the RT problem for polarized radiation under non-local thermodynamic equilibrium (non-LTE) conditions and taking into account partial frequency redistribution (PRD) effects (i.e., frequency correlations between the incoming and the outgoing photons in the scattering process). PRD effects are essential for modeling the large scattering polarization profiles that are observed in some of the lines mentioned above, and, in particular, the broad lobes that are found in their wings (e.g., Holzreuter et al., 2005; Belluzzi and Trujillo Bueno, 2012; Alsina Ballester et al., 2017). Such wing signals would be completely lost if one assumes the limit of complete frequency redistribution (CRD), in which no frequency correlation between the incoming and the outgoing photons in the scattering process is accounted for.

The main difficulty in developing diagnostic methods based on the Hanle effect is that the amplitude of scattering polarization strongly depends on the anisotropy of the radiation field illuminating the atoms, which, however, is not known a priori. There are two main strategies to overcome this difficulty. The first one is to consider different spectral lines that encode information on the same atmospheric region but have different sensitivities to the Hanle effect (*differential Hanle effect*, see Stenflo et al., 1998). This technique is successfully applied at the photospheric level, exploiting the scattering polarization signals of molecular lines (e.g., Berdyugina et al., 2002; Berdyugina and Fluri, 2004; Trujillo Bueno et al., 2004; Kleint et al., 2010) The alternative strategy is to model the observed scattering polarization signals in realistic 3D models of the solar atmosphere, in which the anisotropy of the radiation field can be reliably estimated. Thanks to the availability of magneto-hydrodynamic (MHD) simulations of the solar atmosphere that reach the chromosphere and corona (e.g., Carlsson et al., 2016), and thanks to the computational power of modern supercomputers, this strategy is today feasible (e.g., Trujillo Bueno et al. (2018)).

This thesis work aims at developing a series of numerical tools needed to solve the RT problem for polarized radiation in state-of-the-art 3D models of the solar atmosphere, taking scattering polarization and PRD effects into account.

1.2 State of the art

Numerous applications (RT codes), some of which are publicly available, have been developed to solve the RT problem, under different assumptions and considering different geometries. A very popular one is the RH code (Uitenbroek, 2001). This application allows solving the RT problem for multilevel atomic systems under non-LTE conditions, in 1D, 2D, and 3D models of the solar atmosphere, both in the limit of CRD and taking PRD effects into account. RH takes into account the polarization produced by the Zeeman effect, but it neglects scattering polarization. To date, the only code that allows solving the non-LTE RT problem, taking scattering polarization into account, in 3D models of the solar atmosphere, is PORTA (Štěpán and Trujillo Bueno, 2013). PORTA allows considering multilevel atomic systems, but only within the limit of CRD. On the High-Performance Computing (HPC) side, PORTA implements an innovative algorithm for domain decomposition, the "snake algorithm", which allows massive parallel computation.

The numerical solution of the non-LTE RT problem taking scattering polarization and PRD effects into account is known to be particularly challenging from the computational point of view, also in the 1D case. For this reason, in most of the available applications, the PRD effects are included by assuming the so-called *angle-averaged* (AA) simplifying approximation (e.g., Mihalas, 1978; Rees and Saliba, 1982; Bommier, 1997b; Anusha and Nagendra, 2011; Alsina Ballester et al., 2017). By smearing out geometrical dependencies in the description of the scattering processes, this approximation can, however, introduce significant, and difficult-to-predict, inaccuracies, especially in the modeling of scattering polarization (e.g., Faurobert, 1987a, 1988a; Sampoorna et al., 2011a, 2017; Janett et al., 2021a), clearly compromising the effort of solving the RT problem in realistic 3D models of the solar atmosphere.

During the last decades, significant efforts have been devoted to the development of numerical methods tailored for modeling scattering polarization taking PRD effects into account, in their most general, angle-dependent (AD) formulation (see, for instance, the review by Nagendra, 2019). Worth mention is the approach based on the decomposition of the four Stokes parameters (i.e., the main physical quantities used to describe polarized radiation, see Sect. 1.4.1) into nine irreducible spherical components characterized by cylindrical symmetry, and on the Fourier expansion of the PRD redistribution matrix (i.e., the tool used to describe PRD phenomena, see Sect. 1.4.2) with respect to the azimuth (see Frisch, 2007, 2009, 2010). Exploiting this method, Anusha and Nagendra (2011) pioneered the problem of modeling scattering polarization with AD-PRD effects in multi-D media. By performing applications in academic models, these authors showed that this method allows obtaining accurate results retaining a number of terms of the Fourier expansion appreciably lower than the number of azimuths that one would need to consider in a standard approach. That work was carried out considering an approximated expression of the redistribution matrix, namely approximation II of Bommier (1997b), which is valid under the so-called *weak-field approximation* and is based on a decomposition of the frequency domain leading to a decoupling of frequency redistribution and polarization (e.g., Frisch, 2009). We recall that the weak-field approximation is valid when the magnetic splitting due to the Zeeman effect is much smaller than the Doppler width. In this case, the magnetic splittings can be neglected in the line profiles, with a considerable simplification of the problem, but the polarization signals produced by the Zeeman effect are necessarily neglected. To the authors' knowledge, the work of Anusha and Nagendra (2011) has not been generalized yet to handle the most general expression of the redistribution matrix, and it has not been extended to more realistic atmospheric models.

In this thesis, we aim to work with the most general expression of the redistribution matrix, which allows us to include magnetic fields of arbitrary strength and to account for the polarization signals produced by the Zeeman effect, and not to follow the aforementioned approach. However, the methods that will be developed in this thesis might also be useful for the implementation of the strategy discussed above.

Numerical approaches that could be suited for an efficient approximation of the redistribution matrix are those based on low-rank methods (e.g. Markovsky, 2012). In a recent work, Paganini et al. (2021) proposed a fast approximation of the AA redistribution matrix by means of Chebyshev polynomials using the `chebfun3` software (Hashemi and Trefethen, 2017). Unfortunately, the generalization of this approach to approximate the redistribution matrix in its general, angle-dependent form presents remarkable difficulties and, to the knowledge of the author, has not been attempted so far.

Before the beginning of this project, the non-LTE RT problem for polarized radiation, taking AD PRD effects into account, had always been solved in simplified atmospheric models (e.g., isothermal 1D slabs). The only exception is the work of del Pino Alemán et al. (2016), who modeled the scattering polarization signals of the Mg II h and k lines by solving the aforementioned problem in a 1D semi-empirical atmospheric model, but still assuming cylindrical symmetry in order to lower the computational cost. As it will be discussed in the next sections, the first main objective of this thesis work is to develop an efficient method for calculating the so-called emissivity (in the polarized case) taking AD PRD effects into account. To the best of our knowledge, no details of the quadrature methods applied in previous works to calculate this quantity have been explicitly reported in the literature.

1.3 Scope of the work

This thesis is part of a larger project that aims at developing a code to solve the non-LTE RT problem, taking into account scattering polarization and PRD effects, in a 3D geometry. This new RT code will be a very useful tool for studying the magnetism of the outer solar atmosphere, as it would allow us to model the magnetic sensitivity of strong resonance lines through the combined action of Hanle, Zeeman, and magneto-optical effects in realistic models of the solar atmosphere. A key step to achieve this goal is to have available a fast and accurate algorithm for the computation of the emission coefficient of the RT equation, taking PRD effects into account, without considering any simplifying approximation (see Sect. 1.4). Devising, developing, and optimizing such an algorithm by implementing HPC methods for performing the various steps of this calculation is the first main goal of this Ph.D. thesis work.

In parallel, we will cooperate with the other members of the group to the development of a new solution strategy for the non-LTE RT problem with PRD in both 1D and 3D atmospheric models, considering spatially dependent physical parameters. This solution strategy is briefly described in Sect. 1.8. For more details, we refer the reader to Janett et al. (2021b); Benedusi et al. (2021, 2022); Benedusi et al. (2023).

1.4 Mathematical formulation

1.4.1 The radiative transfer equation for polarized radiation

The polarization of a beam is completely described by the four Stokes parameters: I, Q, U, and V, where $I \in \mathbb{R}_+$ is the intensity, Q, $U \in \mathbb{R}$ quantify the linear polarization, and $V \in \mathbb{R}$ the circular polarization (e.g. Landi Degl'Innocenti and Landolfi, 2004). To simplify the notation, it is common to introduce a vector $\mathbf{I}$ whose components are the four Stokes parameters $\mathbf{I} = [I, Q, U, V]^{\mathsf{T}}$ (Stokes vector). The Stokes parameters and therefore the Stokes vector are in general a function of the spatial point $\mathbf{r} \in \mathbb{R}^3$, the frequency $\nu \in \mathbb{R}_+$ and the direction $\mathbf{\Omega}$ of the considered beam, i.e. $\mathbf{I}(\mathbf{r}, \mathbf{\Omega}, \nu)$.

We consider a right-handed Cartesian reference system with z-axis directed along the vertical (see Fig. 1.4). The direction $\mathbf{\Omega}$ is specified by a unit vector expressed in spherical coordinates (i.e., $\mathbf{\Omega} = [\theta, \chi]^{\mathsf{T}}$, where $\theta \in [0, \pi]$ is the inclination with respect to the local vertical and $\chi \in [0, 2\pi)$ is the azimuth). Unless otherwise specified, we will consider radiation emerging from the atmo-

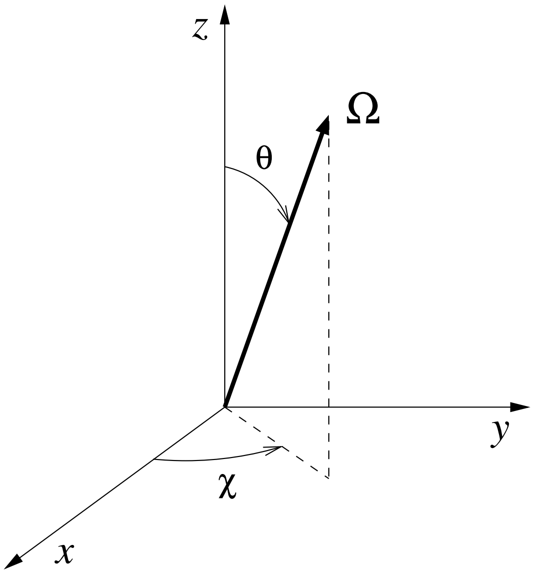

Figure 1.4. *Right-handed Cartesian reference system considered in the problem. The z-axis is directed along the local vertical. Any vector is specified through its polar angles θ (inclination) and χ (azimuth).*

sphere along a line of sight (LOS) lying in the $x-z$ plane, and we will take the reference direction for positive Stokes Q parallel to the y-axis.

When a beam of radiation propagates through a medium, it interacts with the molecules and atoms through various processes (e.g. absorption, scattering, etc.), and its properties are modified. These modifications are fully described by the RT equation:

$$\nabla_{\Omega}\mathbf{I}(\mathbf{r},\boldsymbol{\Omega},\nu)=-K(\mathbf{r},\boldsymbol{\Omega},\nu)\cdot\mathbf{I}(\mathbf{r},\boldsymbol{\Omega},\nu)+\varepsilon(\mathbf{r},\boldsymbol{\Omega},\nu), \tag{1.1}$$

where ∇_{Ω} is the directional derivative along the considered direction $\boldsymbol{\Omega}$. The quantity $\varepsilon\in\mathbb{R}^4$ is the *emission vector*, which describes the contribution to the Stokes parameters of the considered beam due to the radiation emitted by the plasma at point $\mathbf{r}$. The *propagation matrix* $K\in\mathbb{R}^{4\times4}$

$$K=\begin{pmatrix}\eta_I & \eta_Q & \eta_U & \eta_V\\ \eta_Q & \eta_I & \rho_V & -\rho_U\\ \eta_U & -\rho_V & \eta_I & \rho_Q\\ \eta_V & \rho_U & -\rho_Q & \eta_I\end{pmatrix} \tag{1.2}$$

depends on the spatial point $\mathbf{r}$, the direction $\boldsymbol{\Omega}$, and the frequency ν. Its diagonal elements (generally called opacity and indicated with η_I) describe the absorption of radiation, the off-diagonal ones describe the phenomena of dichroism (η_Q, η_U, and η_V), and anomalous dispersion (ρ_Q, ρ_U, and ρ_V) (e.g., Landi Degl'Innocenti and Landolfi, 2004; Landi Degl'Innocenti, 2007).

We observe that the emission vector and the propagation matrix depend on the excitation state of the atoms. Under non-LTE conditions, this has to be calculated by solving the statistical equilibrium equations, which describe the interaction of the atoms with the radiation field (radiative processes), other particles present in the plasma (collisional processes), and the possible presence of external magnetic and/or electric fields. In the general non-LTE case, the radiative transfer equation must be satisfied in all directions and frequencies, and the whole *radiative transfer problem* consists in finding a self-consistent solution for the radiation field and the atomic system at all spatial points.

1.4.2 The emission vector

The emission vector ε appearing in Equation (1.1), whose components are the *emission coefficients* in the four Stokes parameters, is given by the sum of various terms that describe the contributions from different physical processes. Since the calculation of the emissivity is local, for notational simplicity hereafter we do not explicitly indicate the dependence on the spatial coordinate r, i.e.,

$$\begin{aligned}
&\varepsilon : \mathbb{R} \times \mathbb{R}^2 \longrightarrow \mathbb{R}^4 \\
&\varepsilon(\nu, \mathbf{\Omega}) = \varepsilon^{\ell,\mathrm{sc}}(\nu, \mathbf{\Omega}) + \varepsilon^{\ell,\mathrm{th}}(\nu, \mathbf{\Omega}) + \\
&\qquad\qquad\quad \varepsilon^{c,sc}(\nu, \mathbf{\Omega}) + \varepsilon^{c,th}(\nu, \mathbf{\Omega}) .
\end{aligned} \tag{1.3}$$

The first two terms represent the scattering (label sc) and thermal (label th) contributions from *line* (label ℓ) processes (i.e., transitions between bound states of the considered atomic system). The last two terms represent the analogous contributions from *continuum* (label c) processes (e.g., scattering on electrons and on atoms and ions, radiative recombination, etc.).

Hereafter, we will focus the attention on the term that describes the contribution from scattering processes by the considered atomic system (line scattering contribution), which is the most demanding one to compute. Describing the PRD effects by means of the redistribution matrix formalism (e.g., Bommier, 1997a,b), this term is given by:

$$\boxed{\begin{aligned}
&\varepsilon^{\ell,\mathrm{sc}} : \mathbb{R} \times \mathbb{R}^2 \longrightarrow \mathbb{R}^4 \\
&\varepsilon^{\ell,\mathrm{sc}}(\nu, \mathbf{\Omega}) = k_L \int_{\mathbb{R}_+} \mathrm{d}\nu' \oint \frac{\mathrm{d}\mathbf{\Omega}'}{4\pi} R\left(\nu', \mathbf{\Omega}', \nu, \mathbf{\Omega}\right) \mathbf{I}\left(\nu', \mathbf{\Omega}'\right),
\end{aligned}} \tag{1.4}$$

where $\mathbf{I}$ is the Stokes vector of the radiation field that illuminates the atom, k_L is the *frequency integrated absorption coefficient*, and R is the *redistribution matrix*, which describes the joint probability that in a scattering process, an atom absorbs radiation (incoming radiation) with direction $\mathbf{\Omega}'$, frequency ν', and a given polarization state, and re-emits radiation (outgoing radiation) with direction $\mathbf{\Omega}$, frequency ν, and a new polarization state.

In this work, we consider an atomic system composed of two levels (two-level atom), and we assume that the lower level is unpolarized and infinitely sharp (this is generally a good assumption when the lower level is the ground state). More details on this atomic model are provided in Sect. A.1. Under these assumptions, it can be shown that the redistribution matrix is given by the linear combination of two terms (Bommier, 1997a,b) that, following the notation of Hummer (1962), are generally indicated with R^{II} and R^{III}:

$$\begin{aligned} &R : \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}^2 \longrightarrow \mathbb{R}^{4\times4} \\ &R\left(\nu', \mathbf{\Omega}', \nu, \mathbf{\Omega}\right) = R^{\text{II}}\left(\nu', \mathbf{\Omega}', \nu, \mathbf{\Omega}\right) + R^{\text{III}}\left(\nu', \mathbf{\Omega}', \nu, \mathbf{\Omega}\right) . \end{aligned} \tag{1.5}$$

The R^{II} matrix describes scattering processes that are coherent in frequency (complete correlation between ν' and ν) in the reference system of the atom (i.e., a reference system in which the atom is at rest). The incoming and outgoing frequencies are fully correlated *if* the atom, while excited during the scattering process, is not perturbed by elastic collisions with neutral particles (i.e. atoms and molecules constituting the solar atmosphere).

The R^{III} matrix describes instead scattering processes that are completely incoherent in frequency (no correlation between ν' and ν) in the same reference system. This happens if the atom, while excited, is affected by elastic collisions with *neutral particles* that completely relax the coherence between the incoming and outgoing frequencies.

The relative weight between R^{II} and R^{III} (given by a suitable branching ratio) depends on the rate of elastic collisions. In the limit of no elastic collision, R^{III} vanishes, while when elastic collisions are very efficient, R^{III} dominates over R^{II}. A convenient way to approximately quantify this relative weight is through the coherence fraction:

$$\alpha = \frac{\Gamma_R + \Gamma_I}{\Gamma_R + \Gamma_I + \Gamma_E} , \tag{1.6}$$

where the quantities Γ_R, Γ_I, and Γ_E are the broadening constants due to spontaneous emission, inelastic de-exciting collisions, and elastic collisions, respectively (for more details, see Alsina Ballester et al., 2017, and Equation (1.14)).

A value of α close to 1 means that the contribution of R^{III} is negligible compared to that of R^{II}. The R^{III} matrix is usually dominant in the lower part of the solar atmosphere (photosphere), where the number density of perturbers is higher. Its contribution is therefore relevant in the formation of photospheric lines (e.g. the Sr I 4607 Å line). On the other hand, the contribution of R^{II} dominates in medium-upper regions of the solar atmosphere (chromosphere)[1], and therefore it characterizes the formation of chromospheric lines (e.g. the Ca I 4227 Å and Sr II 4078 Å lines, see Chapter 7 for detailed results for these lines and a comprehensive discussion).

The problem is formulated in the observer's reference system, in which the atom is moving with a given velocity. In this reference frame, it is necessary to take into account the Doppler effect due to the atomic motions, which induces further frequency redistribution phenomena. The atomic velocity has in general a thermal and a bulk component. Focusing on the thermal component, this is generally well described by an isotropic Maxwellian distribution, depending on the temperature and the atomic mass. Accounting for these thermal velocities leads to a convolution of the expressions of the redistribution matrix in the atomic reference frame with the Maxwellian distribution model, which induces the appearance of the Faddeeva function (see also Eq. (1.16)). A complete analytical derivation of the redistribution matrices in the observer's frame is presented in Sect. A.2.2. The treatment of bulk velocities is instead discussed in Sect. 1.5.

For the analysis and numerical treatment of the redistribution matrices, it is generally convenient to express the equations in terms of the *reduced frequencies* of the outgoing and incoming radiation, u and u', respectively, defined as:

$$u = \frac{\nu_0 - \nu}{\Delta\nu_{\mathrm{D}}}, \qquad u' = \frac{\nu_0 - \nu'}{\Delta\nu_{\mathrm{D}}}, \tag{1.7}$$

where $\nu_0 = (E_u - E_\ell)/h$ (with E_u and E_ℓ the energies of the upper and lower levels, respectively, and h the Planck constant) is the line center frequency, and $\Delta\nu_{\mathrm{D}}$ is the Doppler width (e.g. Mihalas, 1978). This is defined as (see Alsina Ballester et al., 2017):

$$\Delta\nu_{\mathrm{D}} = \frac{\nu_0}{c}\sqrt{\frac{2K_B T}{M_a} + v_t^2}, \tag{1.8}$$

where M_a is the atomic mass, v_t the micro-turbulent velocity[2], K_b the Boltzmann constant, and c the speed of light in vacuum. The use of reduced frequencies,

[1] NASA: Layers of the Sun

[2] The microturbulent velocity v_t is a free parameter introduced in 1D atmospheric models to compensate the absence of convective phenomena in these models (e.g. Mucciarelli, 2011).

where the origin corresponds to the central line frequency ν_0, makes some relevant properties of the redistribution matrices invariant with respect to the thermal parameters (depending on the spatial point) of the atmospheric model.

Remark. *To move the resulting emission vectors from the reduced frequency space to the frequency space, it is necessary to multiply this result by* $\frac{1}{\Delta\nu_D}$ *(see also Eq.* (1.4)*), i.e.*

$$\varepsilon^{\ell,\mathrm{sc}}(\nu,\mathbf{\Omega}) = \frac{1}{\Delta\nu_D}\varepsilon^{\ell,\mathrm{sc}}(u,\mathbf{\Omega})\,.$$

The calculation of R^{II} and its contribution to the emission vector by Eq. (1.4) generally takes the vast majority of the *time-to-solution* of the entire RT problem (Benedusi et al., 2023). The first aim of this thesis was to develop and optimise suitable methods to perform such calculations. For the case of R^{III}, it is instead possible to introduce some approximations, such as the assumption of complete frequency redistribution in the observer's frame (e.g., Alsina Ballester et al., 2017), which greatly simplify its evaluation. A quantitative analysis of the suitability of this assumption was the second main goal of this thesis. This investigation is reported in Chapter 6 (see also Riva et al. (2023)).

1.4.3 Redistribution matrix in the formalism of the irreducible spherical tensors

In this work, we use the redistribution matrices derived by Bommier (1997a,b). In the formalism of *irreducible spherical tensors for polarimetry* (see Chap. 5 of Landi Degl'Innocenti and Landolfi, 2004), the R^{II} and R^{III} redistribution matrices in the observer's frame can be written as:

$$\begin{aligned} &R^{\mathrm{X}} : \mathbb{R}\times\mathbb{R}^2\times\mathbb{R}\times\mathbb{R}^2 \longrightarrow \mathbb{R}^{4\times 4} \\ &R^{\mathrm{X}}\left(\mathbf{\Omega}', u', \mathbf{\Omega}, u\right) = \sum_{K,K'=0}^{2}\sum_{Q=-K_{\min}}^{K_{\min}} \mathcal{R}_Q^{\mathrm{X},KK'}\left(u',\mathbf{\Omega}',u,\mathbf{\Omega}\right)\mathcal{P}_Q^{KK'}\left(\mathbf{\Omega}',\mathbf{\Omega}\right), \end{aligned} \tag{1.9}$$

where $x = \mathrm{II}$, III and $K_{\min} = \min(K, K')$. The indices K, K', and Q are related to the rank of the polarization tensors and give rise to a set of 19 different values.

The quantity $\mathcal{P}_Q^{KK'} \in \mathbb{C}^{4\times 4}$ is the *scattering phase matrix* (see Chap. 5 Landi Degl'Innocenti and Landolfi, 2004). Its calculation can be easily managed, as one notices that it does not depend on u' and u, but only on the directions of the incoming $\mathbf{\Omega}'$ and outgoing $\mathbf{\Omega}$ radiation and on the orientation of the magnetic field. In the case of a deterministic magnetic field, its explicit expression is given by (see Alsina Ballester et al., 2017, see also Appendix A.2):

$$\begin{aligned}
&\mathcal{P}^{KK'}_{Q} \,:\, \mathbb{R}^2 \times \mathbb{R}^2 \longrightarrow \mathbb{C}^{4\times 4} \\
&\mathcal{P}^{KK'}_{Q}\left(\boldsymbol{\Omega}', \boldsymbol{\Omega}\right) = \sum_{Q'=-K}^{K} \sum_{Q''=-K'}^{K'} (-1)^{Q'} \left[\mathcal{T}^{K'}_{Q''}(\boldsymbol{\Omega})\right] \left[\mathcal{T}^{K}_{-Q'}(\boldsymbol{\Omega}')\right]^{\mathsf{T}} \overline{\mathcal{D}}^{K'}_{QQ''} \, \mathcal{D}^{K}_{QQ'} \, ,
\end{aligned} \tag{1.10}$$

where the quantities $\mathcal{D}^{K}_{QQ'} \in \mathbb{C}$ are the elements of the rotation matrix (see Sect. 2.6 of Landi Degl'Innocenti and Landolfi, 2004) that depend on the local direction of the magnetic field, and $\mathcal{T}^{K}_{-Q'} \in \mathbb{C}^4$ are *irreducible geometrical tensors for polarimetry* (see Chap. 5, Table 5.6 of Landi Degl'Innocenti and Landolfi, 2004).

The quantities

$$\mathcal{R}^{\text{x},KK'}_{Q} : \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}^2 \longrightarrow \mathbb{C}$$

are commonly referred to as *redistribution functions*. From the computational standpoint, the evaluation of these quantities is one of the most demanding steps of the whole problem.

Assuming that no bulk velocities are present (the inclusion of bulk velocities is discussed in Sect. 1.5), the angular dependence of the $\mathcal{R}^{\text{x},KK'}_{Q}$ functions is fully contained in the *scattering angle* $\Theta \in [0, \pi]$, that is, the angle between the directions of the incoming $\boldsymbol{\Omega}'$ and the outgoing $\boldsymbol{\Omega}$ radiation beams:

$$\Theta = \left|\arccos\left(\boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}\right)\right| . \tag{1.11}$$

1.4.4 The R^{II} redistribution matrix

The expression of the R^{II} redistribution matrix in the observer's frame is given by

$$\begin{aligned}
&R^{\text{II}} : \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}^2 \longrightarrow \mathbb{R}^{4\times 4} \\
&R^{\text{II}}\left(u', \boldsymbol{\Omega}', u, \boldsymbol{\Omega}\right) = \sum_{K=0}^{2} \sum_{K'=0}^{2} \sum_{Q=-K_{min}}^{K_{min}} \mathcal{R}^{\text{II},KK'}_{Q}\left(u', u, \Theta\left(\boldsymbol{\Omega}', \boldsymbol{\Omega}\right)\right) \mathcal{P}^{KK'}_{Q}\left(\boldsymbol{\Omega}', \boldsymbol{\Omega}\right) .
\end{aligned} \tag{1.12}$$

The analytical expression of the $\mathcal{R}_Q^{\text{II},KK'}$ function is:

$$\begin{aligned}
\mathcal{R}_Q^{\text{II},KK'} &: \quad \mathbb{R}\times\mathbb{R}\times(0,\pi)\longrightarrow\mathbb{C} \\
\mathcal{R}_Q^{\text{II},KK'}\left(u',u,\Theta\right) &= \alpha_Q \sum_{\{M_u M_u' M_\ell M_\ell'\}} \mathcal{C}_{KK'QM_uM_u'M_\ell M_\ell'} \\
&\quad \times \frac{1}{\pi\sin(\Theta)}\exp\left[-\left(\frac{u-u'+u_{M_\ell M_\ell'}}{2\sin(\Theta/2)}\right)^2\right] \\
&\quad \times \frac{1}{2}\left[W\left(\frac{a}{\cos(\Theta/2)},\frac{u+u'+u_{M_u' M_\ell}+u_{M_u' M_\ell'}}{2\cos(\Theta/2)}\right)+\right. \\
&\quad \left.\overline{W}\left(\frac{a}{\cos(\Theta/2)},\frac{u+u'+u_{M_u M_\ell}+u_{M_u M_\ell'}}{2\cos(\Theta/2)}\right)\right],
\end{aligned} \tag{1.13}$$

where the notation $\overline{f}\,(\cdot)$ indicates a complex conjugate function. It should be noted that the function $\mathcal{R}_Q^{\text{II},KK'}$ is not defined for $\Theta=0$ and $\Theta=\pi$. In these cases, $\mathcal{R}_Q^{\text{II},KK'}$ assumes different analytical expressions, which are discussed at the end of this section. The quantity α_Q is the branching ratio of R^{II}, given by:

$$\alpha_Q=\frac{\Gamma_R}{\Gamma_R+\Gamma_I+\Gamma_E+2\pi \mathrm{i}\,\nu_L g_u Q}, \tag{1.14}$$

The quantities Γ_R, Γ_I, and Γ_E are the broadening constants due to spontaneous emission, inelastic de-exciting collisions, and elastic collisions, respectively (for more details, see Alsina Ballester et al., 2017). The last (imaginary) term at the denominator of α_Q in Eq. (1.14) is the Hanle term, where ν_L is the Larmor frequency and g_u is the Landé factor of the upper level (for more details, see Alsina Ballester et al., 2017). All these quantities depend only on the spatial point.

The quantities M_u, M_u', M_ℓ, and M_ℓ' are the so-called magnetic quantum numbers, which characterize the magnetic sub-levels of a given atomic energy level. An energy level with angular momentum quantum number J (which can only take integer or half-integer values) is actually composed of $(2J+1)$ magnetic sub-levels characterized by the quantum number $M=-J,-J+1,\ldots,J$. The quantity $\mathcal{C}_{KK'QM_uM_u'M_\ell M_\ell'}$ Bommier (1997a) is a real number that depends on the tensorial indices (K, K' and Q), the angular momentum quantum numbers of the upper and lower level (J_ℓ and J_u, respectively), and the magnetic quantum numbers (M_u, M'_u, M_ℓ, M'_ℓ). Its explicit expression is:

$$
\begin{aligned}
\mathcal{C}_{KK'QM_uM'_uM_\ell M'_\ell} = \sum_{pp'p''p'''=-1}^{1} & 3\,(2J_u+1)\,\sqrt{2K+1}\,\sqrt{2K'+1}\,(-1)^{2J_u-M_\ell-M'_\ell} \\
\times & \begin{pmatrix} J_u & J_\ell & 1 \\ M_u & -M_\ell & -p \end{pmatrix} \begin{pmatrix} J_u & J_\ell & 1 \\ M'_u & -M_\ell & -p' \end{pmatrix} \\
\times & \begin{pmatrix} J_u & J_\ell & 1 \\ M_u & -M'_\ell & -p'' \end{pmatrix} \begin{pmatrix} J_u & J_\ell & 1 \\ M'_u & -M'_\ell & -p''' \end{pmatrix} \\
\times & \begin{pmatrix} 1 & 1 & K \\ -p & p' & Q \end{pmatrix} \begin{pmatrix} 1 & 1 & K' \\ -p'' & p''' & Q \end{pmatrix} ,
\end{aligned}
\tag{1.15}
$$

where the operator:

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}$$

stands for the Wigner 3j symbol (e.g. Landi Degl'Innocenti and Landolfi, 2004, chap. 2). It should be noted that $\mathcal{C}_{KK'QM_uM'_uM_\ell M'_\ell}$ depends only on the atomic system under consideration and not on the spatial point, frequency, and direction of the radiation. Therefore, this quantity can be easily pre-calculated and stored in a table.

Recalling that the Wigner 3j symbols vanish if the sum of the elements of the second row is different from zero, it can be seen that the indices p, p', p'', and p''', which can take values -1, 0, and +1, guarantee that the selection rule for the magnetic quantum numbers $|M_u - M_\ell| \leq 1$ is satisfied. In the numerical algorithm, this selection rule can be taken into account in the loops over the magnetic quantum numbers (see Algorithm 1). If this is done, then the sum over p, p', p'', and p''' becomes redundant, and the values of these indices can be fixed a priori: $p = M_u - M_\ell$, $p' = M'_u - M_\ell$, $p'' = M_u - M'_\ell$, and $p''' = M'_u - M'_\ell$. Using Algorithm 1 it can be seen that, for example, in the case of Mg II k 2795 Å and Sr II 4078 Å lines originating from transition $J_u = 3/2 \longleftrightarrow J_\ell = 1/2$ there are 26 combinations of magnetic quantum numbers, while in the case of Ca I 4227 Å and Sr I 4607 Å lines with $J_u = 1 \longleftrightarrow J_\ell = 0$ the number of combinations is 9.

Algorithm 1: Algorithm for the selection of the sets of magnetic quantum numbers contributing to R^{II}, given J_u and J_ℓ.

```
Function M^II $(J_u, J_\ell)$ :
    for $M_u = -J_u \dots J_u$ do
        for $M_\ell = -J_\ell \dots J_\ell$ do
            if $|M_u - M_\ell| > 1$ then
                Continue
            for $M'_u = -J_u \dots J_u$ do
                if $\left|M'_u - M_\ell\right| > 1$ then
                    Continue
                for $M'_\ell = -J_\ell \dots J_\ell$ do
                    if $\left|M'_\ell - M_u\right| > 1$ then
                        Continue
                    if $\left|M'_\ell - M'_u\right| > 1$ then
                        Continue
                    Yield $M_u, M'_u, M_\ell, M'_\ell$
```

Finally the function:

$$\begin{aligned} &W : \mathbb{C} \longrightarrow \mathbb{C} \\ &W(z) = e^{-z^2} \operatorname{erfc}(-iz) \\ &W(x+iy) = V(y,x) + iL(y,x) \end{aligned} \tag{1.16}$$

is the ***Faddeeva function*** (a.k.a. complex error function, or Kramp's function) (e.g., Faddeeva and Terent'ev, 1961; Poppe and Wijers, 1990; Landi Degl'Innocenti and Landolfi, 2004), where $\operatorname{erfc}(\cdot)$ is the complementary error function. The real part of $W(\cdot)$ is the Voigt profile $V(\cdot,\cdot)$, which is defined as the convolution between a Gaussian $g(\cdot)$ and a Lorentzian distribution $\mathcal{L}_{(y)}(\cdot)$, i.e.

$$\begin{aligned} V(y,x) &= \left[g * \mathcal{L}_{(y)}\right](x) \\ &= \frac{y}{\pi} \int_{\mathbb{R}} e^{-t^2} \frac{1}{(x-t)^2 + y^2} \mathrm{d}t \end{aligned} \tag{1.17}$$

and the imaginary part is the associated dispersion profile $L(\cdot,\cdot)$ which is defined as a convolution between a Gaussian and the profile $\mathcal{F}_{(y)}(\cdot)$, i.e.,

$$\begin{aligned} L(y,x) &= \left[g * \mathcal{F}_{(y)}\right](x) \\ &= \frac{1}{\pi}\int_{\mathbb{R}} e^{-t^2}\frac{x-t}{(x-t)^2+y^2}\mathrm{d}t, \end{aligned} \tag{1.18}$$

where:

$$\begin{aligned} g(x) &= e^{-x^2} \\ \mathcal{L}_{(y)}(x) &= \frac{y}{\pi(x^2+y^2)} \\ \mathcal{F}_{(y)}(x) &= \frac{x}{\pi(x^2+y^2)}. \end{aligned} \tag{1.19}$$

Hereafter, the Faddeeva function will be represented with the following notation:

$$W(y,\, x) = W(y + ix). \tag{1.20}$$

The quantity a appearing in the first argument of the Faddeeva function in Eq. (1.13) is the damping parameter (e.g., Mihalas, 1978), which only depends on the spatial point. The terms $u_{M_\ell M'_\ell}$, $u_{M'_u M_\ell}$, $u_{M'_u M'_\ell}$, $u_{M_u M_\ell}$ and $u_{M_u M'_\ell}$, which appear in the second argument of the Faddeeva function, are shifts in reduced frequency due to the possible presence of a magnetic field. They depend on the magnetic quantum numbers and the intensity of the magnetic field $\mathbf{B}$, i.e.

$$u_{M_u M_\ell} = \frac{\nu_L\left(g_u M_u - g_\ell M_\ell\right)}{\Delta\nu_{\mathrm{D}}}, \tag{1.21}$$

where g_u and g_ℓ are the Landé factors of the upper and lower levels, respectively (Alsina Ballester et al., 2017), and the Larmor frequency is given by:

$$\nu_L = \frac{e\,||\mathbf{B}||}{4\pi m_e c}, \tag{1.22}$$

where e is the elementary charge, m_e is the electron mass, and c is the speed of light in vacuum.

Limit cases of $\mathcal{R}_Q^{\mathrm{II},KK'}$

As can be seen in the definition, the function $\mathcal{R}_Q^{\mathrm{II},KK'}$ of (1.13)) is defined for scattering angles Θ in the open interval $(0,\pi)$. Below we provide the analytical forms that it assumes in the limit cases of $\Theta = 0$ (forward scattering) and $\Theta = \pi$ (backward scattering).

In chapter 7, where a series of numerical applications will be reported, one of the most noticeable results concerns the so-called forward scattering Hanle effect (see Trujillo Bueno, 2001) in strong chromospheric lines, such as Ca I 4227 Å and Sr II 4078 Å. In the modeling of this effect, the role of the limit cases and their correct numerical treatment is fundamental.

Limit case of $\mathcal{R}_Q^{\mathrm{II},KK'}$ for $\Theta = \pi$

In the case of $\Theta = \pi$ the $\mathcal{R}_Q^{\mathrm{III},KK'}$ function (1.13) assumes the following analytical form:

$$\begin{aligned}\mathcal{R}_Q^{\mathrm{II},KK'}\left(u', u, \Theta = \pi\right) = & \frac{\Gamma_R}{\Gamma_R + \Gamma_I + \Gamma_E + 2\pi \mathrm{i} \nu_L g_u Q} \sum_{\{M_u M_u' M_\ell M_\ell'\}} \mathcal{C}_{KK'QM_uM_u'M_\ell M_\ell'} \\ & \times \frac{1}{2\sqrt{\pi}} \exp\left(-y^2\right) \\ & \times \frac{1}{2}\left[\Phi\left(a, u + u_{M_u' M_\ell'} - y\right) + \overline{\Phi}\left(a, u + u_{M_u M_\ell'} - y\right)\right]\end{aligned} \tag{1.23}$$

where the quantity y is:

$$y = \frac{u - u' + u_{M_\ell M_\ell'}}{2},$$

and the $\Phi\left(\cdot,\cdot\right)$ function is defined as:

$$\begin{aligned}&\Phi : \mathbb{R}^2 \longrightarrow \mathbb{C} \\ &\Phi(a,x) = \frac{a}{\pi(x^2+a^2)} + \frac{ix}{\pi(x^2+a^2)},\end{aligned}$$

where the real part of Φ is the Lorentzian distribution and the imaginary part is the associated dispersion function.

As it will be discussed in the next chapter, the numerical integration over u' (see Eq. (1.4)) in this limit case is far from trivial due to the properties of the integrand (see Sect. 2.4).

Limit case of $\mathcal{R}_Q^{\text{II},KK'}$ for $\Theta = 0$

In the case of $\Theta = 0$ the function $\mathcal{R}_Q^{\text{II},KK'}$ converges to the analytical form defined as follows:

$$\begin{aligned}\mathcal{R}_Q^{\text{II},KK'}\left(u', u, \Theta = 0\right) = & \frac{\Gamma_R}{\Gamma_R + \Gamma_I + \Gamma_E + 2\pi \mathrm{i} \nu_L g_u Q} \\ & \sum_{\{M_u M_u' M_\ell M_\ell'\}} \mathcal{C}_{KK'QM_u M_u' M_\ell M_\ell'} \\ & \qquad \frac{1}{2\sqrt{\pi}} \left[W\left(a, u' + u_{M_u' M_\ell}\right) + \overline{W}\left(a, u' + u_{M_u M_\ell}\right) \right] \\ & \qquad \delta\left(u' - u - u_{M_\ell M_\ell'}\right),\end{aligned} \tag{1.24}$$

where $\delta(\cdot)$ is the *Dirac delta function*. The analytical form of Equation (1.24) allows the integral (1.4) to u' to be solved analytically by applying the sifting property of the Dirac delta function. Considering that the function in (1.24) does not depend on the scattering angle (Θ), in order to solve the scattering integral in Eq. (1.4), it is possible to write integral over u' of the innermost component of the above equation as follows:

$$A = \int_{\mathbb{R}} \mathrm{d}u' \left[W\left(a, u' + u_{M_u' M_\ell}\right) + \overline{W}\left(a, u' + u_{M_u M_\ell}\right) \right] \, \delta\left(u' - u - u_{M_\ell M_\ell'}\right) \mathbf{I}\left(u', \mathbf{\Omega}'\right) .$$

By using the sifting property of the Dirac delta function, it can easily be solved analytically, i.e.

$$A = \left[W\left(a, u + u_{M_\ell M_\ell'} + u_{M_u' M_\ell}\right) + \overline{W}\left(a, u + u_{M_\ell M_\ell'} + u_{M_u M_\ell}\right) \right] \mathbf{I}\left(u + u_{M_\ell M_\ell'}, \mathbf{\Omega}'\right) . \tag{1.25}$$

We can thus conclude that for this limit case the solution of the integral over the incoming reduced frequencies u' assumes an analytical form, and consequently, its numerical calculation is simple and fast, since to solve it only two evaluations of the Faddeeva function and an interpolation of the incident radiation field are needed (see also Equation (1.4)). For this reason, we will not further discuss this case in this document.

1.4.5 The R^{III} redistribution matrix

The expression of the R^{III} redistribution matrix in the observer's frame is given by

$$
\begin{aligned}
&R^{\text{III}} \,:\, \mathbb{R}\times\mathbb{R}^2\times\mathbb{R}\times\mathbb{R}^2 \longrightarrow \mathbb{R}^{4\times4}\\
&R^{\text{III}}\left(u',\boldsymbol{\Omega}',u,\boldsymbol{\Omega}\right) = \sum_{K,K'=0}^{2}\sum_{Q=-K_{\min}}^{K_{\min}} \mathcal{R}_Q^{\text{III},KK'}\left(u',\boldsymbol{\Omega}',u,\boldsymbol{\Omega}\right)\mathcal{P}_Q^{KK'}\left(\boldsymbol{\Omega}',\boldsymbol{\Omega}\right),
\end{aligned}
\tag{1.26}
$$

The analytical expression of the $\mathcal{R}_Q^{\text{III},KK'}$ function is

$$
\begin{aligned}
&\mathcal{R}_Q^{\text{III},KK'} \,:\, \mathbb{R}\times\mathbb{R}^2\times\mathbb{R}\times\mathbb{R}^2 \longrightarrow \mathbb{C}\\
&\mathcal{R}_Q^{\text{III},KK'}\left(u',\boldsymbol{\Omega}',u,\boldsymbol{\Omega}\right) = \sum_{K''=|Q|}^{2J_u}\left(\beta_Q^{K''}-\alpha_Q\right)\\
&\qquad\times\sum_{M_u,M_u'}\sum_{M_\ell}\sum_{q,q'}\mathcal{B}_{K''K'QM_uM_u'M_\ell qq'}\sum_{M_u'',M_u'''}\sum_{M_\ell'}\sum_{q'',q'''}\mathcal{B}_{K''KQM_u''M_u'''M_\ell'q''q'''}\\
&\qquad\times\frac{1}{4}\Big[\mathcal{I}_{(M_uM_\ell),(M_u''M_\ell')}\left(u',u,\Theta\left(\boldsymbol{\Omega},\boldsymbol{\Omega}'\right)\right)+\mathcal{I}_{(M_uM_\ell),\overline{(M_u'''M_\ell')}}\left(u',u,\Theta\left(\boldsymbol{\Omega},\boldsymbol{\Omega}'\right)\right)\\
&\qquad\quad+\mathcal{I}_{\overline{(M_u'M_\ell)},(M_u''M_\ell')}\left(u',u,\Theta\left(\boldsymbol{\Omega},\boldsymbol{\Omega}'\right)\right)+\mathcal{I}_{\overline{(M_u'M_\ell)},\overline{(M_u'''M_\ell')}}\left(u',u,\Theta\left(\boldsymbol{\Omega},\boldsymbol{\Omega}'\right)\right)\Big],
\end{aligned}
\tag{1.27}
$$

where Θ is the scattering angle (1.11). The quantity $(\beta_Q^K-\alpha_Q)$ is the branching ratio for R^{III}. The quantity α_Q has already been defined in Eq. (1.14), while β_Q^K is defined as

$$
\beta_Q^K(\mathbf{r}) = \frac{\Gamma_R}{\Gamma_R+\Gamma_I(\mathbf{r})+D^{(K)}(\mathbf{r})+2\pi\mathrm{i}\,\nu_L(\mathbf{r})g_uQ}\,.
$$

where $D^(K)$ are the depolarizing rates due to elastic collisions (for more details, see Chapter 7 of Landi Degl'Innocenti and Landolfi (2004). The quantity $\mathcal{B}_{KK'QM_uM_u'M_\ell qq'}\in\mathbb{R}$ appearing is given by

$$
\begin{aligned}
\mathcal{B}_{KK'QM_uM_u'M_\ell qq'} &= (-1)^{1+J_u-M_u+q'}\sqrt{3(2J_u+1)(2K+1)(2K'+1)}\\
&\times\begin{pmatrix} J_u & J_\ell & 1\\ -M_u & M_\ell & -q\end{pmatrix}\begin{pmatrix} J_u & J_\ell & 1\\ -M_u' & M_\ell & -q'\end{pmatrix}\\
&\times\begin{pmatrix} J_u & J_u & K\\ M_u' & -M_u & -Q\end{pmatrix}\begin{pmatrix} 1 & 1 & K'\\ q & -q' & -Q\end{pmatrix}.
\end{aligned}
\tag{1.28}
$$

The indices q and q' can take the values -1, 0, and +1 and, recalling that a Wigner 3j symbol is zero if the sum of the arguments of the lower row is not zero, they guarantee that the selection rule for the magnetic quantum numbers $|M_u - M_\ell| \leq 1$ is satisfied. In the numerical algorithm, this selection rule can be taken into account in the loops over the magnetic quantum numbers (see Algorithm 2). If this is done, the sum over q and q' (as well as over q'' and q''') becomes redundant, and these indices can be fixed a priori: $q = M_\ell - M_u$ and $q' = M_\ell - M_u'$.

Algorithm 2: Algorithm for the calculation of the magnetic quantum numbers for the R^{III} given J_u and J_ℓ.

```
Function M^III ($J_u$, $J_\ell$) :
    for $M_u = -J_u \dots J_u$ do
        for $M_\ell = -J_\ell \dots J_\ell$ do
            if $|M_u - M_\ell| > 1$ then
                Continue
            for $M_u' = -J_u \dots J_u$ do
                if $\left|M_u' - M_\ell\right| > 1$ then
                    Continue
                Yield $M_u$, $M_u'$, $M_\ell$
```

The quantity

$$\mathcal{J}_{(M_u M_\ell),(M_u' M_\ell')} \; : \; \mathbb{R} \times \mathbb{R} \times [0, \pi] \longrightarrow \mathbb{C} \tag{1.29}$$

is given by

- If $\mathbf{\Omega}' \neq \mathbf{\Omega}, -\mathbf{\Omega}$, $(\Theta \in (0, \pi))$:

$$\begin{aligned} \mathcal{J}_{(M_u M_\ell),(M_u' M_\ell')}(u', u, \Theta) = \frac{1}{\pi^2 \sin\Theta} & \int \mathrm{d}y \exp\left(-y^2\right) \\ & \times W\left(\frac{a}{\sin\Theta}, \frac{u + u_{M_u M_\ell} + y\cos\Theta}{\sin\Theta}\right) \\ & \times \varphi\left(a, u' + u_{M_u' M_\ell'} + y\right) . \end{aligned} \tag{1.30}$$

- If $\mathbf{\Omega}' = \mathbf{\Omega}$ (backward scattering, $\Theta = \pi$):

$$\begin{aligned} \mathcal{J}_{(M_u M_\ell),(M_u' M_\ell')}(u', u, \Theta) = \frac{1}{\pi^{5/2}} & \int \mathrm{d}y \exp\left(-y^2\right) \\ & \times \varphi\left(a, u + u_{M_u M_\ell} + y\right) \\ & \times \varphi\left(a, u' + u_{M_u' M_\ell'} + y\right) . \end{aligned} \tag{1.31}$$

- If $\boldsymbol{\Omega}' = -\boldsymbol{\Omega}$ (forward scattering, $\Theta = 0$):

$$\begin{aligned}\mathcal{J}_{(M_u M_\ell),(M_u' M_\ell')}(u', u, \Theta) = \frac{1}{\pi^{5/2}} & \int \mathrm{d}y \exp\left(-y^2\right) \\ & \times \varphi\left(a, u + u_{M_u M_\ell} - y\right) \\ & \times \varphi\left(a, u' + u_{M_u' M_\ell'} + y\right) . \end{aligned} \qquad (1.32)$$

where Θ is the scattering angle (Eq. (1.11)), where $\varphi(y,x) = \frac{1}{y - \mathrm{i}x}$, and $(u_{M_u M_\ell})$ are the magnetic shift (see Eq. (1.21)). In the notation of Eq. (1.27), a bar over the first subscript indicates that the second term of the integrand is a complex conjugate and, similarly, a bar over the second subscript indicates that the third term is a complex conjugate.

We must emphasize that the integral over y, which appears in the definitions of $\mathcal{J}_{(M_u M_\ell),(M_u' M_\ell')}$, introduces a fourth dimension of the scattering integral (see also (1.4)), which makes its numerical computation very challenging. Indeed, the time complexity to compute ε^ℓ by including R^{III} in its angle-dependent formulation is $O\left({N_\Omega}^2 {N_\nu}^3\right)$.

1.4.6 The R^{III} redistribution matrix under the CRD assumption.

The exact (angle-depend) expression of the R^{III} redistribution matrix presented in the previous section is very challenging to be treated numerically (see also Chapter 6). For this reason, in numerical applications, the R^{III} matrix is usually considered by applying the simplifying assumption of *complete frequency redistribution* (CRD) in the observer fram.

This implies that the joint probability of absorbing radiation with a given frequency ν' and remitting radiation with frequency ν is given by the product of two independent profiles. This simplification allows numerous algebraic manipulations, leading to fast computational algorithms (see Chapter 6. for a detailed discussion).

The function $\mathcal{R}_Q^{\mathrm{III},KK'}$ in the complete frequency redistribution (CRD) formulation is given by (see also Eq. (49) of (Bommier, 1997b))

Using the formalism of irreducible spherical tensors for polarimetry, the $R^{\mathrm{III-CRD}}$ matrix under the assumption of CRD in the observer's frame, hereafter indicated as $R^{\mathrm{III-CRD}}$, can be writen as

$$R^{\mathrm{III-CRD}}\left(u', \boldsymbol{\Omega}', u, \boldsymbol{\Omega}\right) = \sum_{K,K'=0}^{2} \sum_{Q=-K_{min}}^{K_{min}} \mathcal{R}_Q^{\mathrm{III-CRD},KK'}\left(u', u\right) \mathcal{P}_Q^{KK'}\left(\boldsymbol{\Omega}', \boldsymbol{\Omega}\right) . \qquad (1.33)$$

The function $\mathcal{R}_Q^{\text{III}-\text{CRD},KK'}$ is given by (see Eq. (49) of Bommier, 1997b)

$$\mathcal{R}_Q^{\text{III}-\text{CRD},KK'}\left(u',u\right) = \sum_{K''=|Q|}^{2J_u} \left(\beta_Q^{K''} - \alpha_Q\right) \Phi_Q^{K''K'}(u)\, \Phi_Q^{K''K}\left(u'\right), \tag{1.34}$$

where the quantities Φ are the so-called generalized profiles (see Eq. (A.42) for their definition).

The computational complexity of the exact expression of R^{III}, and the suitability of the $R^{\text{III}-\text{CRD}}$ approximation are analyzed in detail in Chapter 6.

1.4.7 The limit of CRD

The limit of CRD consists in assuming that in a scattering process there is no correlation between the frequencies of the incoming and scattered radiation (e.g. Mihalas, 1978; Landi Degl'Innocenti and Landolfi, 2004). This is generally a good approximation in dense plasmas, where collisions completely relax any coherence in the scattering processes. Indeed, the limit of CRD is usually suitable to model the intensity and polarization of photospheric lines (e.g., the Sr I line at 4607 Å, see Chapter 7), while it is not adequate for strong resonance lines forming in the chromosphere, for which coherent scattering and PRD effects are dominant.[3] Under the assumption of CRD, the RT problems becomes much lighter from a computational point of view, and fast algorithms (analogous to the one used to compute $R^{\text{III}-\text{CRD}}$) can be applied (see Chapter 6 for further discussions).

We consider the line emissivity in the limit of CRD as derived within the framework of the theory of polarization of Landi Degl'Innocenti and Landolfi (2004). Although that theory is not based on the redistribution matrix formalism, in order to better highlight the differences with the PRD case, hereafter we write their CRD line emissivity in terms of a scattering integral (1.4), introducing a suitable redistribution matrix, R^{CRD}. From the expressions of Chapt. 10 of Landi Degl'Innocenti and Landolfi (2004), we have

$$R^{\text{CRD}}\left(u',\mathbf{\Omega}',u,\mathbf{\Omega}\right) = \sum_{K,K'=0}^{2} \sum_{Q=-K_{min}}^{K_{min}} \mathcal{R}_Q^{\text{CRD},KK'}\left(u',\mathbf{\Omega}',u,\mathbf{\Omega}\right) \mathcal{P}_Q^{KK'}\left(\mathbf{\Omega}',\mathbf{\Omega}\right), \tag{1.35}$$

[3] It should be observed that there are also chromospheric lines, such as Ca II 8542 Å or He I 10830 Å, which are not particularly strong in the solar spectrum, and can be correctly modeled under the assumption of CRD.

The redistribution function $\mathcal{R}_Q^{\text{CRD},KK'}$ is given by

$$\begin{aligned} &\mathcal{R}_Q^{\text{CRD},KK'} : \mathbb{R}\times\mathbb{R}^2\times\mathbb{R}\times\mathbb{R}^2 \longrightarrow \mathbf{C} \\ &\mathcal{R}_Q^{\text{CRD},KK'}\left(u',u\right) = \beta_Q^K\,\Phi_Q^{KK'}(u)\,w^{(K)}_{J_uJ_\ell}(-1)^Q\phi\left(u'\right), \end{aligned} \tag{1.36}$$

where $\beta_Q^{K''}$ is defined in Eq. (A.4) and $\Phi_Q^{KK'}$ are the generalized profiles (A.42). The quantity $w^{(K)}_{J_uJ_\ell}$ is given by

$$w^{(K)}_{J_uJ_\ell} = (-1)^Q(-1)^{1+J_\ell+J_u}\sqrt{3(2J_u+1)}\begin{Bmatrix} 1 & 1 & K \\ J_u & j_u & J_\ell \end{Bmatrix}$$

where the operator

$$\begin{Bmatrix} j_1 & j_2 & j_3 \\ j_4 & j_5 & j_6 \end{Bmatrix}$$

is the *Wigner's 6J symbol* (Landi Degl'Innocenti and Landolfi, 2004, Chap. 2). Finally the profile ϕ is:

$$\phi\left(u',\mathbf{\Omega}'\right) = \frac{1}{\sqrt{\pi}\Delta\nu_{\text{D}}}V\left(a,u'\right)$$

1.4.8 Normalization

In numerical applications, it is important that the values of the redistribution matrix computed to evaluate the scattering integral are suitably normalized so that there are no spurious sources or sinks of photons (e.g., Adams et al., 1971). This normalization can be performed through Kirchhoff's law, which states that at thermodynamic equilibrium the ratio between the emission and absorption coefficients must be equal to the Planck function (in the Wien limit when stimulated emission is neglected, as in this work).

Generalizing Kirchhoff's law to the polarized case (the polarization being due to the Zeeman effect produced by a magnetic field), we must have

$$\hat{\varepsilon}_i^{\ell,sc}(\nu,\mathbf{\Omega}) = (1-\epsilon)\,W_T(\nu)\,\eta_i(\nu), \tag{1.37}$$

where W_T is the Planck function in the Wien limit, $\hat{\varepsilon}_i^{\ell,sc}$ is the line scattering contribution to the emissivity, calculated at the thermodynamic equilibrium (i.e., assuming an isotropic and unpolarized incident radiation field equal to W_T), and $\epsilon = \Gamma_I/(\Gamma_R+\Gamma_I)$ is the branching ratio between scattering and thermal processes. Recalling Eq. (1.4), we have:

$$\hat{\varepsilon}_i^{\ell,sc}(\nu,\mathbf{\Omega}) = k_L\int \mathrm{d}\nu'\oint\frac{\mathrm{d}\mathbf{\Omega}'}{4\pi}\sum_{j=0}^{3}R(\nu',\mathbf{\Omega}',\nu,\mathbf{\Omega})_{ij}\,\delta_{i0}\,W_T(\nu'). \tag{1.38}$$

To normalize the redistribution matrix we thus divide it by the factor

$$C_i(\nu, \mathbf{\Omega}) = \frac{\hat{\varepsilon}_i^{\ell,sc}(\nu, \mathbf{\Omega})}{(1-\epsilon)\, W_T(\nu)\, \eta_i(\nu)}\,, \tag{1.39}$$

where $\hat{\varepsilon}_i^{\ell,sc}$ is calculated with the proposed numerical algorithm. In general, the main source of inaccuracy is due to the finite frequency interval considered for the quadrature over ν' in Eq. (1.4).

1.5 Impact of bulk velocity

In the previous sections, we considered the expression of the emission coefficient without considering a relevant variable of the problem, the *bulk velocity* $\mathbf{v}_b$, which describes the flow of the atmospheric plasma. The bulk velocity affects the expression of the emission coefficients in the observer's reference frame by Doppler shifting the frequencies of the incoming and outgoing radiation fields in the redistribution matrix as follows:

$$\varepsilon\,(u, \mathbf{\Omega}) = k_L \int du' \oint \frac{d\mathbf{\Omega}'}{4\pi} \left[R\left(u' + u'_\zeta,\, \mathbf{\Omega}',\, u + u_\zeta,\, \mathbf{\Omega}\right)\right] \mathbf{I}\left(u', \mathbf{\Omega}'\right), \tag{1.40}$$

where the frequency shifts u'_ζ and u_ζ which depend on $\mathbf{v}_b$ and $\mathbf{\Omega}'$ and on $\mathbf{\Omega}$, respectively, are given by:

$$\begin{aligned} u_\zeta &= w_T\,(\mathbf{v}_b \cdot \mathbf{\Omega}) \\ &= w_T \left\|\mathbf{v}_b\right\| \cos(\zeta) \end{aligned} \tag{1.41}$$

and where

$$\begin{aligned} u_\zeta &= w_T \left(\mathbf{v}_b \cdot \mathbf{\Omega}'\right) \\ &= w_T \left\|\mathbf{v}_b\right\| \cos\left(\zeta'\right) \end{aligned} \tag{1.42}$$

where $w_T = \left(\sqrt{\frac{2K_bT}{M_a} + v_t^2}\right)^{-1}$ and ζ and ζ' are the angles between the direction of the bulk velocity $\mathbf{v}_b$ and the direction of the outgoing $\mathbf{\Omega}$ and incoming $\mathbf{\Omega}'$ beams, respectively.

The angle-dependent terms u'_ζ and u_ζ prevent the use of the fast algorithms described in section 4.1. This problem can be overcome by considering the *comoving reference frame*, i.e. the reference frame in which the bulk velocity of the plasma is zero. In the comoving reference frame, the expression of the emission coefficients is

$$\varepsilon_{\text{com}}\,(u, \mathbf{\Omega}) = k_L \int du' \oint \frac{d\mathbf{\Omega}'}{4\pi} \left[R\left(u', \mathbf{\Omega}', u, \mathbf{\Omega}\right)\right] \mathbf{I}\left(u' + u'_\zeta, \mathbf{\Omega}'\right), \tag{1.43}$$

where **I** is expressed in the observer's frame. In this way, the redistribution matrix does not depend on the Doppler shifts induced by the bulk velocity. The transformation of the emission coefficients from the comoving reference frame to the observer reference frame is given by

$$\varepsilon\left(u,\mathbf{\Omega}\right)=\varepsilon_{\rm com}\left(u-u_{\zeta},\mathbf{\Omega}\right). \tag{1.44}$$

This calculation strategy requires shifting the incident radiation field in Eq. (1.43), and then shifting the resulting emission coefficients through Eq. (1.44). In a numerical application, these two steps can be performed by using an interpolation (e.g., a cubic spline). As it will be shown in Sect. 4.1, the computational cost of these interpolations is more than compensated by the fast algorithms that can be implemented when the redistribution matrix does not depend on the Doppler shifts.

1.6 Start of the project

At the beginning of the project, we had available a prototype code for calculating the quantities $\mathcal{R}_Q^{\rm II,KK'}$ of Eq. (1.13). This code was based on square gird for the frequencies that is a grid with the same nodes on ν' and ν. The approach based on a square frequency grid immediately revealed to be completely unsuitable because the typical grid of the problem for the frequencies of the emitted radiation (i.e., a grid that suitably samples the considered spectral line) does not allow a proper sampling of the u subdomains in which the function (1.13) is relevant. As a second attempt, we implemented a quadrature method inspired to the one that was successfully used to evaluate the same integral, but considering the redistribution matrix $R^{\rm II}$ under the so-called angle-averaged (AA) approximation. Also in this case, the results were not satisfactory, as can be seen in Fig. 1.5, where the profiles obtained with that method (lower panels) are compared to those calculated with the most recent version of the code (upper panels). The inaccuracy of the initial method was mainly because it did not properly consider the strong dependence of the function (1.13) on the scattering angle Θ (see Sect. 2.3), with the case $\Theta=\pi$ being particularly critical (see Sect. 2.4). It should also be noted that the computation of these profiles (at a single spatial point and for a single direction) initially required several minutes, while the most recent algorithm (see Sect. 6) needs less than a second. Due to the lack of scientific literature discussing methods for calculating the redistribution matrix (1.12) and integrals (1.4), without considering the AA approximation, we decided to develop a new original strategy.

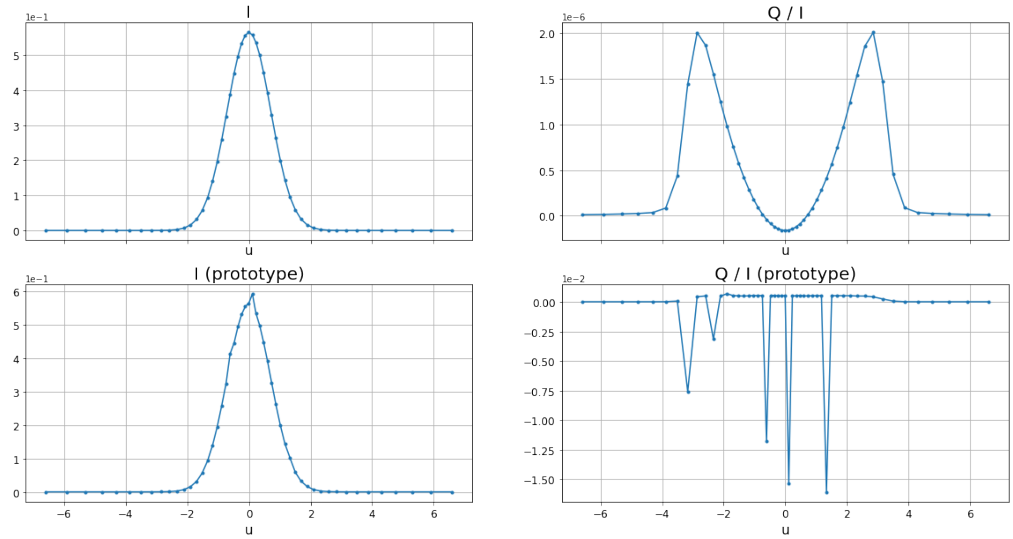

Figure 1.5. *Examples of the profiles of the emission coefficient ε_I (left panels) and of the ratio $\varepsilon_Q/\varepsilon_I$ (right panels), calculated with the most recent version of the algorithm (upper panels) and with the prototype code, implementing a quadrature strategy similar to the one used when considering the AA approximation (lower panels).*

1.7 Numerical solution of the RT equation

To reliably test the algorithm that we have developed during this thesis to calculate the contribution of R^{II} to the emission coefficients (see Chapter 4), we developed a set of routines to evaluate the other contributions to the emissivity (see Sect. 1.4.2) and (see Sect. 1.4.2) and the elements of the propagation matrix (see Sect. 1.4.1).

Subsequently, we implemented a method to solve the RT equation (1.1) along a given direction $\mathbf{\Omega}$ (this procedure is generally referred to as *formal solution*). For this purpose, we considered formal solvers belonging to the *Diagonal Element Lambda Operators (DELO)* family (e.g., Rees et al., 1989; Janett et al., 2017). These commonly used solvers are exponential methods for the numerical solution of ordinary differential equations, which rely on the conversion of the spatial scale to the optical depth scale (e.g., Janett et al., 2017).

The optical depth τ is a frequency-dependent metric defined by the following differential equation:

$$\mathrm{d}\tau\,(s, \mathbf{\Omega}, \nu) = -\eta_I(\mathbf{r}, \mathbf{\Omega}, \nu)\mathrm{d}s\,, \tag{1.45}$$

where η_I is the opacity of the medium (i.e. the diagonal element of the propagation matrix, see Eq. (1.2)), and s the spatial coordinate along the considered propagation direction $\mathbf{\Omega}$. It can be seen that the optical depth is defined in the opposite direction than the propagation of the beam, which reflects the point of view of the observer.

The total cumulative optical depth $\Delta\tau$ from an observer at coordinate $\mathbf{r}_0$ to a point $\mathbf{r}_1$ following the direction of the beam $\mathbf{\Omega}$ is given by the solution of the above differential equation, i.e:

$$\Delta\tau(\mathbf{r}_0, \mathbf{r}_1, \nu) = -\int_0^1 \eta_I(\mathbf{r}_0 + t\,(\mathbf{r}_1 - \mathbf{r}_0), \mathbf{\Omega}, \nu)\,\mathrm{d}t\,, \tag{1.46}$$

where t is the integration variable. For more details on the optical depth see also Landi Degl'Innocenti (2007); Janett and Paganini (2018).

By applying the optical depth scale, the radiative transfer equation (1.1) takes the form:

$$\left[\frac{\mathrm{d}}{\mathrm{d}\tau_\nu} - \mathbf{1}\right]\mathbf{I}(\tau_\nu) = -\mathcal{K}(\tau_\nu)\mathbf{I}(\tau_\nu) + \tilde{\epsilon}(\tau_\nu)\,, \tag{1.47}$$

where $\mathbf{1} \in \mathbb{R}^{4\times4}$ is the identity matrix, $\tilde{\epsilon} = \varepsilon/\eta_I$ is the modified emission vector, and $\mathcal{K} = K/\eta_I - \mathbf{1}$ is the modified propagation matrix (e.g., Landi Degl'Innocenti, 2007; Janett et al., 2017).

In the first experiments, we considered the Ca I 4227 Å line, and we used the one-dimensional (1D) semi-empirical solar atmospheric `model C` of Fontenla et al. (1993, herafter FAL-C). In 1D models, the physical quantities are assumed to be horizontally invariant, and the atmosphere is discretized only in altitude. We solved the RT equation applying the *DELO linear* formal solver (e.g., Janett et al., 2017), using as incoming radiation field the one calculated with RH (Uitenbroek, 2001), accounting for PRD effects under the angle-averaged (AA) simplifying approximation.

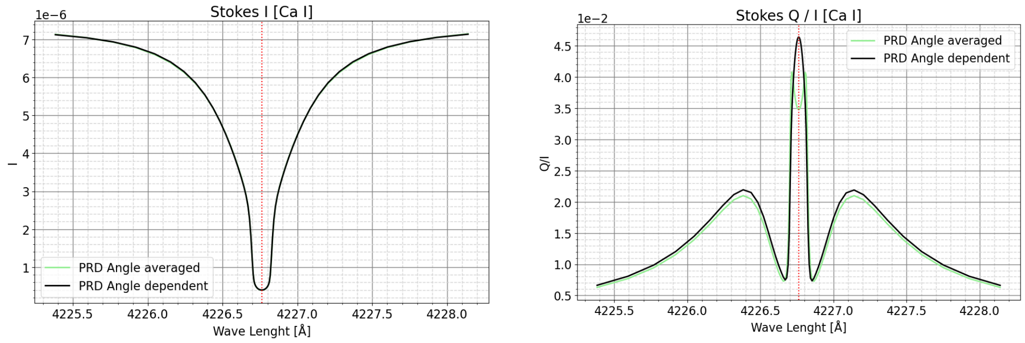

Figure 1.6. *Intensity (left panel) and Q/I (right panel) profiles of the Ca* I *line at 4227 Å, calculated in model C of Fontenla et al. (1993), for a line of sight with* $\mu = \cos\theta = 0.1$*, with* θ *the angle between the local vertical and the direction of the considered radiation. No magnetic fields or bulk velocities are considered. The profiles calculated after one formal solution accounting for AD PRD effects (black line) are compared to those calculated under the AA approximation (green line). The plot reported here is the result of the first successful application of the algorithm for calculating the emissivity including PRD effects in the most general angle-dependent formulation.*

The results of these first applications were very interesting, as they highlighted significant differences between PRD calculations carried out under the AA approximation and in the general angle-dependent (AD) case. This can be clearly seen in Fig. 1.6, which shows the intensity and Q/I profiles of the Ca I line at 4227 Å, obtained after performing a formal solution along a given direction in the atmospheric model mentioned above. The figure shows that the central peak of the Q/I profile obtained under the AA approximation has a throat in the core, which is not found in the observations. However, the peak obtained when the AD-PRD effects are taken into account is sharper and higher and is much better in agreement with the observed profiles. This result highlights the importance of including AD PRD effects in calculations (also in 1D) and motivated an in-depth investigation on this spectral line, which has been published in Janett et al. (2021a). A result from that investigation is reported in Fig. 1.7. More results highlighting the differences between AA and AD PRD calculations in different spectral lines are presented in Chapter 7.

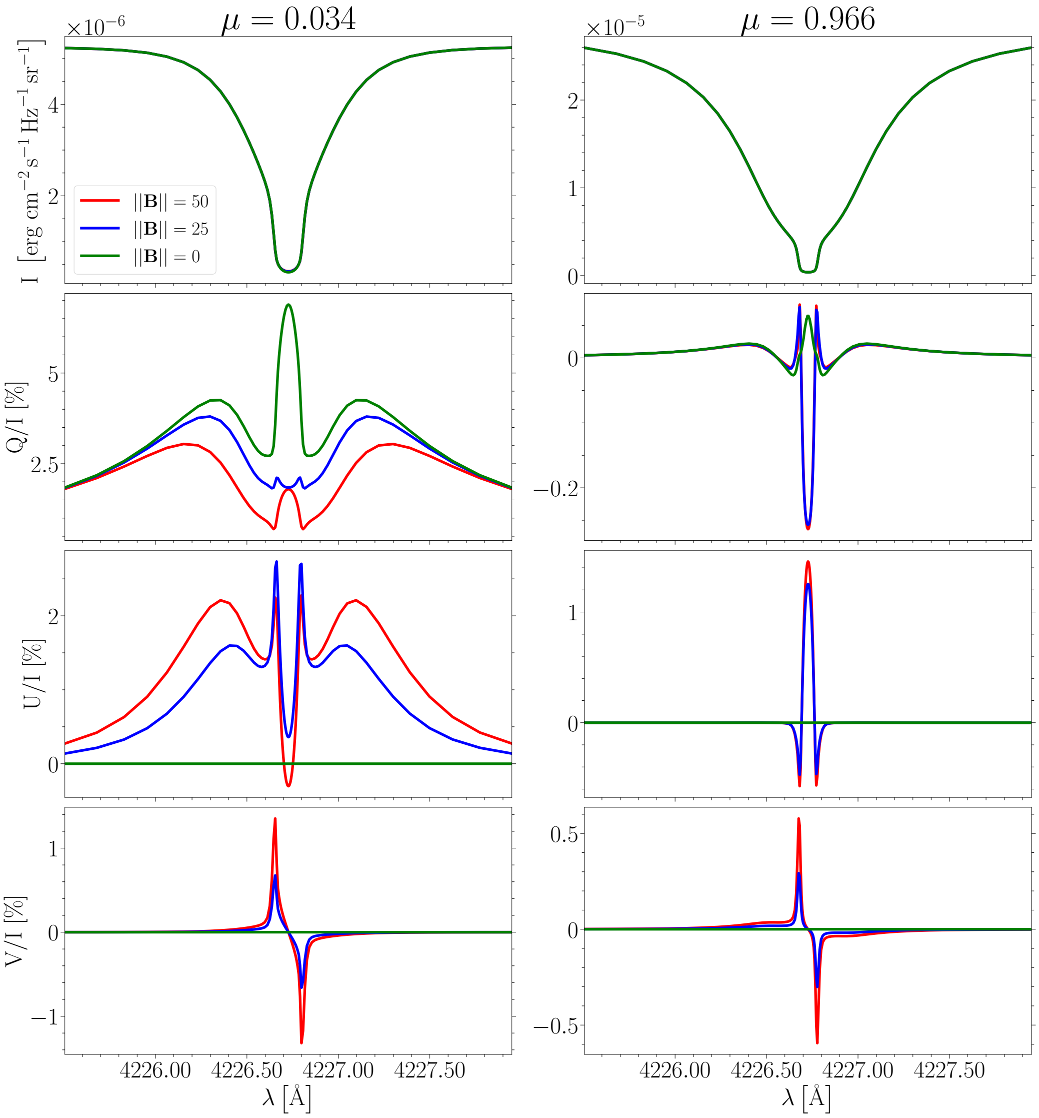

Figure 1.7. *Stokes parameters of the Ca I line at 4227 A, calculated in the 1D semi-empirical model C of Fontenla et al. (1993), for an emergent direction with* $\mu = \cos(\theta) = 0.033$ *(left column), and* $\mu = 0.966$ *(right column). The plotted profiles are obtained in the absence of magnetic fields (green curves) and in the presence of magnetic fields with* $\theta_B = \pi/2$ *and* $\chi_B = \pi/4$ *of 25 G (blue curves) and 50 G (red curves). The RT equation is solved by applying the DELO-linear formal solver, while the whole RT problem is solved via Krylov iterations (Benedusi et al., 2022). A spherical grid with 96 directions and a frequency grid with 99 nodes have been considered. Taking into account that the atmospheric model has 70 discrete altitudes, the total number of d.o.f. is 2661120.*

1.8 Solution strategy for the non-LTE RT problem

The algorithm for the computation of the emissivity accounting for angle-dependent PRD effects has then been implemented in both 1D and 3D codes for the solution of the full non-LTE RT problem for polarized radiation. In this section, we briefly describe the solution strategy that has been developed by our group during the last few years and that is implemented in such codes (see also Benedusi et al. (2022); Benedusi et al. (2023)).

The following subsection are extracted from the paper:
Assessment of the CRD approximation for the observer's frame R^{III} redistribution matrix Riva et al. (2023)

1.8.1 Linearization

The RT problem consists in finding a self-consistent solution for the RT equation (1.1) and the equation for the scattering contribution to the emissivity (1.4). This problem is in general non-linear because of the factor k_L appearing both in the elements of the propagation matrix and in the emission coefficients. This factor is proportional to the population of the lower level, which in turn depends non-linearly on the incident radiation field through the statistical equilibrium equations.

We linearize the problem with respect to $\mathbf{I}$, by fixing a-priori the population of the lower level, and thus k_L. In such a scenario, whose suitability is discussed in Benedusi et al. (2021), the propagation matrix is independent of $\mathbf{I}$ and the emission coefficients depend on it linearly through the scattering integral (1.4).

The population of the lower level can be taken either from the atmospheric model (if provided) or from independent RT calculations. Since the population of ground (or metastable) levels is expected to be marginally impacted by the polarization of the radiation field, the latter calculations can be carried out with available RT codes that solve the non-linear non-LTE RT problem possibly neglecting polarization and magnetic fields. It must be observed that such codes generally allow considering multi-level atomic models, and thus provide accurate estimates of the lower level population. Once the population of the lower level has been pre-computed, we keep it fixed and we solve the linear non-LTE RT problem inlcuding polarization and magnetic fields.

1.8.2 Numerical method for the RT problem

Following Janett et al. (2021b); Benedusi et al. (2021, 2022), we first present an algebraic formulation of the considered linearized RT problem for polarized radiation. Starting from this formulation, we then apply a parallel solution strategy, based on Krylov iterative methods with physics-based preconditioning. This strategy allows us to routinely solve the RT problem for polarized radiation in semiempirical models of the solar atmosphere, considering the exact expression of both R^{II} and R^{III} redistribution matrices. The same approach, coupled with a new domain decomposition technique, has recently been generalized to the 3D case Benedusi et al. (2023).

As previously pointed out (see Sect. 1.4.1), the radiation field and the other quantities entering the RT problem are in general functions of three continuous variables: the spatial point $\mathbf{r}$, the propagation direction $\mathbf{\Omega}$, and the frequency ν. These variables are discretized by introducing a spatial grid $P = \{\mathbf{r}_k\}_{k=1}^{N_p}$, which is provided by the considered solar atmosphere model, an angular grid $S = \{\mathbf{\Omega}_i\}_{i=1}^{N_\Omega}$, where the angular nodes are those of the spherical quadrature chosen to evaluate the angular integral of eq. (1.4) (see Section 3.4), and the frequency grid $U = \left\{\nu_j\right\}_{j=1}^{N_\nu}$ chosen to sample the spectral line of interest. Since the emission coefficient is calculated locally, hereafter we will consider a single point of the spatial grid. At any spatial point $\mathbf{r}_k$ the radiation field (i.e. the four Stokes parameters) is stored in a three-dimensional grid G, given by the Cartesian product between the frequency grid U and the spherical grid S, that is, $G = U \times S$. The emission vector must also be calculated at the nodes of the grid $G = \left\{\mathbf{\Omega}_i, \nu_j\right\}_{i,j=1}^{N_\Omega,N_\nu}$. As discussed previously, the emission vector will be calculated in terms of reduced frequencies u_i instead of frequencies ν_i (see Eq. (1.7) and Sect. 4.1).

Normally we discretize the inclination through two Gauss-Legendre grids with $N_\theta/2$ points each, one for $\mu = \cos(\theta) \in (-1,0)$, and one for $\mu \in (0,1)$ (see also Sec. 3.4). For the azimuth, we chose a grid of N_χ equidistant points. The angular quadrature is then the spherical Cartesian-product (e.g. Davis and Rabinowitz, 2007) of the Gauss-Legendre rule for the inclination and the trapezoidal rule for the azimuth. This approach is commonly used in RT applications and allows the implementation of fast algorithms.

In a 1D (plane-parallel) setting, the spatial coordinate $\mathbf{r} = (x,y,z)$ can be replaced by the vertical coordinate $z \in [z_{\min}, z_{\max}]$, and the RT equation (1.1) can be rewritten as

$$\cos(\theta)\frac{\mathrm{d}}{\mathrm{d}z}I_i(z,\theta,\chi,\nu) = -\sum_{j=1}^{4} K_{ij}(z,\theta,\chi,\nu) I_j(z,\theta,\chi,\nu) + \varepsilon_i(z,\theta,\chi,\nu)\,. \tag{1.48}$$

The spatial grid $\{z_k\}_{k=1}^{N_s}$ is provided by the considered atmospheric model. Moreover, we assume that the radiation entering the atmosphere from the lower boundary is isotropic, unpolarized, and equal to the Planck function and that no radiation is entering from the upper boundary. Equation (1.48) is thus equipped with the following boundary conditions:

$$\begin{aligned} &\mathbf{I}(z_{\min},\theta,\chi,\nu) = [B_P(\nu,T(z_{\min})),0,0,0]^T \qquad && \theta\in[0,\pi/2),\ \forall\chi,\ \forall\nu,\\ &\mathbf{I}(z_{\max},\theta,\chi,\nu) = \mathbf{0} && \theta\in(\pi/2,\pi],\ \forall\chi,\ \forall\nu, \end{aligned}$$

where B_P is the Planck function at the effective temperature T at $z_{\min}$. Given the knowledge of elements of the propagation matrix and the emission coefficients at all height points $\{z_k\}_{k=1}^{N_s}$, for a given direction (θ_l,χ_m) and frequency ν_n, the RT equation (1.1) can be numerically solved along that direction and at that frequency by applying a suitable integrator. In this work, we use the L-stable DELO-linear method combined with a linear conversion to optical depth (e.g. Janett et al., 2017). An analysis of the stability properties of this method can be found in Janett and Paganini (2018).

Being $N = 4N_sN_\theta N_\chi N_\nu$ the total number of unknowns, we introduce the collocation vectors $\mathrm{I}\in\mathbb{R}^N$ and $\epsilon\in\mathbb{R}^N$, which contain the numerical approximations at all nodes for the Stokes parameters and the emission vector, respectively. Given ϵ, the solution of all discretized RT equations (1.48) can be written as

$$\mathrm{I} = \Lambda\epsilon + \mathbf{t}\,, \tag{1.49}$$

where $\Lambda:\mathbb{R}^N\to\mathbb{R}^N$ is the transfer operator, which encodes the formal solver and the propagation matrix, and $\mathbf{t}\in\mathbb{R}^N$ is a vector encoding the boundary conditions. Similarly, given I, the discrete computation of the emission coefficients can be written as

$$\epsilon = \epsilon^{\mathrm{sc}} + \epsilon^{\mathrm{th}} = \Sigma\mathrm{I} + \epsilon^{\mathrm{th}}\,, \tag{1.50}$$

where ϵ^{sc} and ϵ^{th} encode the scattering and thermal contributions (including both line and continuum processes), respectively, as described in Sect. 1.4.2. The scattering operator $\Sigma:\mathbb{R}^N\to\mathbb{R}^N$ encodes the numerical evaluation of the scattering integral (1.4) and thus depends on the chosen numerical quadratures. In general, the operator Σ is given by the sum of different components, namely,

$$\Sigma = \Sigma^{\text{II}} + \Sigma^{\text{III}} + \Sigma^{\text{c}} \tag{1.51}$$

where Σ^{II} and Σ^{III} encode the contributions of R^{II} and R^{III}, respectively, and Σ^{c} the contribution of the continuum. The vector $\epsilon^{\text{th}} \in \mathbb{R}^N$ encodes the thermal emissivity.

Under the assumption that k_L is known a priori (see Section 1.8.1), the operators Λ and Σ are linear with respect to I. Combining (1.49) and (1.50), the whole discrete RT problem can be formulated as a linear system of size N with unknown I, that is,

$$(Id - \Lambda\Sigma)\mathrm{I} = \Lambda\epsilon^{\text{th}} + \mathbf{t}\,, \tag{1.52}$$

where $Id \in \mathbb{R}^{N\times N}$ is the identity matrix. The right hand-side vector $\Lambda\epsilon^{\text{th}} + \mathbf{t}$ can be computed a priori by solving (1.48) with thermal contributions only (i.e., by performing a single formal solution with $\epsilon^{\text{sc}} = \mathbf{0}$). The action of the matrices Λ, Σ, and $Id - \Lambda\Sigma$ can be encoded in a matrix-free form (see Benedusi et al., 2022).

We solve the linear system (1.52) by applying a matrix-free, preconditioned GMRES method. Preconditioning is performed by describing scattering processes in the limit of CRD (see Sect. 1.4.7). This corresponds to substituting the operator $\Sigma^{\text{II}} + \Sigma^{\text{III}}$ with a new operator Σ^{CRD} that is much cheaper to evaluate. The explicit expression of Σ^{CRD} for the atomic model considered can be extracted from Sect. 1.4.7. The reader is referred to Benedusi et al. (2022) for more details on this solution strategy.

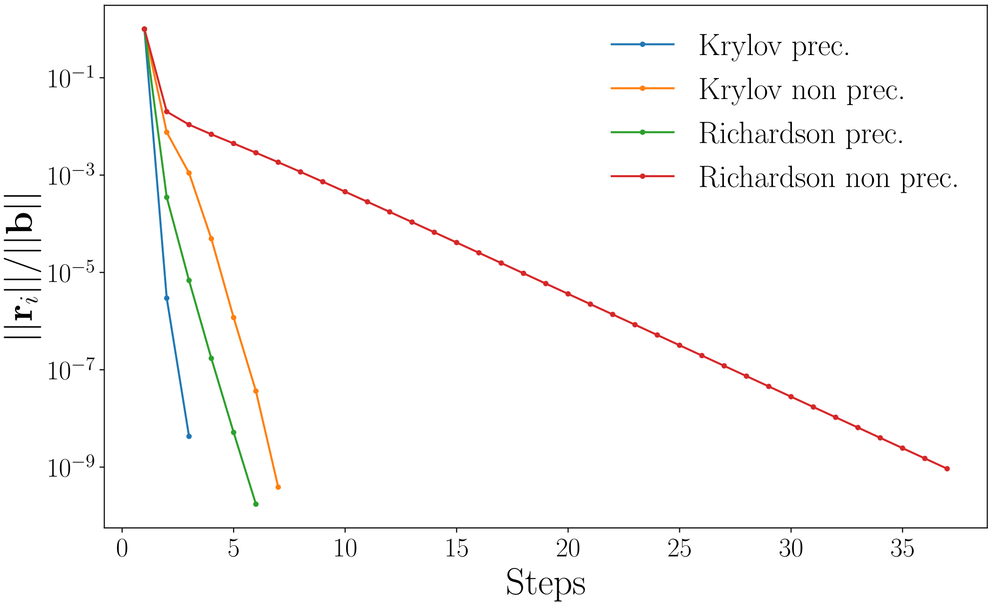

Figure 1.8. *Convergence rate of different methods for solving a linearized non-LTE RT problem for polarized radiation. The plot reports the residual $||r_i||/||b||$ as a function of the iteration step. The reference problem is formulated considering a 1D atmospheric model extracted from a 3D MHD simulation, including height-dependent magnetic and bulk velocity fields, with 118 discrete heights (the same model is used for a series of applications shown in Sect. 6.5). We model the Sr* I *4607 Å line considering 129 frequency nodes. An angular grid with 96 discrete directions is used. The blue line refers to the preconditioned Krylov method mentioned in this section.*

Chapter 2

Analytical study of the R^{II} redistribution matrix

2.1 Introduction to the numerical integration of the R^{II} contribution to the emission coefficient

The first main goal of this Ph.D. work is to develop reliable and efficient numerical methods to evaluate R^{II} and calculate the quantity:

$$\varepsilon^{\ell,II}(u,\mathbf{\Omega}) = k_L \int_{\mathbb{R}} \mathrm{d}u' \oint \frac{\mathrm{d}\mathbf{\Omega}'}{4\pi} R^{\text{II}}\left(\mathbf{\Omega}', u', \mathbf{\Omega}, u\right) \mathbf{I}\left(u', \mathbf{\Omega}'\right). \tag{2.1}$$

The numerical integration of Eq. (2.1) is a task that involves various difficulties. The **first** main problem is that the computation of the function $\mathcal{R}_Q^{\text{II},KK'}$ (1.13) (and consequently the redistribution matrix (1.12)) requires complex algorithms so that this single task can saturate the *time-to-solution* (> 90%) of an application for solving the whole RT problem (e.g. Benedusi et al., 2022). The **second** is the control of the numerical stability of the quadrature routine, which can be ensured only if all the parameters of the integrands are properly taken into account.

More in detail the four main challenges related to the numerical integration of the emission coefficient can be summarized as follows:

1. The computation of the *Faddeeva function* needs complex and slowly converging numerical procedures (e.g. Oeftiger et al., 2016).

2. The subdomain of $\mathbb{R}^3$ where the function $\mathcal{R}_Q^{\text{II},KK'}$ (1.13) is relevant is non-uniform and is strongly influenced by its parameters and arguments.

3. The limit case of $\Theta = \pi$ is critical in the sense that a failure of its numerical integration can lead to a failure of the entire computation of the emission coefficient. Moreover, it is difficult to define an accurate quadrature rule for this case.

4. A very large number of evaluations of the redistribution matrix R^{II} are required to solve the whole RT problem (see also Chapter 5).

2.2 Computational time complexity

We consider an atomic model for which there are N_M combinations of the *magnetic quantum numbers* (i.e. N_M elements in the sum over M_u, M'_u, M_ℓ and M'_ℓ appearing in Eq. (1.13), generated through Algorithm 1) and N_T combinations of the tensorial indices (i.e. N_T elements in the sum over K, K', and Q appearing in Eq. (1.12)). Indicating with N_ν and N_Ω the number of frequencies and directions used to discretize the incoming radiation field in one spatial point (see Section 1.8.2), the time complexity (e.g. Sipser, 1996) for calculating the emission vector at *all* outgoing directions and frequencies in one spatial point is $O\left({N_\Omega}^2 {N_\nu}^2 N_M N_T\right)$.

As noted previously, the numerical evaluation of the Faddeeva function is particularly demanding (Oeftiger et al., 2016). Normally, its computation is done using convergent series or iterative algorithms, where the convergence rate is generally slow and varies greatly depending on the algorithm and the argument (e.g. Wells, 1999; Mohankumar and Sen, 2019). For our application, we chose the method of Johnson (2012), where a hybrid approach (based on different algorithms) is used, ensuring good performance and accuracy. The time complexity of the evaluation of the Faddeeva function alone in the calculation of the emission coefficient $\varepsilon^{\ell,II}$, at one spatial point and at all frequencies and diffused directions, is of the order of $O\left({N_\Omega}^2 {N_\nu}^2 N_M\right)$.

From the above considerations, we can conclude that one of the main objectives in the development of an algorithm for the computation of the emission coefficient $\varepsilon^{\ell,II}$ (2.1) is the minimization of the evaluations of the Faddeeva function. Second, but not least, it is necessary to have an algorithm capable of controlling the risk of error by switching between different quadrature strategies according to the parameters of the functions, especially in the limit case of $\Theta = \pi$.

2.3 The two main components of $\mathcal{R}_Q^{\mathrm{II},KK'}$

We first observe that the behavior of the integrands in eq. (2.1) is strongly driven by the redistribution matrix (1.12) and only slightly by the incoming radiation field, which is usually well-conditioned and "smooth enough". For this reason, in order to develop a reliable quadrature method over u', we first isolated and analyzed the two main functional components of the redistribution function $\mathcal{R}_Q^{\mathrm{II},KK'}$ (see Eq. (1.13)), which in the absence of magnetic fields and bulk velocities are defined as

$$\begin{aligned} &h_{(u,\Theta,a)} : \mathbb{R} \longrightarrow \mathbb{R}_{>0} \\ &h_{(u,\Theta,a)}(\mathtt{t}) = \exp\left[-\left(\frac{u-\mathtt{t}}{2\sin(\Theta/2)}\right)^2\right] V\left(\frac{a}{\cos(\Theta/2)}, \frac{u+\mathtt{t}}{2\cos(\Theta/2)}\right) \end{aligned} \tag{2.2}$$

$$\begin{aligned} &g_{(u,\Theta,a)} : \mathbb{R} \longrightarrow \mathbb{R} \\ &g_{(u,\Theta,a)}(\mathtt{t}) = \exp\left[-\left(\frac{u-\mathtt{t}}{2\sin(\Theta/2)}\right)^2\right] L\left(\frac{a}{\cos(\Theta/2)}, \frac{u+\mathtt{t}}{2\cos(\Theta/2)}\right), \end{aligned} \tag{2.3}$$

where $V(\cdot,\cdot)$ and $L(\cdot,\cdot)$ are the Voigt and associated dispersion profiles of the Faddeeva function, respectively (see Eq. (1.17)). For the three parameters in the subscript we have

$$u \in \mathbb{R}, \quad \Theta \in (0,\pi), \quad a \in \mathbb{R}_{>0}. \tag{2.4}$$

For simplicity, we will use the notation $h_\beta(\cdot)$ and $g_\beta(\cdot)$, where the subscript β denotes the set of these three parameters, i.e.

$$\beta = (a, \Theta, u) \tag{2.5}$$

The following points summarize the most important properties of $h_\beta(\cdot)$ and $g_\beta(\cdot)$ that can be used to design a quadrature strategy:

- The functions are infinitely many times continuously differentiable, i.e. h_β, $g_\beta \in C^\infty$. Their derivatives never vanish.
- They asymptotically tend to zero at infinity:

$$\lim_{\mathtt{t}\to\pm\infty} h_\beta(\mathtt{t}) = 0, \qquad \lim_{\mathtt{t}\to\pm\infty} g_\beta(\mathtt{t}) = 0 \tag{2.6}$$

- As a consequence of the two previous properties, they are bounded.
- The codomain of $h_\beta(\cdot)$ is $\mathbb{R}_{>0}$, while the codomain of $g_\beta(\cdot)$ is $\mathbb{R}$.

- The function $g_\beta(\cdot)$ has exactly one real root at $\mathtt{t} = -u \in \mathbb{R}$, its global maximum and minimum are close to $-u$.
- No analytical expression for the stationary points is known.
- The global maximum of $h_\beta(\cdot)$ is close to $\mathtt{t} \sim u$.

Asymptotic behavior of $h_\beta(\cdot)$ for large values of u

For large values of the parameter u, the function $h_\beta(\cdot)$ assumes an asymptotic behavior where the argument of the maximum of $h_\beta(\cdot)$ asymptotically converges to u. Therefore, we can conclude that the limit of the argument of the maximum of $h_\beta(\cdot)$ subtracted by u tends to zero:

$$\lim_{u\to\pm\infty}\left[\arg\max_{\mathtt{t}\in\mathbb{R}}\left[h_{(u,\Theta,a)}(\mathtt{t})\right](u)-u\right]=0 \tag{2.7}$$

When we study how $h_\beta(\cdot)$ behaves when the parameter $|u| > 6$, we usually find that $h_\beta(\cdot)$ is a function similar to a Gaussian (but not perfectly symmetric) with a global maximum in a neighborhood of u.

Asymptotic behavior of $g_\beta(\cdot)$ for large values of u

When the parameter u is large and positive, the function $g_\beta(\cdot)$ is similar to a Gaussian (but not perfectly symmetric) with an argument of the maximum that asymptotically converges to u. Thus, we can conclude that the argument of the maximum of $g_\beta(\cdot)$ subtracted by u asymptotically tends to 0 at positive infinity, i.e.:

$$\lim_{u\to+\infty}\left[\arg\max_{\mathtt{t}\in\mathbb{R}}\left[g_{(u,\Theta,a)}(\mathtt{t})\right](u)-u\right]=0 \tag{2.8}$$

Whereas for large negative values of u we observe a function similar to a negative Gaussian (but not perfectly symmetric), where $g_\beta(\cdot) < 0$ has a minimum argument that asymptotically converges to u at negative infinity. Thus, we can conclude that the argument of the minimum of $g_\beta(\cdot)$ subtracted by u tends to 0 at negative infinity, i.e.:

$$\lim_{u\to-\infty}\left[\arg\min_{\mathtt{t}\in\mathbb{R}}\left[g_{(u,\Theta,a)}(\mathtt{t})\right](u)-u\right]=0 \tag{2.9}$$

Qualitative analysis of the behavior of $h_\beta(\cdot)$ and $g_\beta(\cdot)$

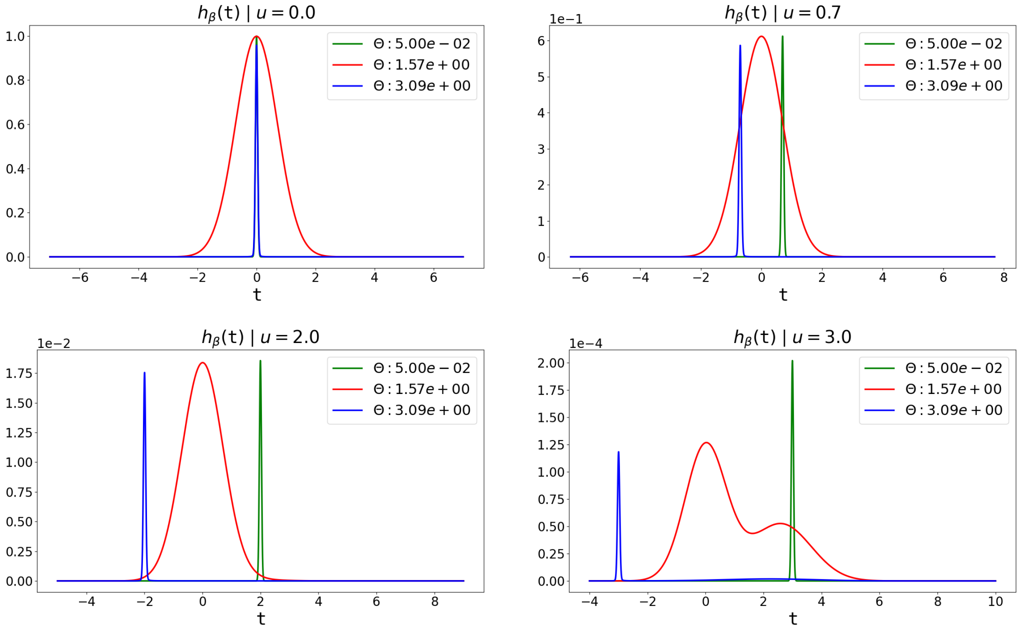

Figure 2.1. *Three examples of* $h_\beta(\cdot)$ *for four different values* $u \in \{0, 0.7, 2, 3\}$ *and at three different values for* $\Theta \in \{0.05, \pi/2, 3\}$.

In figures 2.1 and 2.2 we can see that the behavior of the two functions is strongly influenced by the values of the parameters Θ and u, while the parameter a has less influence. In general, we see that if the parameter u is large, they behave similarly to a Gaussian with an absolute maximum or minimum close to u, regardless of the value of Θ. When the parameter u is smaller than about 8, the behavior of the two functions is strongly influenced by Θ, and we can distinguish three main regimes:

- If Θ is close to π we observe sharp peaks induced by the Voigt profile or the associated dispersion function that tend to be close to $\mathtt{t} \sim -u$.
- If Θ is close to 0 we observe a sharp peak close to $\mathtt{t} \sim u$ induced by the Gaussian.
- For intermediate values, sufficiently far from 0 and π the two functions show smooth profiles.

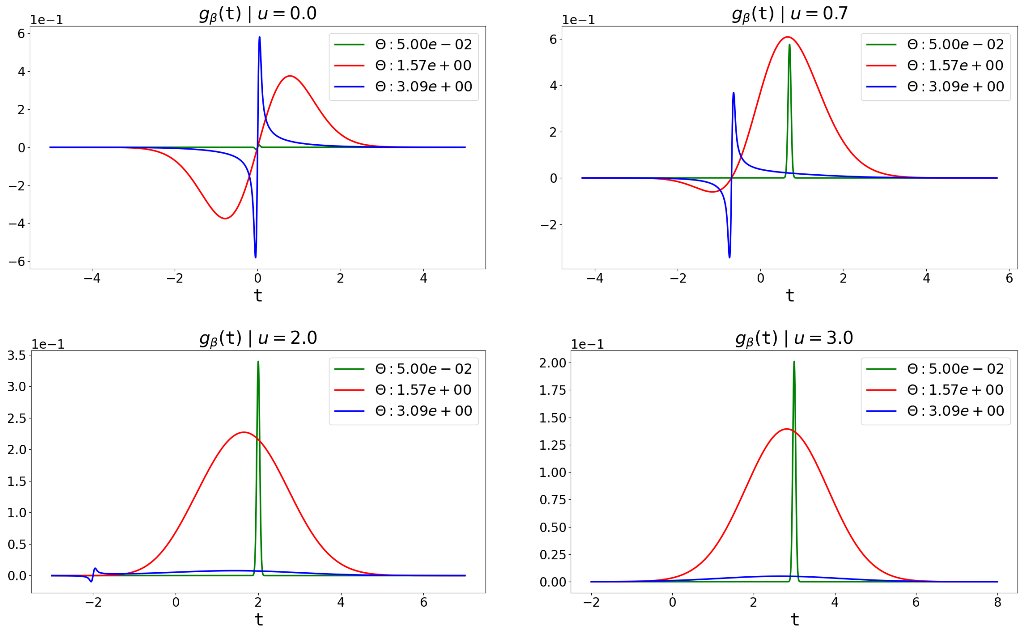

Figure 2.2. *Same as Fig. 2.1 but for the $g_\beta(\cdot)$ function.*

Quadrature methods (for the integration over u') based on the distinction between different regimes of the integrands with respect to the subdomains of u have already been proposed for the angle-averaged case (e.g. Gouttebroze, 1986).

Observing that the two functions converge to zero at infinity (see (2.6)), from their definition we can conclude that both show an exponential decay. Therefore, we can say that there are bounded intervals $[b,c]$ and $[d,e]$ within which the functions h_β and g_β, respectively, are relevant. Consequently, the integrals over $\mathbb{R}$ of $h_\beta(\cdot)$ and $g_\beta(\cdot)$ can be approximated with bounded integrals with a minor error, i.e.:

$$\begin{aligned} \tilde{I}_h(u,\Theta,a) &= \int_b^c h_\beta(\mathtt{t})\,\mathrm{d}\mathtt{t} \approx \int_{\mathbb{R}} h_\beta(\mathtt{t})\,\mathrm{d}\mathtt{t} \\ \tilde{I}_g(u,\Theta,a) &= \int_d^e g_\beta(\mathtt{t})\mathrm{d}\mathtt{t} \approx \int_{\mathbb{R}} g_\beta(\mathtt{t})\mathrm{d}\mathtt{t}. \end{aligned} \tag{2.10}$$

Then, it is possible to define a quadrature rule on a bounded integration interval such that the integration error induced by this simplification is small.

2.4 Limit case $\Theta = \pi$

As we have already mentioned, in the case of $\Theta = \pi$ the functions $h_\beta(\cdot)$ and $g_\beta(\cdot)$ converge to two different analytical forms:

$$\begin{aligned} &h^\pi_\beta : \mathbb{R} \longrightarrow \mathbb{R} \\ &h^\pi_\beta(\mathtt{t}) = \exp\left[-\left(\frac{u-\mathtt{t}}{2}\right)^2\right]\frac{a}{\pi(\mathtt{t}^2+a^2)} \end{aligned} \tag{2.11}$$

$$\begin{aligned} &g^\pi_\beta : \mathbb{R} \longrightarrow \mathbb{R} \\ &g^\pi_\beta(\mathtt{t}) = \exp\left[-\left(\frac{u-\mathtt{t}}{2}\right)^2\right]\frac{\mathtt{t}}{\pi(\mathtt{t}^2+a^2)} \end{aligned} \tag{2.12}$$

From Fig. (2.3) we can see that in this particular case, the two functions show profiles with sharp and smooth features with different relative amplitudes. This behavior makes them difficult to be integrated numerically, and the use of an adaptive or dedicated method is therefore recommended.

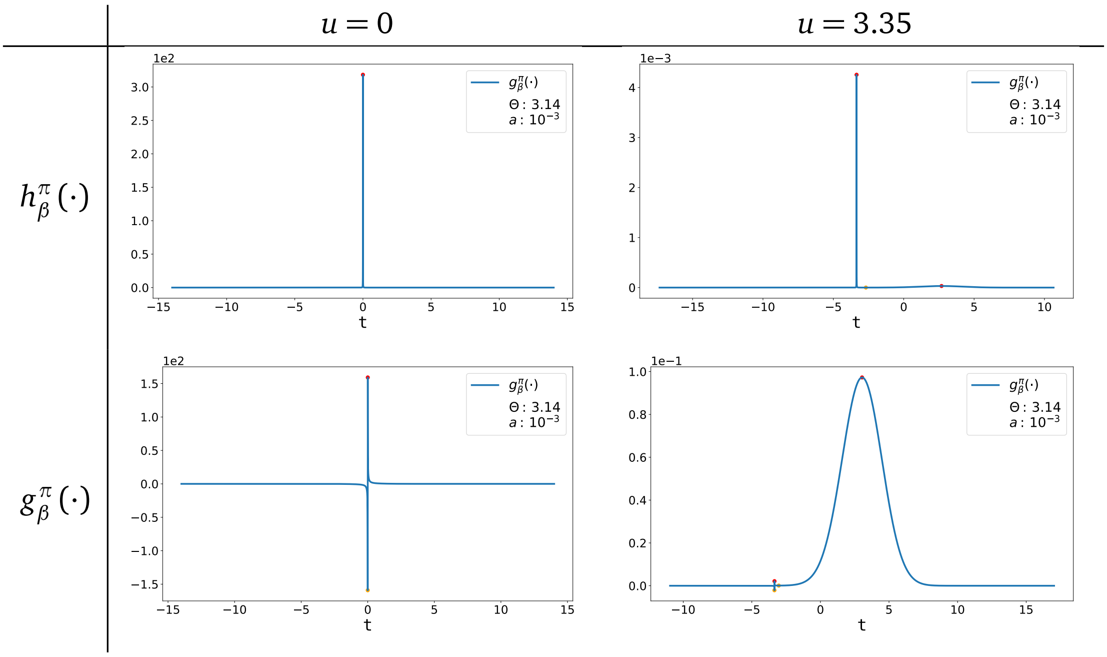

Figure 2.3. *Examples of typical profiles of $h^\pi_\beta(\cdot)$ and $g^\pi_\beta(\cdot)$. The local maxima and minima are marked in red and yellow, respectively. The scale parameter of the sharp peaks is a. In the case of $h^\pi_\beta(\cdot)$, the full width at half maximum, of the sharp peak, is approximately $2a$.*

2.5 Definition of the functions in the presence of a magnetic field

In this section we analyze how the functions $h_\beta\left(\cdot\right)$ and $g_\beta\left(\cdot\right)$ generalize in the presence of a magnetic field. The influence of the magnetic field on the redistribution matrix can be described by two scalars, S_1 and S_2, which depend on the intensity of the magnetic field and the magnetic quantum numbers. Including these two additional parameters, the functions $h_\beta(\cdot)$ and $g_\beta(\cdot)$ generalize to $\tilde{h}_{(u,\Theta,a,S_1,S_2)}(\cdot)$ and $\tilde{g}_{(u,\Theta,a,S_1,S_2)}(\cdot)$, which correspond to the previously defined functions with the parameter u shifted by the quantity S_1 and the argument ω shifted by S_2:

$$
\begin{aligned}
&\tilde{h}_{(u,\Theta,a,S_1,S_2)} : \mathbb{R} \longrightarrow \mathbb{R} \\
&\tilde{h}_{(u,\Theta,a,S_1,S_2)}\left(\mathtt{t}\right) = h_{(u+S_1,\Theta,a)}(\mathtt{t} + S_2) \\
&\tilde{g}_{(u,\Theta,a,S_1,S_2)} : \mathbb{R} \longrightarrow \mathbb{R} \\
&\tilde{g}_{(u,\Theta,a,S_1,S_2)}\left(\mathtt{t}\right) = g_{(u+S_1,\Theta,a)}(\mathtt{t} + S_2)
\end{aligned}
\tag{2.13}
$$

where

$$
\begin{aligned}
S_1 &= \frac{S_x + S_y}{2} \\
S_2 &= \frac{S_x - S_y}{2},
\end{aligned}
\tag{2.14}
$$

and S_x is the total magnetic shift appearing in the argument of the Faddeeva function (i.e. $u_{M'_u M_\ell} + u_{M'_u M'_\ell}$ or $u_{M_u M_\ell} + u_{M_u M'_\ell}$) in the definition of $\mathcal{R}^{\mathrm{II},KK'}_{Q}$ (see Equation (1.13) on page 15), while S_y is the magnetic shift appearing in the argument of the exponential (i.e. $u_{M_\ell M'_\ell}$).

Then, all the properties of $h_\beta\left(\cdot\right)$ and $g_\beta\left(\cdot\right)$ discussed in the previous section also hold in the presence of magnetic fields. The relations in Eq. (2.13) are also valid for the limit case where $\Theta = \pi$, given by Eqs. (2.11) and (2.12).

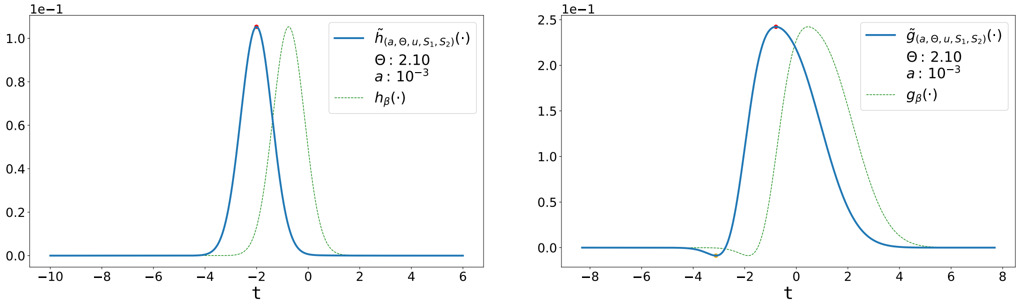

Figure 2.4. *Comparison between the case with a magnetic field (solid line) and the case in the absence of a magnetic field (dotted line). For the case with a magnetic field, we consider an illustrative example with* $u = 1.5$*,* $S_1 = 1.5$ *and* $S_2 = -1$*, which allows observing the shift of the two functions.*

2.6 Quadrature of $h_\beta(\cdot)$ and $g_\beta(\cdot)$

In Equation (2.10) we showed that unbounded integrals over $\mathbb{R}$ of the functions $h_\beta(\cdot)$ and $g_\beta(\cdot)$ can be approximated by bounded integrals with a small error, but we did not provide a practical method for determining optimal integration intervals. In Section 2.3 we studied the properties of the functions $h_\beta(\cdot)$ and $g_\beta(\cdot)$. Furthermore, in Section 3.3 we will show that the Gaussian quadrature rules are accurate methods for the numerical integration of these two functions. Thus, we have all the mathematical tools necessary to define efficient numerical procedures for performing the quadrature of the functions $h_\beta(\cdot)$ and $g_\beta(\cdot)$.

2.6.1 Introduction

In this section, we present a method for constructing a sequence of integration intervals to perform the quadrature of the functions $h_\beta(\cdot)$ and $g_\beta(\cdot)$ over the incoming reduced frequencies, based on the prior knowledge acquired in Section 2.3.

To implement an efficient quadrature method on $\mathtt{t}$ (i.e., the incoming reduced frequencies), it is necessary to define optimal quadrature intervals by taking into account the behavior of the integrands as a function of the parameters in $\beta = (a, \Theta, u)$. After analyzing the $h_\beta(\cdot)$ and $g_\beta(\cdot)$ functions, we found that it is possible to define such a procedure in cases where Θ is not too close to π. However, in cases where $\Theta = \pi$, it is necessary to use a special procedure or an adaptive algorithm based on the Gauss-Kronrod quadrature rules (see also

Appendix B.3).

In our strategy, the main integration interval $[a,b]$ is divided into M adjacent subintervals $[a_i,b_i]_{i=1}^{M}$ that are tailored to the local behavior of the integrands. Therefore, in the case of $h_\beta(\cdot)$, the main integral can be written as:

$$\int_a^b h_\beta(\mathtt{t})\,\mathtt{dt} = \sum_{i=1}^{M} \int_{a_i}^{b_i} h_\beta(\mathtt{t})\,\mathtt{dt}. \tag{2.15}$$

A similar equation can be written for $g_\beta(\cdot)$.

A common approach used to define the integration intervals of redistribution functions, which has been proposed in the past, is the use of the concept of "regimes" (e.g., Gouttebroze, 1986). In this approach, the domain of the outgoing reduced frequencies (i.e., the parameter u in β, see eq. (2.4)) is divided into subdomains (i.e., regimes), where each regime is a priori associated with a specific quadrature approach. However, methods based on the concept of regimes have the disadvantage of being inflexible as they rigidly define the boundaries between subdomains.

Regimes
The following regimes, in which the redistribution functions show different behaviors, are typically defined in the literature:

- **Core** the subdomain where $0 \le |u| \lesssim 2$.
- **Near wings** the subdomain where $2 \lesssim |u| \lesssim 8$.
- **Far wings** the subdomain where $|u| \gtrsim 8$.

In this section, we propose a new approach to overcome the limitations of regime-based approaches. This method returns a set of adjacent quadrature intervals as a function of the parameters in β (eq. 2.5), each of which is associated with a difficulty factor: $\{[a_i,b_i],D_i\}_{i=1}^{M}$. The difficulty factor, D_i, is a positive scalar that is used to determine the difficulty of the quadrature, which leads to the order of the Gaussian quadrature.

Introduction to the integration of the *Gaussian function* and of the normal distribution

The generic analytical solution of the integral of the Gaussian function is as follows:

$$\int_{-\infty}^{\infty} a\, e^{-\left(\frac{x-b}{c}\right)^2} \mathrm{d}x = a\, |c|\, \sqrt{\pi} \tag{2.16}$$

In the case of the normal distribution $\mathcal{N}\left(x;\mu,\sigma^2\right)$, we observe that the value of the bounded integral

$$I(k) = \int_{\mu-k\sigma}^{\mu+k\sigma} \mathcal{N}\left(x;\mu,\sigma^2\right) \mathrm{d}x \tag{2.17}$$

depends only on the width factor k and is independent of the values of μ and σ. Its value is given by:

$$I(k) = \operatorname{erf}\left(\frac{k}{\sqrt{2}}\right), \tag{2.18}$$

where $\operatorname{erf}(\cdot)$ is the error function.

Then, given a normal distribution, we can calculate the integral value independently of the values of σ and μ. Or, conversely, we can calculate the interval of the integral given its result.

The following table shows some results of the integral (2.17):

k	$I(k)$
1	0.682689492137086
2	0.954499736103642
3	0.997300203936740
4	0.999936657516334
5	0.999999426696856
6	0.999999998026825

The normalized Voigt profile

The normalized Voigt profile, denoted by $\mathcal{V}_{(\sigma,\gamma)}(x)$, is a convolution of a normal distribution $\mathcal{N}\left(x;0,\sigma^2\right)$ with a Lorentzian distribution $\mathcal{L}_{(\gamma)}(x)$ with a scale parameter γ:

$$\begin{aligned}\mathcal{V}_{(\sigma,\gamma)}(x) &= \left[\mathcal{N}\left(0,\sigma^2\right) * \mathcal{L}_{(\gamma)}\right](x)\\ &= \int_{\mathbb{R}} \mathcal{N}\left(t;\, 0,\sigma^2\right)\mathcal{L}_{(\gamma)}(x-t)\,\mathrm{d}t.\end{aligned} \tag{2.19}$$

The normalized Voigt profile can be expressed in terms of the real part of the Faddeeva function (see also Equation (1.16) on page 17):

$$\mathcal{V}_{(\sigma,\gamma)}(x) = \frac{\operatorname{Re}\left(W\left(\frac{x+i\gamma}{\sigma\sqrt{2}}\right)\right)}{\sigma\sqrt{2\pi}}. \tag{2.20}$$

The use of the normalized Voigt profile is convenient in many applications due to its dependence on the known parameter σ.

2.6.2 Quadrature intervals

First, we recall that the functions $h_\beta(\cdot)$ and $g_\beta(\cdot)$ (see eqs. (2.2), and (2.3)) are the product of two components: a Gaussian and a Voigt profile for $h_\beta(\cdot)$, and a Gaussian and an associated dispersion profile for $g_\beta(\cdot)$. The behavior of these functions is controlled by the parameters $\beta = (u,\Theta,a)$ (as shown in Sect. 2.3).

In the case where a magnetic field is present, the function $\tilde{h}_{(\beta,S_x,S_y)}(\cdot)$ (as described in Section 2.5 on page 44) is the product of two components:

$$\begin{aligned}&\tilde{h}_{(\beta,S_x,S_y)} : \mathbb{R}^3 \longrightarrow \mathbb{R}_{>0}\\ &\tilde{h}_{(\beta,S_x,S_y)}(\mathtt{t}) = \underbrace{\exp\left[-\left(\frac{u-\mathtt{t}+S_y}{2\sin(\Theta/2)}\right)^2\right]}_{G_\beta(\mathtt{t})} \underbrace{V\left(\frac{a}{\cos(\Theta/2)}, \frac{u+\mathtt{t}+S_x}{2\cos(\Theta/2)}\right)}_{V_\beta(\mathtt{t})}.\end{aligned} \tag{2.21}$$

We now define the integrals over $\mathbb{R}$ of the two factors of the previous equation as follows:

$$\begin{aligned} I_{\mathbb{R}}^{G} &= \int_{\mathbb{R}} G_\beta(x)\,dx,\\ I_{\mathbb{R}}^{V} &= \int_{\mathbb{R}} V_\beta(x)\,dx.\end{aligned} \tag{2.22}$$

In this section, quantities associated with the Gaussian component are denoted by the superscript G, while those related with the Voigt profile are denoted by the superscript V. Since the purpose is to define the integration intervals, it is not necessary to study the function $\tilde{g}_{(\beta,S_x,S_y)}(\cdot)$ (as shown in Equation (2.3) on page 39), since the resulting intervals, derived from the properties of $\tilde{h}_{(\beta,S_x,S_y)}(\cdot)$, are practically valid in both cases.

In the following paragraphs, we first examine the error introduced by the use of bounded integration intervals in the approximation of $I^G_{\mathbb{R}}$ and $I^V_{\mathbb{R}}$. Then we present a procedure that, given the intervals where $G_\beta(\cdot)$ and $V_\beta(\cdot)$ are relevant, produces a sequence of intervals and difficulty factors that can be used to construct a reliable quadrature rule.

Interval of the Gaussian $G_\beta(\cdot)$

To define the bounded integration interval for $G_\beta(\cdot)$ given in the equation (2.21), we must determine its standard deviation σ_G and its mean value μ_G. Which are equal to

$$\begin{aligned} \sigma_G &= \sqrt{2}\sin(\Theta/2) \\ \mu_G &= u + S_y. \end{aligned} \tag{2.23}$$

Then we can define the interval for $k_G > 0$ as

$$\mathrm{itrv}_G = \left[\mu_G - k_G\sigma_G, \quad \mu_G + k_G\sigma_G\right]. \tag{2.24}$$

The relative approximation error of the integral $G_\beta(\cdot)$ over the interval itrv_G can be calculated analytically (see also equation (2.22)):

$$\left|\frac{\int_{\mathrm{itrv}_G} G_\beta(x)\,\mathrm{d}x - I^G_{\mathbb{R}}}{I^G_{\mathbb{R}}}\right| = 1 - \mathrm{erf}\left(\frac{k_G}{\sqrt{2}}\right). \tag{2.25}$$

Thus, the integration interval for the Gaussian $G_\beta(\cdot)$ can be easily defined by specifying a priori an approximation error. It should be noted that for large values of u the behavior of $\tilde{h}_{(\beta,S_x,S_y)}(\cdot)$ is dominated by $G_\beta(\cdot)$ (see section 2.3), so in these specific cases it is sufficient to use itrv_G as the unique integration interval.

Interval of the Voigt profile $V_\beta(\cdot)$

By definition the Voigt profile (see also eq. (1.16)) is a convolution between a Gaussian function with a standard deviation of σ_V and a mean value of μ_V, whose values can be easily derived from the definition of $\tilde{h}_{(\beta,S_x,S_y)}(\cdot)$, i.e.:

$$\begin{aligned} \sigma_V &= \sqrt{2}\cos(\Theta/2) \\ \mu_V &= -(u + S_x)\,, \end{aligned} \tag{2.26}$$

and a Lorentzian distribution with a scale parameter of γ_V, that is:

$$\gamma_V = \frac{a}{\cos(\Theta/2)}. \tag{2.27}$$

Since a convolution between an arbitrary function $f(\cdot)$ and a Dirac delta function is equal to $f(\cdot)$, it is easy to conclude that if $\gamma_V = 0$, the Voigt profile is equivalent to a Gaussian (because in the case of $\gamma_V = 0$, the Lorentzian converges to a Dirac delta function). Therefore, if $0 < \gamma_V \ll \sigma_V$, the Voigt profile $V_\beta(\mathtt{t})$ is close to a Gaussian with standard deviation: $\sigma_V = \sqrt{2}\cos(\Theta/2)$.

In solar atmospheric models the values of a are typically less than 1×10^{-2} hence in most cases where Θ is not close to the limit case of $\Theta = \pi$, we can assume that $V_\beta(\cdot)$ is generally similar to a Gaussian. This allows us to state that the sub-domain where V_β is relevant is well approximated by the interval based on σ_V and μ_V, i.e.,

$$\mathrm{itrv}_V = \left[\mu_V - k_V\sigma_V, \quad \mu_V + k_V\sigma_V\right]. \tag{2.28}$$

On the other hand, as Θ approaches π, the behavior of the integrand approaches that of the limit case for $\Theta = \pi$ (see also Section 2.4). Therefore, the method described above is not applicable and another strategy must be used.

The relative error of the bounded integral of the Voigt profile over itrv_V is defined as

$$\mathrm{error}_V = \left| \frac{\int_{\mathrm{itrv}_V} V_\beta(x)\,dx - I^V_{\mathbb{R}}}{I^V_{\mathbb{R}}} \right|. \tag{2.29}$$

Note that if $\gamma_V = 0$, $V_\beta(\cdot)$ is equivalent to a Gaussian, and $\mathrm{error}_V = 1 - \mathrm{erf}\left(\frac{k_V}{\sqrt{2}}\right)$. Therefore, the error of the bounded integral grows as a function of γ_V, and we can state that:

$$1 - \mathrm{erf}\left(\frac{k_V}{\sqrt{2}}\right) \le \mathrm{error}_V \le 1 - \mathrm{erf}\left(\frac{k_V}{\sqrt{2}}\right) + \Delta(\gamma_V), \tag{2.30}$$

where $\Delta(\cdot)$ grows monotonically with γ_V and $\Delta(0) = 0$. For this motivation, it is necessary to increase the integration interval and the difficulty factor D proportionally with γ_V.

2.6.3 Note on the limit case for $\Theta = \pi$

The analytical form of the limit cases $h^\pi_\beta(\cdot)$ and $g^\pi_\beta(\cdot)$ defined in equations (2.11), and (2.12) does not involve the Faddeeva function, and it is easy to see that their derivatives are:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{dt}}h_\beta^\pi(\mathtt{t}) &= e^{-\frac{1}{4}(u-\mathtt{t})^2}\frac{\left(a^3(u-\mathtt{t})-a\mathtt{t}\left(-u\mathtt{t}+\mathtt{t}^2+4\right)\right)}{2\pi\left(a^2+\mathtt{t}^2\right)^2}\\ \frac{\mathrm{d}}{\mathrm{dt}}g_\beta^\pi(\mathtt{t}) &= e^{-\frac{1}{4}(u-\mathtt{t})^2}\frac{\left(a^2\left(u\mathtt{t}-\mathtt{t}^2+2\right)-\mathtt{t}^2\left(-u\mathtt{t}+\mathtt{t}^2+2\right)\right)}{2\pi\left(a^2+\mathtt{t}^2\right)^2}.\end{aligned}$$

Whit some algebra it is possible to show that their zeros can be expressed through polynomials. Despite the apparent complexity of the resulting expressions (which we do not report here), it is possible to compute exactly the stationary points of both expressions only by evaluating some polynomials and then we can define reliable integration intervals and quadrature grids. Given the intervals, the main problem in the quadrature of this limit case is the fact that the resulting grid has a large number of nodes whose density is highly variable. In fact, the construction of this quadrature grid requires a dedicated procedure.

2.6.4 Description of the procedure

In this section, we define the general procedure (defined in the Algorithm 4) used to generate reliable quadrature intervals given parameters u, a, and Θ. This procedure is a set of empirical rules designed to construct a sequence of integration intervals by comparing the relative widths and mean values of the Gaussian and Voigt profiles defined in the equation (2.21).

The width of the integration interval for the Gaussian $G_\beta(\cdot)$ is given by equation (2.24). In the algorithm 4 it is controlled by the variable k_G (line 1). For the Voigt profile, we can assume that in the range of values of a that normally appear in atmospheric models and for values of $\Theta < 3.1$, the relation $\gamma_V \ll \sigma_V$ holds in most cases, or more generally, the value of γ_V is not critical (see also Section 2.6.2). The criticality of γ_V in the algorithm 4 can be checked through the ratio $r = \frac{\gamma_V}{\sigma_V}$ (line 3). Thus, if r is small, we can assume that σ_V (see Eq. (2.26)) is a good approximation of the width of $V(\cdot)$ and we can use the interval in Equation (2.28), otherwise an alternative procedure must be used. The width of the integration interval induced by $V(\cdot)$ is controlled by the variable k_V (see line 2 in the algorithm 4).

The rules of the algorithm are based on the relative positions of the optimal integration intervals for the Gaussian function (line 7) and the Voigt profile (line 8). It should be noted that the Gaussian function decays faster than the Voigt profile. Therefore, if the two intervals do not overlap, the contribution of the Voigt profile is numerically canceled out by the decay of the Gaussian. On the other hand, the decay of the Voigt profile, and especially the associated dispersion profile, is slower and insufficient to numerically cancel out the contribution of the Gaussian (note that these two phenomena occur in the far wings). This fact explains the asymptotic behavior discussed in Section 2.3 on page 40.

Descriptions of the rules

First rule (line 12): when the Gaussian interval is a subset of the Voigt profile interval i.e.

$$\mathrm{itrv}_G \subset \mathrm{itrv}_V\,,$$

we use the Gaussian interval as the only integration interval. This usually occurs in the *near wings* and in the *core* when $\Theta \lesssim 1.5$. This rule is justified by the fact that the decay of the Gaussian is fast enough to cancel the contribution of the Voigt profile.

Second rule (line 15): if the two intervals do not overlap i.e.

$$\mathrm{itrv}_G \cap \mathrm{itrv}_V = \emptyset\,,$$

we use only the Gaussian interval as the unique integration interval. This usually occurs in the far wings. As described above, if the two intervals do not overlap, the decay of $G_\beta(\cdot)$ numerically cancels the contribution of the Voigt profile. Therefore, we can only use the Gaussian interval $G_\beta(\cdot)$.

Third rule (line 18): if the interval of the Voigt profile is a subset of the Gaussian interval i.e.

$$\mathrm{itrv}_V \subset \mathrm{itrv}_G\,,$$

we set the left and right limits as those of the Gaussian because the decay of the Voigt profile is slow and does not cancel out the contribution of the Gaussian. Therefore, in these cases, it is necessary to divide the integration interval into *three subintervals* with different difficulty factors. This is because the central subinterval, which is dominated by the Voigt profile, requires a higher order of quadrature, while the left and right intervals, which show a monotonic decay and are dominated by the Gaussian, require a lower order of quadrature. This typically occurs in the core and near the wings, where $\Theta \gtrsim 1.5$. Note that this rule distinguishes between cases where the mean of the Gaussian is inside in the interval of the Voigt profile

$$\mu_G \in \mathrm{itrv}_V$$

and cases where it is not. This last condition allows us to reduce the number of quadrature nodes without a loss of accuracy.

Fourth rule (line 28): in addition, we consider cases where the two intervals partially overlap.

$$(\mathrm{itrv}_G \cap \mathrm{itrv}_V \neq \emptyset) \quad \wedge \quad (\mathrm{itrv}_G \cap \mathrm{itrv}_V \neq \{\mathrm{itrv}_G \,|\, \mathrm{itrv}_V\})$$

For these cases, we create a sub-interval corresponding to itrv_V, and another one from the left (right) boundary of itrv_V to the left (right) limit of itrv_G.

In our experiments, the factor of difficulty D is the ratio between the width of the various sub-intervals and the σ associated with the functions corresponding to that subintervals.

Algorithm 3: Compute the difficulties of the intervals.

```
Function ComputeDifficulties (Intervals[:], $\sigma$[:]):
    Output: $D = []$
    for $i = 1 :$ Size($\sigma$[:]) do
        $D[i] = \frac{|\mathrm{Intervals}[i+1] - \mathrm{Intervals}[i]|}{\sigma[i]}$
    return $D$
```

Algorithm 4: A priori definition of the integrations intervals

```
Input: $\{\sigma_G, \mu_G, \sigma_V, \mu_V, \gamma_V\}$

$k_G \geq 5$
$k_V \geq 5$
$r := \frac{\gamma_V}{\sigma_V}$
if $r$ is too large then
    Consider to use an adaptive quadrature.
    return
$\mathrm{itrv}_G := [\mu_G - k_G\sigma_G,\quad \mu_G + k_G\sigma_G]$
$\mathrm{itrv}_V := [\mu_V - k_V\sigma_V,\quad \mu_V + k_V\sigma_V]$
$\delta = 1/2$
$B_{left} := \delta\,(\mathrm{itrv}_G[1] + \mathrm{itrv}_V[1])$
$B_{right} := \delta\,(\mathrm{itrv}_G[2] + \mathrm{itrv}_V[2])$
if $itrv_G \subset itrv_V$ then
    Intervals := $\mathrm{itrv}_G$
    $D$ := ComputeDifficulties (Intervals, $[\sigma_G]$)
else if $itrv_G \cap itrv_V == \emptyset$ then
    Intervals := $\mathrm{itrv}_G$
    $D$ := ComputeDifficulties (Intervals, $[\sigma_G]$)
else if $itrv_V \subset itrv_G$ then
    if $\mu_G \in itrv_V$ then
        Intervals := $[B_{left},\quad \mathrm{itrv}_V[1],\quad \mathrm{itrv}_V[2],\quad B_{right}]$
        $D$ := ComputeDifficulties (Intervals, $[\sigma_G,\quad \sigma_V,\quad \sigma_G]$)
    else
        if $\mu_G > \mu_V$ then
            Intervals := $[B_{left},\quad \mathrm{itrv}_V[1],\quad \mathrm{itrv}_V[2],\quad \mathrm{itrv}_G[2]]$
        else
            Intervals := $[\mathrm{itrv}_G[1],\quad \mathrm{itrv}_V[1],\quad \mathrm{itrv}_V[2],\quad B_{right}]$
        $D$ := ComputeDifficulties (Intervals, $[\sigma_G,\quad \sigma_V,\quad \sigma_G]$)
else if $itrv_G \cap itrv_V \neq \emptyset$ then
    if $\mu_G > \mu_V$ then
        Intervals := $[\mathrm{itrv}_V[1],\quad \mathrm{itrv}_V[2],\quad \mathrm{itrv}_G[2]]$
        $D$ := ComputeDifficulties (Intervals, $[\sigma_V,\quad \sigma_G]$)
    else
        Intervals := $[\mathrm{itrv}_G[1],\quad \mathrm{itrv}_V[1],\quad \mathrm{itrv}_V[2]]$
        $D$ := ComputeDifficulties (Intervals, $[\sigma_G,\quad \sigma_V]$)
Output: {Intervals, $D$}
```

Finally, given a list of intervals and a list of "difficulty" D returned by the algorithm 4, the integrals of $\tilde{h}_{(\beta,s_x,s_y)}(\cdot)$ and $\tilde{g}_{(\beta,s_x,s_y)}(\cdot)$ can be calculated using the algorithm 5.

Algorithm 5: Computation of the Integral

```
   Input: Intervals[:], $D[:]$, $f(\cdot)$
   Output: $I = 0$
1  for $i = 1 : \textbf{\textit{Size}}(D[:])$ do
2      $N \propto D[i]$
3      $\{x_j, w_j\}_{j=1}^{N} :=$
        MakeGaussianQuadrature($N$, $Intervals[i]$, $Intervals[i+1]$)
4      $I := I + \sum_{j=1}^{N} w_j f(x_j)$
5  return $I$
```

In the algorithm 5, the number of nodes N in a given interval i ((line 2)) is obtained by multiplying $D[i]$ by a scalar (in practice between $5\dots8$).

2.6.5 Fine tuning

- We recommend increasing the proportionality factor for N in the algorithm 5 at line 2 in the *core* and *near wings*.

- In the *far wings*, the functions have a shape very similar to a Gaussian. Thus, high accuracy can be achieved with a number of quadrature nodes $N = 40\dots60$.

- In a practical application, we recommend always returning an array of flags indicating whether the sub-intervals are induced by the Gaussian (only) or by the Voigt profile. This helps to optimize the order of quadrature and the building of the grid.

- The quadrature of the associated dispersion profile is usually more difficult than that of the Voigt profile. Therefore, it is recommended to adjust the order of the quadrature accordingly.

2.6.6 Direct approximation of the integral in the very far wings

In Section 2.3, we observed that for large values of the parameter u, $\tilde{h}_{(\beta,s_x,s_y)}(\cdot)$ and $\tilde{g}_{(\beta,s_x,s_y)}(\cdot)$ exhibit a shape close to a Gaussian. We can exploit this property to approximate the integrals of $\tilde{h}_{(\beta,s_x,s_y)}(\cdot)$ and $\tilde{g}_{(\beta,s_x,s_y)}(\cdot)$ in the *very far wings*.

If we use the analytical result of the integral of the Gaussian (2.16), the value of the integration of $G_\beta(\cdot)$ (see Eq. (2.21)) is:

$$\int_{\mathbb{R}} \exp\left[-\left(\frac{u-\mathrm{t}+S_y}{2\sin(\Theta/2)}\right)^2\right]\mathrm{dt} = 2\sqrt{\pi}\,\sin(\Theta/2). \tag{2.31}$$

In the cases of a large value of u (e.g. $|u| \gtrsim 40$) in the interval where $G_\beta(\cdot)$ is relevant (2.24), the values of $V_\beta(\cdot)$ and $L_\beta(\cdot)$ are *quasi constants*. Then, the integral of $\tilde{h}_{(\beta,S_x,S_y)}(\cdot)$ can be approximated by replacing $V_\beta(\cdot)$ with its value at μ_G (i.e. $V_\beta(\mu_G)$ (2.23)):

$$\int_{\mathbb{R}} \tilde{h}_{(\beta,S_x,S_y)}(\mathrm{t})\,\mathrm{dt} \approx \left[\int_{\mathbb{R}} G_\beta(\mathrm{t})\,\mathrm{dt}\right] V_\beta(\mu_G) = 2\sqrt{\pi}\,\sin(\Theta/2)\cdot V_\beta(\mu_G). \tag{2.32}$$

Similarly for the case $\tilde{g}_{(\beta,S_x,S_y)}(\cdot)$ we have:

$$\int_{\mathbb{R}} \tilde{h}_{(\beta,S_x,S_y)}(\mathrm{t})\,\mathrm{dt} \approx \left[\int_{\mathbb{R}} G_\beta(\mathrm{t})\,\mathrm{dt}\right] L_\beta(\mu_G) = 2\sqrt{\pi}\,\sin(\Theta/2)\cdot L_\beta(\mu_G). \tag{2.33}$$

The limit of *very far wings* could reasonably be set at around $|u| \approx 40$, but smaller or larger values can also be used depending on the required accuracy. We must emphasize that if this limit is too small, using this strategy could produce results with significant errors. Observing that in the far wings, the density of nodes in the frequency grid U (see Section 1.8.2) is usually very low, this approximation is not expected to have a significant impact in reducing the total computational time.

2.7 Notes and remarks on the quadrature of R^{II}

By studying the analytical properties of the functions $\tilde{h}_{(\beta,S_x,S_y)}(\cdot)$ and $\tilde{g}_{(\beta,S_x,S_y)}(\cdot)$, we have acquired the prior knowledge necessary to define a specialized quadrature rule, namely, to define a priori a sequence of integration intervals that exploit the behavior of the integrand. We observed that the abscissa of the zero of $\tilde{g}_{(\beta,S_x,S_y)}(\cdot)$ and the approximate abscissas of the local maxima of $\tilde{h}_{(\beta,S_x,S_y)}(\cdot)$ are known (see section. 2.3). We also noted that the behavior of the integrand of $\tilde{h}_{(\beta,S_x,S_y)}(\cdot)$ is the result of the cross-influence of a Gaussian function, whose standard deviation and mean are known, and a Voigt profile, whose mean and approximate width are known. Thanks to this information, we have developed a method capable of generating a dedicated quadrature rule for any triple of the parameters a, u, and Θ. The result of this procedure is a sequence of adjacent integration intervals of two different types: those where the integrand is dominated by the Gaussian (where quadrature is easier) and those dominated by the Voigt profile (where quadrature is more difficult). We then verified that the same rule also provides accurate results for the numerical integration of $\tilde{g}_{(\beta,s_x,s_y)}(\cdot)$. In Fig. 2.5, the red intervals are those where the Voigt profile dominates the integrand, while the green intervals are those where the Gaussian dominates.

To check the validity of our method, we evaluated an estimated relative local quadrature error using the Gauss-Kronrod rule (see Section 3.3.2). Dividing the whole integration interval into N subintervals, we define the relative error for each $j = 1 \ldots N$ as

$$[\tilde{\epsilon}_{\mathrm{rel}}]_j = \log_{10}\left(\frac{[\tilde{\epsilon}]_j}{\max_{i=1\ldots N}([\tilde{\epsilon}]_i)}\right), \tag{2.34}$$

with

$$[\tilde{\epsilon}]_j = \frac{[\tilde{\epsilon}_{\mathrm{GK}}]_j}{\tilde{I}},$$

where $[\tilde{\epsilon}_{\mathrm{GK}}]_j$ is the quadrature error in the sub-interval j calculated with the Gauss-Kronrod rule and $\tilde{I}$ is an approximation of the integral over the whole interval.

In the tests shown in Fig. 2.5, we only report examples with $\Theta = 3$ (i.e. close to the limit case $\Theta = \pi$), because in these cases the functions show behaviors that may be difficult to handle. The approximated integral $\tilde{I}$ is calculated over the whole interval plotted in Fig. 2.5. By observing the estimated relative error (red curve), we see that the integration intervals correctly catch the behavior of the integrands (the finest sub-intervals, namely those in red, associated with the Voigt profile are used where the error is maximal). Using this method as a basis, we were able to develop a procedure capable of constructing quadrature grids for the R^{II} function that proved reliable even when we applied strong magnetic and bulk velocity fields. In practical applications, the effective number of quadrature nodes needed to approximate these integrals can vary from ~ 30 to ~ 250, depending on the values of the parameters in β and the tuning of the main numerical application. The total number of nodes needed to achieve a given accuracy generally grows with the strength of the magnetic field. In the limit case of $\Theta = \pi$ the required number of quadrature nodes can grow significantly, up to ~ 2500 in the case of a moderate magnetic field (e.g. $||\mathbf{B}|| > 50G$).

The accuracy of the whole procedure has been tested by comparing the emission vector (see Eq. (2.1)) obtained with the method described here with that obtained with a Gauss-Kronrod adaptive method (see also App. B.3), which pro-

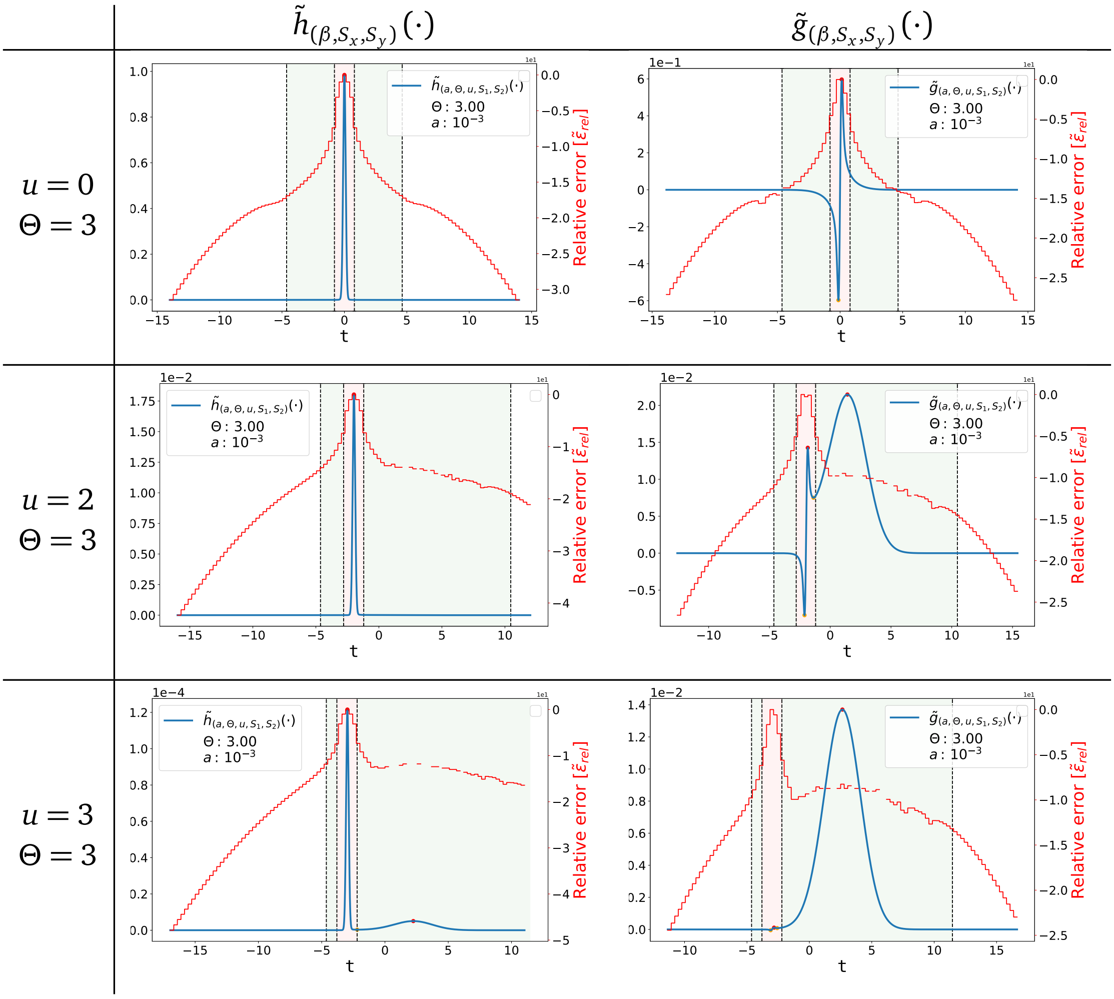

Figure 2.5. *Examples of a priori calculations of some integration intervals. The intervals highlighted in red are those in which the integrand is dominated by the Voigt (or associated dispersion) profile, while those in green are the ones in which the integrand is dominated by the Gaussian. The red curve is the estimated relative error (see Eq. (2.34)). In the lower right plot, we observe that the largest error appears where $\tilde{g}_{(\beta,s_x,s_y)}(\cdot)$ shows a small oscillation. Note that the maximum absolute error observed in these specific tests is less than 10^{-6}.*

vides very accurate results but it is too slow to be used in practical applications (other tests based on the Kirchhoff law can be found in Chapter 4).

Chapter 3

Analysis of quadrature methods

Defining an efficient and accurate method to perform the quadrature of the functions $h_\beta(\cdot)$ and $g_\beta(\cdot)$ by minimizing the number of function evaluations is one of the main tasks necessary to develop an efficient numerical integration of the emission coefficients. Moreover, for the limit case $\Theta = \pi$, it is necessary to provide a stable and robust quadrature method, which may differ from the general one. In this section, we describe three different methods that are suitable candidates for performing the quadrature of these functions.

3.1 A priori adapted quadratures

In this method, we use the prior knowledge of the integrand to divide the main integration interval into a sequence of subintervals, assigning a quadrature order to each according to the predicted behavior of the integrand. This method has the advantage of being simple once the adapted mesh (i.e., the rule for defining the subintervals and assigning the quadrature order) is set. In fact, the numerical integration then consists simply of a weighted sum. However, the method is reliable only if the prior knowledge of the integrand is robust enough (i.e., we know the positions of the stationary points and the regions of the domain where the higher-order derivatives are close to zero). We must also point out that in some cases having robust prior knowledge of the integrand is not a sufficient condition for constructing an efficient and reliable adapted mesh. For example, for the limit cases $h_\beta^\pi(\cdot)$ and $g_\beta^\pi(\cdot)$ we have a good prior knowledge of the two functions, but defining a quadrature mesh and subintervals a priori turns out to be difficult because of the presence of sharp peaks that make it difficult to decide the width and positions of the intervals and the order of the quadrature.

3.2 Adaptive quadratures

An adaptive quadrature is a dynamic algorithm capable of automatically adjusting the sequence of subintervals and, eventually, the order of the quadrature to detect the behavior of the integrand. The advantage of adaptive quadratures is that they do not require a priori knowledge of the integrand. The disadvantage is that they do not allow the implementation of highly effective optimizations (see also Appendix C.1).

3.3 Choice of quadrature methods

3.3.1 Gaussian quadratures

We have chosen Gaussian quadrature (e.g. Deuflhard and Hohmann, 2003; Davis and Rabinowitz, 2007) to be used in algorithms for intensive computation because they offer a high order of accuracy. Indeed a Gaussian quadrature with N nodes is exact for all polynomials $P \in \mathbb{P}_{2N-1}$ (where with $\mathbb{P}_D$ we mean the set of all polynomials with degree $\leq D$). The quadrature algorithm consists of a weighted sum, and therefore it is easy to implement and optimize (see also Appendix B).

3.3.2 Gauss-Kronrod adaptive quadrature

The Gauss-Kronrod method (Kronrod, 1965) is a nested quadrature rule based on an extension of a Gaussian rule of $N+1$ nodes to generate a new quadrature rule that is exact for all polynomials $P \in \mathbb{P}_{3N+1}$ (Notaris, 1993). When combined with a reliable error estimator (e.g., the sharper error estimator (Laurie, 1983; Gonnet, 2012) or the QUADPACK error (Piessens et al., 2012)), it allows the development of adaptive quadrature algorithms that are able to estimate the value of an integral with machine precision (i.e., the relative error is in the order of the machine epsilon) with a small number of function evaluations compared to other adaptive methods (see Appendix B.3 for more details).

We first used a Gauss-Kronrod adaptive method to generate reference results of the emission vector and to study the exact behavior of the R^{II} function. We then used the Gauss-Kronrod adaptive quadrature as one of the main strategies to integrate the redistribution matrix R^{III} in its exact form (see Chapter 6 on page 137).

3.3.3 Kronrod-Patterson extensions

The Gauss-Kronrod rule (Kronrod, 1965) allows the extension of a Gaussian quadrature with only one set of nested nodes. Patterson overcame this limitation by introducing a numerically stable procedure for iterating Kronrod extensions to a tower of nested rules (e.g. Patterson, 1968; Bourquin, 5 04) by progressively increasing the order of the quadrature (see also B.4).

We developed an order-adaptive algorithm using the Kronrod-Patterson extensions. Compared to a classical approach based on Gauss-Kronrod, the performance of this algorithm in terms of efficiency is significantly more dependent on the properties of the integrand. However, we observed that it shows good efficiency in the case of R^{II}, which makes it interesting for our problem.

3.4 Spherical quadrature

We define the *surface integral* of a function $f : \mathbb{R}^3 \longrightarrow \mathbb{R}$ over a sphere S with radius r as:

$$I = \int_S f(\mathbf{x})\, \mathrm{d}S. \tag{3.1}$$

The integral I is generally parameterized in the spherical coordinate system $\mathbf{x} = (r, \theta, \chi)$, where any point in the three-dimensional space is uniquely defined by the inclination $\theta \in [0, \pi]$ (or colatitude) with respect to the vertical axis, the azimuth $\chi \in [0, 2\pi]$, i.e., the angle between the projection of $\mathbf{x}$ on the equatorial plane and a reference horizontal axis, and the distance from the origin $r \in \mathbb{R}_+$ (or radial distance).

If the radius of the sphere is unity, the integral I is equivalent to an *angular integral* over the solid angle subtending all of the space:

$$I = \oint f(\mathbf{\Omega})\, \mathrm{d}\Omega\,, \tag{3.2}$$

where $\mathbf{\Omega} = (\theta, \chi)^{\mathrm{T}}$, and $\mathrm{d}\Omega$ is the infinitesimal solid angle ($\mathrm{d}\Omega = \mathrm{d}S/r^2$). The latter is the integral that we need to solve the physical problem under consideration.

In the scientific literature, numerous methods for performing spherical quadrature have been proposed (e.g. Beentjes, 2015), also with a focus on the specific problem of polarized RT (e.g. Štěpán et al., 2020; Jaume Bestard et al., 2021b). In general, a spherical quadrature rule consists of a set of nodes (a.k.a. a spherical grid), which we express as a set of angular coordinates, and weights

$\mathcal{S} = \{\theta_i, \chi_i, w_i\}_{i=1}^{M}$, such that the integral over a sphere is approximated with a weighted sum, i.e:

$$\int_S f(r, \theta, \chi)\, dS \approx r^2 \sum_{i=1}^{M} w_i f(r, \theta_i, \chi_i), \tag{3.3}$$

The order of accuracy of a spherical quadrature rule is usually expressed in terms of spherical harmonics (SH), so we can say that a rule is exact for all SH $Y_\ell^m \in \mathbb{Y}_D$, where $\mathbb{Y}_D$ means the set of all spherical harmonics up to degree D, and Y_ℓ^m is an SH of degree ℓ and order m.

3.5 Prerequisites of a spherical quadrature for the computation of the emissivity with PRD

In designing a spherical quadrature for calculating the emission coefficients, it is also necessary to consider some constraints imposed by the nature of the problem. **First**, nodes on the equator (i.e., with an inclination $\theta = \pi/2$) should be avoided. In fact, in 1D applications (or more generally with vertical periodic boundary conditions), horizontal beams cannot be treated numerically with standard techniques because they never encounter the boundaries of the spatial domain. The **second** requirement is related to the optimization of the algorithm: the number N_Θ of distinct scattering angles between *unique pairs* of nodes must be significantly lower than the total number of unique pairs of nodes in the spherical grid (i.e.: $N_\Theta \ll ({N_\Omega}^2 - N_\Omega + 2)/2$). If this second property holds, we can implement an optimization strategy that significantly reduces the time required to compute the emission coefficients.

Finally, considering that the integration of the emission coefficients requires a large time-to-solution, it is necessary to use a quadrature rule capable of ensuring sufficient accuracy with a small number of nodes. For the numerical integration of $\varepsilon^{\ell,II}$ given by Eq. (2.1), we should use spherical grids with no more than 150 nodes to achieve an acceptable time-to-solution.

3.6 Sphere mapping

During the study of spherical quadratures, we developed a method that maps a quadrature rule defined on a polygon to its spherical equivalent by using the integral over parametric surfaces (to our knowledge, such a method has never been used before in the context of RT). In particular, we consider the case of mapping a quadrature rule from a reference triangle to an arbitrary spherical triangle.

Let us define a mapping function from a triangle in a reference system of Cartesian coordinates $(u,\, v)$ to a planar triangle in $\mathbb{R}^3$ with vertices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$:

$$\begin{aligned} &\Phi : \mathbb{R}^2 \longrightarrow \mathbb{R}^3 \\ &\Phi(u,v) = \mathbf{A} + u(\mathbf{B}-\mathbf{A}) + v(\mathbf{C}-\mathbf{A}). \end{aligned} \tag{3.4}$$

If we assume that the three vertices lie on a sphere, then the mapping function from the reference triangle to the spherical triangle T_S with vertices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ is given by the projection of $\Phi(\cdot)$ on the sphere, i.e:

$$\begin{aligned} &\Psi : \mathbb{R}^2 \longrightarrow \mathbb{R}^3 \\ &\Psi(u,v) = r\frac{\Phi(u,v)}{\left\|\Phi(u,v)\right\|}. \end{aligned} \tag{3.5}$$

Given a generic quadrature rule $\mathcal{T} = \{u_i, v_i, w_i\}_{i=1}^N$ on a reference triangle in the coordinate system $(u,\, v)$, such that $u_i \in [0,1]$ and $v_i \in [0, 1{-}u_i]$ for all $i = 1 \ldots N$, the integral over the reference triangle T_R is approximated with a weighted sum, i.e.:

$$\int_{T_R} f(\mathbf{u})\, dT_R \approx \sum_{i=1}^N w_i f(u_i, v_i). \tag{3.6}$$

By applying the integration over parametric surfaces and the change of variable, it is possible to prove that the integral over a spherical triangle T_S can be approximated by mapping the quadrature over the reference triangle T_R with:

$$\int_{T_S} f(\mathbf{x})\, dT_S \approx \sum_{i=1}^N f(\Psi(u_i, v_i))\, w_i \left\|\tfrac{\partial\Psi(u_i,v_i)}{\partial u} \times \tfrac{\partial\Psi(u_i,v_i)}{\partial v}\right\|, \tag{3.7}$$

where the operator × denotes the cross product.

This method has the advantage of being a generic procedure that can be used with any spherical polygon and tessellation. On the other hand, it does not guarantee the same order of accuracy as the Cartesian product rule or Lebedev's rule, because the mapping function (3.5) is not linear. The results of some tests of the spherical mapping quadrature method, considering different polygons, are shown in Fig. 3.1. As the figure illustrates, when the number of nodes is in the order of 10^2, the accuracy of the method is still rather low, which makes it unsuitable for the problem under investigation Other, possibly more accurate, methods can be derived from isogeometric analysis (e.g., Zou et al., 2022, and references therein) .

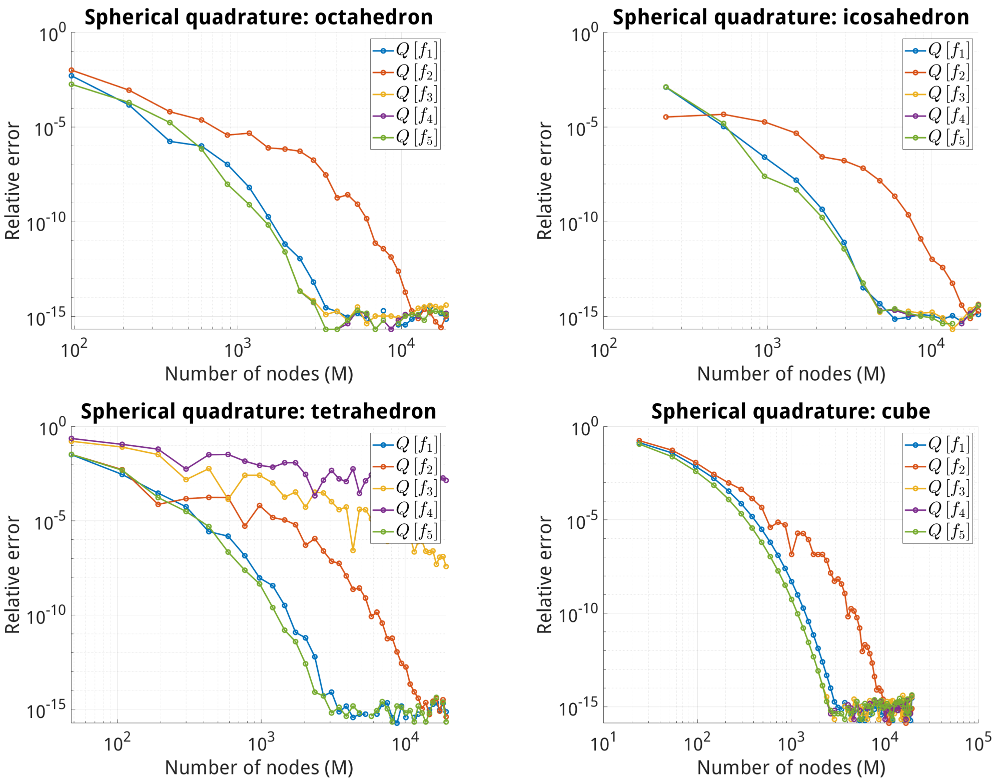

Figure 3.1. *Tests of sphere mapping quadratures. The test functions f_* are the ones proposed by Beentjes (2015). The quadrature rules have been built by mapping 2D Gaussian quadratures on the spherical polygons, which in these tests correspond to the projection of the faces of four Platonic polyhedra. The relative error is calculated with respect to the analytical solutions. We can observe that more than* 1000 *nodes are generally needed to achieve a relative error below* 10^{-5}.

3.7 Comparison of spherical quadratures

Among the spherical quadrature methods described in the scientific literature (e.g. Lebedev's rule, Cartesian product rule, spherical design, etc. (e.g Beentjes, 2015)) we choose the Cartesian product rule and the Lebedev's rule because they are the only ones that fit the prerequisites described in Sec. 3.5.

In numerical radiative transfer, one of the most common methods to perform angular quadrature is the *Cartesian product rule*, since it has a high order of accuracy and easily allows one to avoid nodes on the equator. To exactly integrate all spherical harmonics $Y_\ell^m \in \mathbb{Y}_p$ it is necessary to have a Cartesian product rule with $(p+1)/2$ nodes in azimuth and $(p+1)$ nodes in inclination, corresponding to $(p+1)^2/2$ nodes in total (Beentjes, 2015; McLaren, 1963).

Lebedev's rule has the advantages of ensuring better accuracy (a rule of p nodes integrates exactly all spherical harmonics $Y_\ell^m \in \mathbb{Y}_p$ (Lebedev, 1975)), and of having the nodes uniformly distributed on the spherical surface, but with the disadvantage of having nodes on the equator. On the other hand, the sphere mapping method allows one to set the coordinates of the nodes quasi-freely, but with the problem of low accuracy.

Considering that any spherical quadrature Q is rotational invariant, so that for any rotation operator R and any function f the relation $Q[f] \approx Q[f \circ R]$ holds, the problem of having nodes on the equator in Lebedev's rule can be solved by rotating the spherical grid S. A possible criterion for choosing the rotation is to maximize the minimum distance of the closest node to the equatorial plane. If we define the rotation operator as the composition of the rotation around the Cartesian axes, i.e. $R^{\alpha,\beta,\gamma} = R_x^\alpha \circ R_y^\beta \circ R_z^\gamma$ with Euler angles α, β, and γ, the maximization problem consists in finding a triple of these angles that maximizes the minimal distance, i.e.:

$$\max_{\alpha,\beta,\gamma} \left\{ \min_i \left\{ d_{\text{eq}} \left(R^{\alpha,\beta,\gamma} \mathbf{x}_i \right) \mid \forall \mathbf{x}_i \in S \right\} \right\}, \tag{3.8}$$

where $d_{\text{eq}}(\cdot)$ is the distance from the equatorial plane and $\mathbf{x}_i$ is a node from the spherical grid S. Since this problem is non-linear, we used the genetics algorithms toolbox from MATLAB to solve it.

In conclusion, among the angular quadrature rules that we have analyzed, the most suitable ones for the considered problem are the standard approach based on the Cartesian product rule, or a rotated Lebedev's rule.

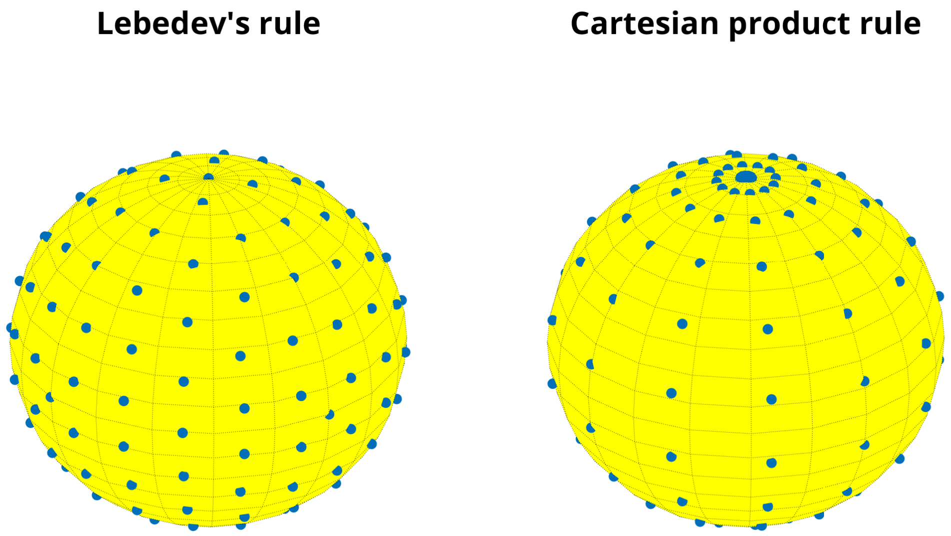

Figure 3.2. *Comparison of Lebedev's rule with* 146 *nodes with respect to a Cartesian product rule with* 144 *nodes.*

Chapter 4

Numerical calculation of the emission coefficient

4.1 Description of the algorithm

In this section, we present an algorithm to perform a reliable and efficient quadrature of the emission coefficients of Eq. (2.1), which we computed on the nodes of the global grid $G = \left\{\mathbf{\Omega}_i, \nu_j\right\}_{i,j=1}^{N_\Omega,N_\nu}$ introduced in Sect. 1.8.2. We designed the algorithm with the primary goal of minimizing the number of evaluations of the Faddeeva function. We achieved this goal by exploiting the properties of the spherical grid and the magnetic shifts. Recalling that the behavior of the integrands is dominated by the redistribution matrix (Eq. (1.5)), when performing the integration over the reduced frequencies of the incoming radiation, we can use the properties of the functions $h_\beta\left(\cdot\right)$ and $g_\beta\left(\cdot\right)$ to define the intervals of integration.

If we use a spherical grid where the number of scattering angles is small (see Section. 3.5), it turns out to be convenient to pre-calculate and store the values of $\mathcal{R}_Q^{\mathrm{II},KK'}$ in a data set and use them in a second phase when performing the effective quadrature of the emission coefficients. It should be noted that the use of the repetitions of the scattering angles allows to reduce the time complexity for the Faddeeva function to $O\left({N_\Omega}^d\,{N_\nu}^2\,N_M\right)$ where $1 < d \ll 2$ and for an ideal spherical grid $d = 1$. We emphasize that this strategy can be applied if the dependence of the redistribution function from the directions is only encoded in the scattering angle. If bulk velocities are present we must therefore calculate the emissivity in the comoving frame, as discussed in Sect. 1.5, taking into account the Doppler shifts in the incoming radiation and in the resulting emissivity as described by Eqs. (1.43) and (1.44).

To define the procedure, we must first calculate the set T, of size N_Θ, of all unique scattering angles Θ (Eq. (1.11)) associated with the chosen spherical grid S, i.e. $T = \{\Theta_i\}_{i=1}^{N_\Theta}$. As previously explained, it is convenient to choose a spherical grid such that N_Θ is significantly less than the total number of unique angular node pairs, i.e., $({N_\Omega}^2 - N_\Omega + 2)/2$, where N_Ω is the number of angular nodes. Observing that, at a given spatial point, the quantity a belonging to the set of parameters $\beta = (u, \Theta, a)$ is constant, we introduce the grid B given by the Cartesian product between the frequency grid U and the set of scattering angles T, i.e. $B = U \times T$. We then store the precalculated values of $\mathcal{R}_Q^{\mathrm{II},KK'}$ in an associative data set with keys $\Theta, u \in B$. Each entry of this data set consists of a sequence of triples, where the first two elements are the nodes and weights of the quadrature rule chosen to evaluate the integral over the frequencies of the incoming radiation, and the third are the precalculated values of $\mathcal{R}_Q^{\mathrm{II},KK'}$, i.e,

$$\mathcal{G}^\beta = \left\{ u'^{\beta}_i, w^\beta_i, \left[\mathcal{R}_Q^{\mathrm{II},KK'}\right]_i^\beta \right\}_{i=1}^{N_\beta},$$

where the superscript indicates the dependence on the parameters in β, and N_β is the number of nodes for a quadrature rule given an outgoing frequency u and a scattering angle Θ. We compute the nodes and weights in $\mathcal{G}^\beta$ using the prior knowledge of the functions $h_\beta(\cdot)$ and $g_\beta(\cdot)$ (see Section 2.3).

Using the data structures described above and a spherical quadrature rule $\mathcal{S} = \left\{\mathbf{\Omega}'_j, w^S_j\right\}_{j=1}^{N_\Omega}$, the numerical calculation of the emission coefficients can be summarized in the following equation.

$$\varepsilon^{\ell,II} : \mathbb{R} \times \mathbb{R}^2 \longrightarrow \mathbb{R}^4$$

$$\varepsilon^{\ell,II}(u, \mathbf{\Omega}) = \frac{k_L}{4\pi} \sum_{j=1}^{N_\Omega} w^S_j \sum_{i=1}^{N_\beta} w^\beta_i \left(\sum_{K,K'=0}^{2} \sum_{Q=-K_{min}}^{K_{min}} \left[\mathcal{R}_Q^{\mathrm{II},KK'}\right]_i^\beta \mathcal{P}_Q^{KK'}\left(\mathbf{\Omega}'_j, \mathbf{\Omega}\right) \right) \mathbf{I}\left(u'^{\beta}_i, \mathbf{\Omega}'_j\right), \tag{4.1}$$

where the direction and reduced frequency of the emitted radiation are elements of the discrete general grid, i.e. $u, \mathbf{\Omega} \in G$ (see also Section 1.8.2), and the quadrature rules in the inner summation use the precalculated values from $\mathcal{G}^\beta$ for each pair of $\Theta, u \in B$.

If $\Theta = 0$, the integral over u' can be solved analytically and a simplified procedure can be used. For the limit case $\Theta = \pi$, constructing a fixed grid presents a number of difficulties, so we developed an ad hoc procedure. Alternatively, we can use Gauss-Kronrod adaptive quadrature, which is capable of evaluating this integral with machine precision (see also Section 1.4.4 on page 18).

For $\Theta \in (0, \pi)$, the use of the Gauss-Kronrod adaptive quadrature is not justified because it is generally possible to achieve the desired accuracy by using the prior knowledge of the functions to define a quadrature mesh based on a Gaussian rule where the total number of nodes is close to the nodes generated by the Gauss-Kronrod adaptive method (Section. 2.7).

In our tests, we also observed that, given a sequence of integration subintervals defined by the method introduced in Sect. 2.7, a suitable method to build an a priori grid is order adaptivity based on Kronrod-Patterson extensions (Sect. 3.3.3). However, this strategy has not been further analyzed in this work.

The use of precalculated values for $\mathcal{R}_Q^{\mathrm{II},KK'}$ and for the scattering phase matrix $\mathcal{P}_Q^{KK'}$ is more convenient than calculating them on the fly, because the function evaluations require more CPU time with respect to the data set access.

The method introduced above can be generalized in the algorithm 6, in which the calculation of the emissivity on all scattering directions and frequencies at one spatial point is performed. The strategy introduced in Algorithm 6 has the advantage of being efficient and fast. On the other hand, generating a database of all $\mathcal{G}^\beta$ at all scattering angles requires in most cases more than 1 Gb of memory (in `float64`). This situation could be problematic, especially when solving the RT problem in 3D atmospheric models. Therefore, we developed an alternative version in which the main loop is based on the scattering angles (Θ) and the $\mathcal{G}^\beta$ are constructed on the fly so that all pairs of directions that share the same Θ are updated in the innermost loops. This second strategy significantly reduces the amount of required memory and is successfully used in the 3D RT-solver (Benedusi et al., 2023).

Algorithm 6: Calculation of the emission coefficients $\varepsilon^{\ell,II}$

```
Input: The incoming radiative filed: $\mathbf{I}$
Given the angular grid find all the distinct scattering angles: $T = \{\Theta_i\}_{i=1}^{N_\Theta}$
Build the database $\mathcal{G}^\beta$ by using $T$.
for $m = 1, \ldots, N_\Omega$ do
    Pick $\mathbf{\Omega}_m$ from the angular grid.
    $\varepsilon^{\ell,II}(u_n, \mathbf{\Omega}_m) = 0 \quad \forall u_n \in U$
    for $j = 1, \ldots, N_\Omega$ do
        Pick $\mathbf{\Omega}'_j$ from the angular grid.
        Find the scattering angle $\Theta$ given the directions $\mathbf{\Omega}'_j$ and $\mathbf{\Omega}_m$.
        From the database select the grid $\mathcal{G}^\beta$ associated with the scattering
          angle $\Theta$ with size $N_\beta$.
        for $n = 1, \ldots N_\nu$ do
            By using $u_n \in U$ select form $\mathcal{G}^\beta$ the
              quadrature weights $\left\{w_i^\beta\right\}_{i=1}^{N_\beta}$
              and the values $\left\{\left[\mathcal{R}_Q^{\mathrm{II},KK'}\right]_i^\beta\right\}_{i=1}^{N_\beta}$.
            $$\varepsilon^{\ell,II}(u_n, \mathbf{\Omega}_m) = \varepsilon^{\ell,II}(u_n, \mathbf{\Omega}_m) + k_L \frac{w_j^S}{4\pi} \sum_{i=1}^{N_\beta} w_i^\beta \left( \sum_{K,K'=0}^{2} \sum_{Q=-K_{min}}^{K_{min}} \left[\mathcal{R}_Q^{\mathrm{II},KK'}\right]_i^\beta \mathcal{P}_Q^{KK'}\left(\mathbf{\Omega}'_j, \mathbf{\Omega}_m, \mathbf{b}\right) \right) \mathbf{I}\left(u'^\beta_i, \mathbf{\Omega}'_j\right),$$
        end
    end
end
Free the database for $\mathcal{G}^\beta$.
Output: $\varepsilon^{\ell,II}$
```

The correctness and accuracy of the method have been verified by considering the limit of thermodynamic equilibrium (i.e., by assuming that the incident radiation field is unpolarized, isotropic, and given by the Planck function) and verifying that Kirchhoff's law (in the presence of a magnetic field producing Zeeman polarization signals) is satisfied (see Fig. 4.1).

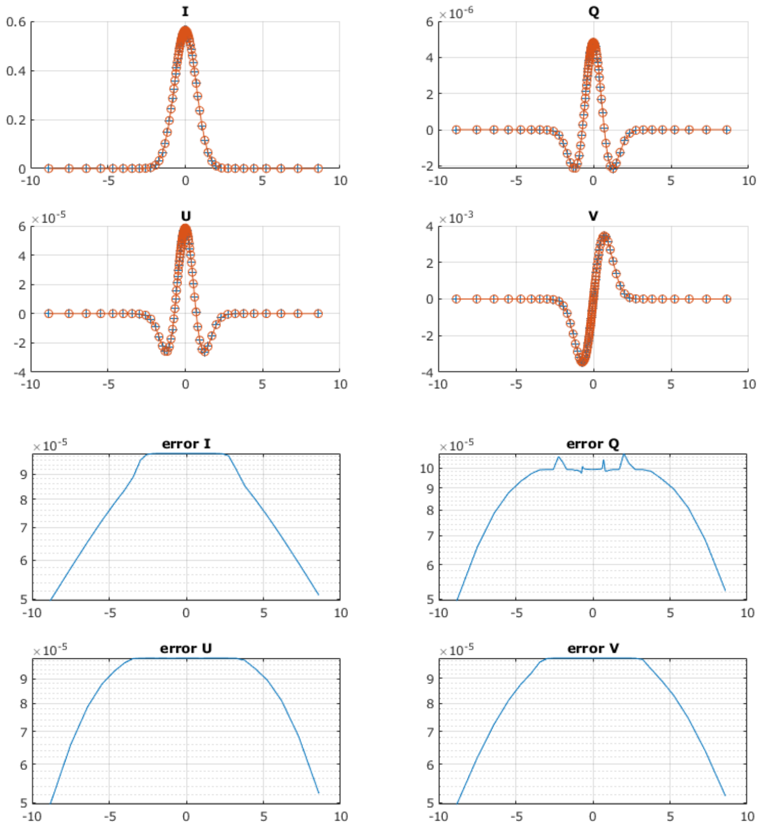

Figure 4.1. *Four upper panels: comparison between the profiles of the emission coefficients calculated with the developed algorithm, assuming an unpolarized, isotropic, and spectrally-flat incident radiation field (orange profiles), and the reference solution (blue profiles), provided by the Kirchhoff's law. Four lower panels: corresponding relative errors as a function of the reduced frequency of the emitted radiation. The calculations have been performed considering the Ca* I *line at 4227 Å, at a height of 2168 km in a 1D semi-empirical model of the solar atmosphere, in the presence of a magnetic field of 220 G, with inclination* $\theta_B = 0.9$ *rad, and azimuth* $\chi_B = 1$ *rad. An emission direction with* $\mu = 0.033$ *is considered.*

4.2 Computation of $\mathcal{R}_Q^{\text{II},KK'}$

4.2.1 Introduction

In the previous sections, we introduced a general strategy for the computation of the emission coefficients $\varepsilon^{\ell,II}$ without discussing the details of the algorithm necessary to calculate the values of the most relevant and critical component, namely the redistribution function $\mathcal{R}_Q^{\text{II},KK'}$ defined in Equation (1.12) on page 14.

In Section 2.3, we studied the behavior of the two main functional components of $\mathcal{R}_Q^{\text{II},KK'}$ (i.e., the functions $h_\beta(\cdot)$ and $g_\beta(\cdot)$) in order to construct reliable quadrature grids using the prior knowledge of the properties of the aforementioned functions. Once the quadrature grids are constructed, the last necessary step is to efficiently compute the quantities $\mathcal{R}_Q^{\text{II},KK'}$.

In this section, we present a method to compute the quantities $\mathcal{R}_Q^{\text{II},KK'}$ given the strength of the magnetic field $||\mathbf{B}||$, and parameters in β, namely: the damping factor a, the scattering angle Θ, and the outgoing reduced frequency u.

4.2.2 The frequency grid

The *frequency grid* (which we indicate with the notation $\text{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$) is a data structure used to store the incoming reduced frequency nodes needed to perform the quadratures for all outgoing frequencies from the discretization of the problem in the set U (see also Section 1.8.2 on page 33), for one scattering angle Θ, and one spatial point $\mathbf{r}$. In practice, it consists of an unstructured mesh in which each outgoing frequency is associated with a quadrature grid over the incoming frequencies (generated with the method described in Sect. 2.6). To access each element of a frequency grid, it is necessary to use a lookup table because the quadrature grids are of different sizes and their values are not aligned.

The position and density of the nodes of a frequency grid depend on the scattering angle, the damping factor (for reasons explained in Section 2.3 on page 39), and the strength of the magnetic field through to the magnetic shifts induced by the Zeeman effects (this last dependence is especially critical in the limit where the scattering angle is equal or close to π, see also Sect. 1.4.4). Finally, it also depends on the Doppler width ($\Delta\nu_{\text{D}}$), since it controls the domain of u (outgoing reduced frequencies) in which the redistribution function (i.e. $\mathcal{R}_Q^{\text{II},KK'}$) is calculated (we recall that in numerical methods only reduced frequencies are used, see also Sect. 1.4.4). In applications, we associate each frequency grid with a *quadrature weight grid* (represented by the notation $\text{W}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$), which is a data structure of the same size as the $\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$ used to store all the quadrature weights. The frequency and quadrature weight grids are used to construct the $\mathcal{G}^{\beta}$ data sets introduced in Section 4.1.

In the following algorithms, we will use the notation:

$$X_{i,j} = f\left(u_i, u'_j\right) \quad \forall\, u_i, u'_j \in \mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$$

to specify that a function is calculated on all nodes of the frequency grid and the resulting value $X_{i,j}$ is an unstructured matrix that fits the size of $\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$.

4.2.3 The Faddeeva function

The most time-consuming part of the numerical computation of the redistribution function $\mathcal{R}_Q^{\text{II},KK'}$ is the Faddeeva function $W(\cdot,\cdot)$ defined in Equation (1.16). In the definition of the redistribution matrix, the Faddeeva function depends on numerous arguments, which are: the parameters (Θ, u, a), the incoming reduced frequencies u', and finally the magnetic shifts $u_{M'_u M_\ell}$, $u_{M'_u M_{\ell'}}$ and $u_{M_u M_\ell}$, $u_{M_u M'_\ell}$, which are calculated with Equation (1.21), which in turn depend on the magnetic quantum numbers from Algorithm 1. We point out that the magnetic shift is proportional to the strength of the magnetic field.

When defining $\mathcal{R}_Q^{\text{II},KK'}$ in Equation (1.12), the Faddeeva function appears as follows:

$$\begin{aligned} &W\left(\frac{a}{\cos(\Theta/2)}, \frac{u+u'+u_{M'_u M_\ell}+u_{M'_u M'_\ell}}{2\cos(\Theta/2)}\right), \\ &\overline{W}\left(\frac{a}{\cos(\Theta/2)}, \frac{u+u'+u_{M_u M_\ell}+u_{M_u M'_\ell}}{2\cos(\Theta/2)}\right). \end{aligned} \tag{4.2}$$

Assuming that the values of Θ, a, u and u' are known, the last remaining dependencies are the magnetic quantum numbers $M_u, M'_u, M_\ell, M'_\ell$ (generated by Algorithm 1 on page 17) and the complex conjugate operator of the Faddeeva function. Thus, we can generalize the calculation of the Faddeeva functions appearing in $\mathcal{R}_Q^{\text{II},KK'}$ with the following modified function:

$$\begin{aligned} &\widetilde{W}_{(\Theta,a,u)} \,:\, \mathbb{R} \times \frac{\mathbb{Z}}{2} \times \frac{\mathbb{Z}}{2} \times \frac{\mathbb{Z}}{2} \times \texttt{Boolean} \longrightarrow \mathbb{C} \\ &\widetilde{W}_{(\Theta,a,u)}\left(u', M_1, M_2, M_3, c\right) = \mathrm{C}\left(c,\, W\left(\frac{a}{\cos(\Theta/2)}, \frac{u+u'+u_{M_1M_2}+u_{M_1M_3}}{2\cos(\Theta/2)}\right)\right), \end{aligned} \tag{4.3}$$

where we recall that the magnetic quantum number are elements of $\frac{\mathbb{Z}}{2}$ being integers and half-integers. The function C returns the complex conjugate of z if c is `true`, i.e.

$$\begin{aligned} \mathtt{C}: \mathtt{Boolean} \times \mathbb{C} &\longrightarrow \mathbb{C} \\ \mathtt{C}(c,z) &= \begin{cases} \overline{z} & \text{if } c = \mathtt{true} \\ z & \text{if } c = \mathtt{false}. \end{cases} \end{aligned} \tag{4.4}$$

The calculation of the Faddeeva function for a given input is performed with Algorithm 7. We also define a modified function to calculate the exponential:

$$\widetilde{E}_{(\Theta,u)}\left(u', M_\ell, M_\ell'\right) = \exp\left[-\left(\frac{u-u'+u_{M_\ell M_\ell'}}{2\sin(\Theta/2)}\right)^2\right], \tag{4.5}$$

which, since it is not critical from the computational stand-point, is computed at each iteration.

Thanks to these definitions, we can describe an efficient algorithm for calculating the quantities $\mathcal{R}_Q^{\text{II},KK'}$, which automatically minimizes the number of evaluations of the Faddeeva function.

Algorithm 7: Method to manage the dictionary of the Faddeeva functiona values

```
Function QueryMQNDict ($\{\widetilde{W}\}$, $\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$, $M_1, M_2, M_3$, $C \in$ Boolean):
    if $[M_1, M_2, M_3, C] \in \text{Keys}\left(\{\widetilde{W}\}\right)$ then
        $W := \text{get}\left(\{\widetilde{W}\}, [M_1, M_2, M_3, C]\right)$
    else if $[M_1, M_2, M_3, \neg C] \in \text{Keys}\left(\{\widetilde{W}\}\right)$ then
        $W := \overline{\text{get}}\left(\{\widetilde{W}\}, [M_1, M_2, M_3, \neg C]\right)$
    else
        $W := \widetilde{W}_{(\Theta,a,u)}(u', M_1, M_2, M_3, C) \quad \forall u, u' \in \mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$
        $\text{insert}\left(\{\widetilde{W}\}, W, [M_1, M_2, M_3, C]\right)$
    return $W$
```

4.2.4 Algorithm

The main goal of the algorithm for calculating $\mathcal{R}_Q^{\text{II},KK'}$ (see Algorithm 8 below) is to build a dictionary that stores the values of $\mathcal{R}_Q^{\text{II},KK'}$ as a function of each triplet of the tensorial indices, i.e. K, K' and Q, which, in combination with the frequency grid, are used to build the set $\mathcal{G}^{\beta}$ introduced at the beginning of this chapter.

Algorithm 8: Calculation of the quantities $\mathcal{R}_Q^{\text{II},KK'}$

```
Input: $\Theta$, $a$, $U$, $||\mathbf{B}||$,
       an atomic model with $J_u$, $J_\ell$
       the local status of the atmosphere.
Build the frequencies grid $\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$ for the parameters $\Theta$, $a$ and $||\mathbf{B}||$
  for each $u \in U$.
Init a dictionary $\left\{\mathcal{R}_Q^{\text{II},KK'}\right\}$ with key $[K, K', Q]$.
Init a dictionary $\left\{\widetilde{W}\right\}$ with key $[M_1, M_2, M_3, \texttt{Boolean}]$
$\tilde{\alpha}_Q$ is the branching ratio function for $R^{\text{II}}$ of the given atomic model.
$\mathcal{C}_{KK'QM_uM'_uM_\ell M'_\ell}$ is a function that returns the value of the quantity in
  Equation (1.15) on page 16.
foreach $K, K', Q \in T$ do
    $\mathcal{R}_Q^{\text{II},KK'} := \text{get\_ref}\left(\left\{\mathcal{R}_Q^{\text{II},KK'}\right\}, [K, K', Q]\right)$
    foreach $M_u, M'_u, M_\ell, M'_\ell \in \boldsymbol{M}^{II}(J_u, J_\ell)$ do
        $W_a :=$ QueryMQNDict $\left(\left\{\widetilde{W}\right\}, \mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}, M'_u, M_\ell, M'_\ell, \texttt{true}\right)$
        $W_b :=$ QueryMQNDict $\left(\left\{\widetilde{W}\right\}, \mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}, M_u, M_\ell, M'_\ell, \texttt{false}\right)$
        $E := \widetilde{E}_{(\Theta,a,u)}\left(u', M_\ell, M'_\ell\right) \quad \forall\, u, u' \in \mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$
        $\mathcal{R}_Q^{\text{II},KK'} := \mathcal{R}_Q^{\text{II},KK'} + \alpha_Q\, \mathcal{C}_{KK'QM_uM'_uM_\ell M'_\ell}\, \frac{1}{\pi \sin\Theta} E \times (W_a + W_b)$
Output: $\left\{\mathcal{R}_Q^{\text{II},KK'}\right\}$
```

In Algorithm 8, $\mathbf{M}^{\text{II}}(J_u, J_\ell)$ refers to the set of magnetic quantum numbers as a function of the respective lower and upper angular momenta, generated with Algorithm 1 on page 17, the notation $\overline{f}(\cdot)$ denotes a complex conjugate function, T is the set of tensorial indices, get_ref returns a reference to an element in the dictionary (i.e., an associative table). The variables W_*, E and $\mathcal{R}_Q^{\text{II},KK'}$ are unstructured matrices storing all values of the respective functions at all coordinates of the given frequency grid $\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$. Finally, the key for accessing an element in a dictionary is a tuple enclosed in square brackets.

The algorithm is based on the fact that the access time required to get an element from a dictionary is constant and significantly less than the direct calculation of the Faddeeva function on all elements of the $\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$. Note that dictionaries are usually implemented as *hash table*, where by definition the access time complexity is uniform: $O(1)$, or as a binary tree, where the access time complexity is logarithmic: $O\left(\log_2 \mathrm{N}\right)$, where N is the number of elements in the dictionary.

In atomic models with a transition $J_\ell = 0 \longleftrightarrow J_u = 1$, namely the models for the Ca I 4227 Å and Sr I 4607 Å lines, the function $\widetilde{W}_{(\Theta,a,u)}(\cdot)$ is calculated 3 times for all elements of the frequency grid, with a total number of combinations of magnetic quantum numbers $N_\mathrm{M} = 9$. On the other hand, in the case of $J_\ell = 1/2 \longleftrightarrow J_u = 3/2$, which is the transition of the Sr II 4078 Å, and Mg II k 2795 Å lines, $\widetilde{W}_{(\Theta,a,u)}(\cdot)$ is calculated 8 times for all elements of $\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$, with $N_\mathrm{M} = 26$. In this way Algorithm 8 automatically minimizes the number of evaluations of the Faddeeva function.

To manage the $\mathcal{C}_{KK'QM_uM'_uM_\ell M'_\ell}$ quantities, it is convenient to store them in a table since they only depend on the atomic model (i.e., they do not depend on the frequencies, scattering angles, and spatial points), and there is no need to recompute them at each iteration (see Sect. 1.8.2).

4.2.5 Conclusive remarks

The calculation of $\varepsilon^{\ell,II}$ is based on three main components:

1. The construction of the frequency grids.
2. The calculation of the quantities $\mathcal{R}_Q^{\text{II},KK'}$.
3. The building of the R^{II} matrices, and the general quadrature performed with Algorithm 6.

If the above components are implemented correctly, the task that takes the majority of the time-to-solution is the calculation of R^{II} given the values of $\mathcal{R}_Q^{\text{II},KK'}$. This is because the calculation of $\varepsilon^{\ell,II}$ requires the construction of a very large number of R^{II} matrices, generating a large amount of data. For this reason, we dedicate Chapter 5 to a detailed discussion of the optimization strategies used to efficiently implement the last part of the algorithm, where optimizations are made taking into account the hardware architecture of the used computer system. In the table 4.1 we can see that, thanks to the methodology introduced here, the time needed to compute the Faddeeva function is *no longer* the bottleneck of the application. Finally, we observed that our approach provides very accurate results, and the re-normalization of R^{II} described in Sect. 1.4.8 is not strictly necessary.

Line	N_{M}	% time-to-solution
Ca I 4227 Å	9	3.9%
Sr I 4607 Å	9	3.9%
Sr II 4078 Å	26	8.4%
Mg II k 2795 Å	26	8.4%

Table 4.1. *Percentage of the total time-to-solution for the R^{II} contribution to the emission vector (i.e. $\varepsilon^{\ell,II}$) spent in computing the Faddeeva function on all emitted directions and frequencies, for the spectral lines considered in this work. We recall that the total number of evaluations of the Faddeeva function is a function of N_M (i.e. the number of combinations of magnetic quantum numbers, see also Algorithm 1, and Sec. 2.2). Reported values were measured with `Linux perf`.*

4.3 Computation of R^{III}

In this section, we present the method used to calculate the contribution to the emissivity of the R^{III} matrix (in its angle-dependent formulation, see also Section 1.4.5 on page 21). This task is particularly difficult because of the appearance of a fourth dimension in the scattering integral. In fact, the fourth dimension significantly increases the number of evaluations of the Faddeeva function, a phenomenon commonly known as the "curse of dimensionality". Furthermore, the integrand appearing in the Eq. (1.30) exhibits a complex behavior that is difficult to handle, and makes the computation of R^{III} more difficult than that of R^{II}.

The generic structure of the algorithm for performing the numerical integration of the contribution of R^{III} to the emission coefficient (i.e. $\varepsilon^{\ell,III}$) is identical to the one used for the $\varepsilon^{\ell,II}$ described in the Algorithm 6. For the computation of the $\mathcal{R}_Q^{\mathrm{III},KK'}$ quantities (see Eq. (1.27)) we have adopted a strategy similar to the one used for $\mathcal{R}_Q^{\mathrm{II},KK'}$, i.e. we exploit the repetitions of the magnetic quantum numbers to minimize the number of evaluations of the Faddeeva function. Finally, an adaptive quadrature based on the Gauss-Kronrod scheme (see Appendix B.3) is used to evaluate the integral over the fourth dimension (i.e. the variable y in Eq. (1.30)). We used a Gauss-Kronrod adaptive quadrature because the construction of frequency grids in the fourth dimension could be time-consuming and memory-consuming without having any advantage in terms of computational time. In fact, the fraction of the solution time required to compute the Faddeeva function is significantly larger than the time required to control the adaptive quadrature.

A detailed comparison of the results obtained using R^{III} and its approximated expression $R^{\text{III-CRD}}$ (see Sect. 1.4.6), for a series of spectral lines of scientific interest, is presented in Chapter 6. The results of this investigation are also collected in Riva et al. (2023).

4.3.1 Algorithm

Following the method introduced in 4.2.3 and using the function $C\,(\cdot)$ (see Eq. (4.4)), the functions $\mathcal{J}$ (see Equations (1.30) to (1.32) on pages 22–23) can be rewritten as:

$$\tilde{\mathcal{J}}_{(\Theta,a,u)} : \mathbb{R} \times \frac{\mathbb{Z}}{2} \times \frac{\mathbb{Z}}{2} \times \texttt{Boolean} \times \frac{\mathbb{Z}}{2} \times \frac{\mathbb{Z}}{2} \times \texttt{Boolean} \longrightarrow \mathbb{C}$$

$$\begin{aligned} \tilde{\mathcal{J}}_{(\Theta,a,u)}\left(u', M_u, M_\ell, c, M'_u, M'_\ell, c'\right) = & \frac{1}{\pi^2 \sin\Theta} \int_{\mathbb{R}} \mathrm{d}y \exp\left(-y^2\right) \\ & C\left(c,\, W\left(\frac{a}{\sin\Theta}, \frac{u + u_{M_u M_\ell} + y\,\cos\Theta}{\sin\Theta}\right)\right) \\ & C\left(c',\, \varphi\left(a,\, u' + u_{M'_u M'_\ell} + y\right)\right), \end{aligned} \tag{4.6}$$

where the complex conjugates are given by the Boolean variables c and c'. The notation above makes it possible to use as a key of a dictionary a tuple of 6 elements consisting of the magnetic quantum numbers (i.e. $M_u, M_\ell, M'_u, M'_\ell$) and the variables c, and c'.

The core component of Algorithm 10, which is used to compute the quantities $\mathcal{R}_Q^{\text{III},KK'}$ (see Eq. (1.27)), is the function **Q** defined in Algorithm 9, which returns the values of the functions $\mathcal{J}$ given the aforementioned key consisting of the tuple of six elements encoding the magnetic quantum numbers and the conjugation flags. The **Q** function simply searches in the dictionary to check if the quantities $\mathcal{J}$ or their conjugates are already stored. If not, it performs the numerical quadrature using an adaptive quadrature based on the Gauss-Kronrod scheme (see also the appendices B.3, and C.1), and inserts the resulting quantities into the dictionary.

The main body of the algorithm is in the pseudo-code 10, where we perform the sum of the quantities $\mathcal{J}$ multiplied by the branching ratio and the quantities $\mathcal{B}$, resulting in the values of $\mathcal{R}_Q^{\text{III},KK'}$. It is straightforward to observe that this method automatically minimizes the number of evaluations of the integral over y in the Equation (4.6), namely the fourth dimension of the problem (and consequently it minimizes the number of evaluations of the Faddeeva function).

Algorithm 9: Query and manage the dictionary $\{\mathcal{J}\}$.

```
Function Q ($\{\mathcal{J}\}$, $\mathrm{F}^{\mathrm{Grid}}_{(U,\Theta,\mathbf{r})}$, $[M_1, M_2, c, M_3, M_4, c']$) :
    key := $[M_1, M_2, c, M_3, M_4, c']$
    conj_key := $[M_1, M_2, \neg c, M_3, M_4, \neg c']$
    if key $\in$ Keys ($\{\mathcal{J}\}$) then
        $\mathcal{J}_{(M_1,M_2),(M_3,M_4)}$ := get ($\{\mathcal{J}\}$, key)
    else if conj_key $\in$ Keys ($\{\mathcal{J}\}$) then
        $\mathcal{J}_{(M_1,M_2),(M_3,M_4)}$ := $\overline{\text{get}}$ ($\{\mathcal{J}\}$, conj_key)
    else
        $\mathcal{J}_{(M_1,M_2),(M_3,M_4)}$ :=
          Gauss_Kronrod $\left(\tilde{\mathcal{J}}_{(\Theta,a,u)}\left(u', M_1, M_2, c, M_3, M_4, c'\right)\right)$   $\forall\, u, u' \in \mathrm{F}^{\mathrm{Grid}}_{(U,\Theta,\mathbf{r})}$
        insert $\left(\{\mathcal{J}\}, \text{key}, \mathcal{J}_{(M_1,M_2),(M_3,M_4)}\right)$
    return $\mathcal{J}_{(M_1,M_2),(M_3,M_4)}$
```

Algorithm 10: Calculation of $\mathcal{R}_Q^{\text{III},KK'}$ in its angle-dependent formulation.

```
Input: $\Theta$, $a$, $U$, $||\mathbf{B}||$,
       an atomic model with $J_u$, $J_\ell$
       the local status of the atmosphere.
Build the frequencies grid $\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$ for the parameters $\Theta$, $a$ and $||\mathbf{B}||$
  for each $u \in U$.
Init a dictionary $\left\{\mathcal{R}_Q^{\text{III},KK'}\right\}$ with key $\left[K, K', Q\right]$.
Init a dictionary $\{\mathcal{I}\}$ with key $\left[M_u, M_\ell, c, M_u', M_\ell', c'\right]$
$\tilde{\alpha}_Q$ is the branching ratio function for $R^{\text{II}}$ of the given atomic model.
$\beta_Q^{K''}$ is the branching ratio function for $R^{\text{III}}$ of the given atomic model.
foreach $K, K', Q \in T$ do
    $\mathcal{I}_{(M_u,M_\ell),(M_u'',M_\ell')} := \text{Zeros}\left(\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}\right)$.
    $\mathcal{I}_{(M_u,M_\ell),\overline{(M_u''',M_\ell')}} := \text{Zeros}\left(\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}\right)$.
    $\mathcal{I}_{\overline{(M_u',M_\ell)},(M_u'',M_\ell')} := \text{Zeros}\left(\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}\right)$.
    $\mathcal{I}_{\overline{(M_u',M_\ell)},\overline{(M_u''',M_\ell')}} := \text{Zeros}\left(\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}\right)$.
    for $K'' = |Q| \ldots 2J_u$ do
        foreach $M_u$, $M_u'$, $M_\ell \in \boldsymbol{M}^{\boldsymbol{III}}(J_u, J_\ell)$ do
            $\mathcal{B}_{K''K'QM_uM_u'M_\ell qq'} :=$ Set by using Equation (A.8)
            foreach $M_u''$, $M_u'''$, $M_\ell' \in \boldsymbol{M}^{\boldsymbol{III}}(J_u, J_\ell)$ do
                $\mathcal{B}_{K''KQM_u''M_u'''M_\ell'q''q'''} :=$ Set by using Equation (A.8)
                $a := \left(\beta_Q^{K''} - \tilde{\alpha}_Q\right)\mathcal{B}_{K''K'QM_uM_u'M_\ell qq'}\mathcal{B}_{K''KQM_u''M_u'''M_\ell'q''q'''}$
                $\mathcal{I}_{(M_u,M_\ell),(M_u'',M_\ell')}$ += $a\,\mathbf{Q}\left(\{\mathcal{I}\}, \left[M_u, M_\ell, \texttt{false}, M_u'', M_\ell', \texttt{false}\right]\right)$
                $\mathcal{I}_{(M_u,M_\ell),\overline{(M_u''',M_\ell')}}$ += $a\,\mathbf{Q}\left(\{\mathcal{I}\}, \left[M_u, M_\ell, \texttt{false}, M_u''', M_\ell', \texttt{true}\right]\right)$
                $\mathcal{I}_{\overline{(M_u',M_\ell)},(M_u'',M_\ell')}$ += $a\,\mathbf{Q}\left(\{\mathcal{I}\}, \left[M_u', M_\ell, \texttt{true}, M_u'', M_\ell', \texttt{false}\right]\right)$
                $\mathcal{I}_{\overline{(M_u',M_\ell)},\overline{(M_u''',M_\ell')}}$ += $a\,\mathbf{Q}\left(\{\mathcal{I}\}, \left[M_u', M_\ell, \texttt{true}, M_u''', M_\ell', \texttt{true}\right]\right)$
    $\mathcal{R}_Q^{\text{III},KK'} := \text{get\_ref}\left(\left\{\mathcal{R}_Q^{\text{III},KK'}\right\}, \left[K, K', Q\right]\right)$
    if $\Theta \in \{0, \pi\}$ then
        $m := \pi^{-5/2}$
    else
        $m := \left(\pi^2 \sin\Theta\right)^{-1}$
    $\mathcal{R}_Q^{\text{III},KK'} :=$
      $\frac{m}{4}\left(\mathcal{I}_{(M_u,M_\ell),(M_u'',M_\ell')} + \mathcal{I}_{(M_u,M_\ell),\overline{(M_u''',M_\ell')}} + \mathcal{I}_{\overline{(M_u',M_\ell)},(M_u'',M_\ell')} + \mathcal{I}_{\overline{(M_u',M_\ell)},\overline{(M_u''',M_\ell')}}\right)$
```

4.4 Parallelization

In this chapter, we observed that the calculation of emission coefficients consists of two main phases, i.e. the calculation of the quantities $\mathcal{R}_Q^{\mathrm{II},KK'}$, followed by the quadrature of the emission coefficients. In the first phase (see Section 4.2), we build the data set $\mathcal{G}^{\beta}$, where the calculation of each entry, which depends on u and Θ, is independent of the other entries. Thus, in each process of a parallel application, we calculate all values of $\mathcal{R}_Q^{\mathrm{II},KK'}$ for all u and u' associated with a single scattering angle independently. In the second phase, where the emission coefficient is calculated (see algorithm 6), we observe that the calculation of each outgoing frequency and Stokes parameter is an independent quadrature, and therefore, also in this case, we subdivide and parallelize the procedure using a pool of independent processes. To implement these methods, in both cases, we used the *producer-consumer* model, where a fixed number of processes (or consumers) receive jobs through a synchronized queue filled by a producer. We developed these methods using the standard C++ thread library.

In “productive” applications, for the considered case of the Ca I 4227 Å line (in the FAL-C model), the average time to compute the emission vector in one direction and all frequencies, taking into account all contributions to the emission coefficient (considering $R^{\mathrm{III-CRD}}$), is ~ 0.03 sec on a 20-core Xeon CPU (USI cluster).

In the case of R^{III} (see section 4.3), since it is impossible to compute the quantities $\mathcal{R}_Q^{\mathrm{III},KK'}$ in their angle-dependent form on the fly (as we do with $\mathcal{R}_Q^{\mathrm{II},KK'}$), they are precomputed and stored out-of-core (practically in the scratch drive). To compute these quantities, we again used a producer-consumer model built in MPI[1] and distributed across many nodes and processes of a supercomputer. This approach allows the workload within the processes to be self-balancing so that no process goes into a wait state. Avoiding waiting states is necessary since the computation time on each node of the mesh is highly variable. Since the two algorithms have the same structure and the only real difference is in the calculation of the quantities $\mathcal{R}_Q^{\mathrm{II},KK'}$ and $\mathcal{R}_Q^{\mathrm{III},KK'}$, in the second phase (where the emission vector is calculated) we applied the same strategies as for R^{II}.

[1] MPI Forum: https://www.mpi-forum.org/

Chapter 5

High-performance programming strategies

The calculation of the emission vector $\varepsilon^{\ell,II}$ is a challenging task due to the difficulties described in the previous chapters, resulting from the complexity of the involved R^{II} redistribution matrix. In Chapter 4 we introduced an algorithm for the efficient calculation of $\varepsilon^{\ell,II}$, which takes advantage of the repetition of the scattering angles (resulting from the geometry of some angular grids) and the property of the functions.

However, simply implementing the above strategies is not enough to achieve optimal performance. In fact, to take full advantage of modern CPUs, it is essential to design the data structures and algorithms taking into account the design and hardware characteristics of superscalar CPUs or GPUs.

5.1 Introduction to the computation of the R^{II} matrices

The computation of R^{II} matrices, followed by the quadrature of the emission vector, is the most time-consuming task due to the large number of matrices to process. Practically, it requires about the 80% of the *time-to-solution* (TTS) in the Ca I 4227 Å line (note that the Faddeeva function computation needs less than the 5% of the TTS, see also Tab. 4.1, Eq. (1.16), and Sect. 2.2).

To give an idea of the size of the problem, we consider the typical mesh for the Ca I 4227 Å line in the atmospheric `model C` from Fontenla et al. (1993) (a.k.a. FAL-C, see also Chap. 7). In the discretization of this problem, the number of discrete frequencies in the grid U is 99, the size N_{Ω} of the angular grid S is

96 (see also Section 1.8.2), and the average size of the quadrature grid over the incoming frequencies ν' for one outgoing frequency is in average $N_\beta \approx 150$ nodes. Therefore, to compute the emission vectors for all outgoing directions and frequencies at a spatial point $\mathbf{r}_i$, it is necessary to compute

$$N_\beta \times N_\nu \times N_\Omega^2 \approx 136 \times 10^6$$

R^{II} matrices. If we perform the calculation in `float64` format (see remark 5.1), it is theoretically necessary to use about 16 GB of DRAM to store all these matrices.

We should also note that to solve the RT problem in a 3D atmospheric model, it is essential to compute the emission vectors for many spatial points on each computing node (at least one per CPU core) of a supercomputer. Considering that the latest HPC systems[1] typically have 64 cores per node and a maximum of 512 GB of DRAM, it is easy to conclude that it is impossible to store all the necessary R^{II} matrices in DRAM and reuse them in each iteration. We must also point out that in practical 3D applications, due to the large memory requirements needed to store the radiation field and the data to solve the linear system, it is difficult to additionally store the sets of all $\mathcal{R}_Q^{\text{II},KK'}$ as a function of the scattering angles (see also the methods introduced in Chapter 4, Sec. 1.8.2, and Benedusi et al. (2023)). For these reasons, it is necessary to implement a highly efficient method that does not require a large amount of memory (in particular, it is imperative to compute the R^{II} matrices, and the emission vectors $\varepsilon^{\ell,II}$, as fast as possible by minimizing the amount of used memory and its transfer).

To achieve this goal, we designed optimal data structures that allowed us to develop applications that use the CPU with an efficiency close to the theoretical maximum (it should also be noted that high-performance applications can reduce energy consumption despite the high peak power absorption of the CPU (e.g. Padoin et al., 2012)). The main necessary step to achieve this goal is to design data structures that respect the *spatial locality* of the data. Namely, the data accessed via the indices of the innermost loops must be contiguous in memory. If this requirement is met, it is possible to implement a highly efficient algorithm.

Remark. *Due to the very large ratios ($\gg 10^6$) between the maximum and minimum values (of variables and parameters) that occur in both emission vector calculations and atmospheric models, it is preferable to use the* `float64` *format to ensure numerical stability.*

[1] The TOP500 project

Structure of this chapter

Part I: *Computer systems architectures*

- **Section** 5.2 briefly introduces the architectural concepts of modern **superscalar CPUs**.
- **Section** 5.3 introduces the problem of the **memory management** in computer systems, pointing out that this is the **most critical** aspect to be considered in algorithms development.
- **Section** 5.4 introduces the **GPUs**.
- **Section** 5.5 briefly introduces the Tensor processors.

Part II: *Performance analysis and data structures*

- **Section** 5.6 proposes the **roofline model** as a performance analysis method and demonstrates its high reliability with some examples.
- **Sections** 5.7, 5.8, and 5.9 introduce **data structures** and the manipulations necessary to implement a simple and fast procedure for the computation of the R^{II} matrices. Finally, we apply the roofline model as a method to decide the most appropriate computer system architecture to execute this procedure.

Part III: *Practical applications and bechmarks*

- **Section** 5.10 introduces the **three main programming strategies**, based on SIMD, that allow the development of a nearly-optimal routine.
- **Section** 5.11 presents the results of some **benchmarks** where, by applying the roofline model, we demonstrate the near-optimality of the method introduced in this Chapter.
- **Section** 5.12 proposes two alternatives strategy for the computation of the emission vector.
- **Section** 5.13 provides the conclusions.

5.2 CPU architecture and concept used in high-performance programming

The goal of this section is to introduce the basic concepts and terminology used in high-performance programming in a way that allows a complete and reliable analysis of the code used to compute the R^{II} matrices, and thus to present a general methodology that has been used in the other parts of this project or can be applied to other numerical applications. Although in this section we refer to the x86-64 architecture (Intel Xeon and Core or AMD's Epyc and Ryzen CPUs), all the concepts discussed here can be applied to other architectures (such as ARM, Apple M, and IBM Power).

5.2.1 Superscalar CPUs

A superscalar CPU is a microprocessor architecture that can hold and execute multiple operations (per core) simultaneously. This result is achieved by several mechanisms that we will describe in this section.

In CPUs and GPUs, the *Program Counter* (hereafter PC) is a register that stores the address of the instruction to be dispatched, and when the current instruction is fetched, the value of PC is incremented by 1 (i.e., it stores the address of the next instruction).

In a CPU (and similarly in GPUs), the execution flow of a program is controlled by *instruction pipelining*, where each stage of one instruction is processed simultaneously with the next stage of the previews instructions (the main stages of a CPU pipeline are Instruction Fetch, Instruction Decode, Execution, Memory Access, Register Write Back).

In superscalar CPUs, the execution unit (a.k.a. *core*) consists of N *ports* that can simultaneously hold up to N instructions in execution. Note that the cores in a CPU are independent of each other and are used for parallelism between threads or processes (symmetric multiprocessing), although *ports* are interdependent and are used to parallelize the execution flow of a program (one thread or one process) by applying the *instruction level parallelism* (e.g. Hennessy and Patterson, 2019).

Figure 5.1 shows a scheme of the Intel Golden Cove CPU microarchitecture, where the structure of the pipeline is visible and each port P0 . . . P11 controls a set of arithmetic logic units (ALU) and a set of address generation units (AGU). For example, port 0 (P0) controls the set of integer (INT) ALUs and the set of VEC (SIMD) units. Ideally, in the case of the Golden Cove microarchitecture, in the best-case scenario, up to 11 simultaneous operations can be kept running.

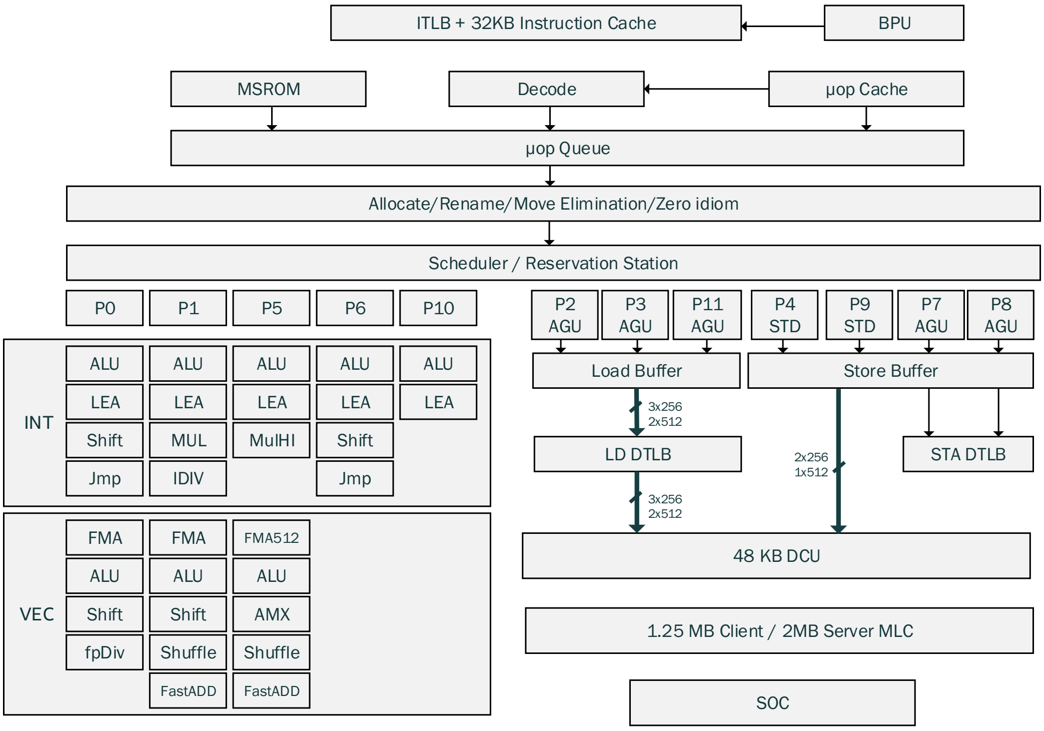

Figure 5.1. *Basic schematic representation of the Intel Golden Cove microarchitecture. Observe that ports 0, 1, and 5 have SIMD capabilities, and port 5 has AMX extensions (see also Sect. 5.5). Source: Intel® 64 and IA-32 Architectures Optimization Reference Manual Figure 2-1.*

One of the most remarkable ideas introduced in superscalar CPUs is the so-called "speculative execution", i.e. the ability to execute the most likely branch of the program (e.g. branch prediction) before the results of the current operations necessary to decide the effective branch are ready.

In high-performance programming, we are particularly interested in cache prefetching, which is the ability of CPUs to speculatively move data from the main memory (DRAM, which typically has a relatively slow access time) to the CPU's cache before it is used (or referenced). The simplest way to successfully use cache prefetching is to respect the locality of the data so that the CPU can easily copy into the cache the data that will be used. Note that cache prefetching is speculative, so there is no guarantee that the copied data will be used effectively.

The successful cache pre-fetching rate is measured by the mean of the *Prefetching accuracy*[2], i.e:

$$\text{Prefetching accuracy} = \frac{\text{Used prefetches}}{\text{Missed prefetches} + \text{Used prefetches}}, \qquad (5.1)$$

where "Used prefetches" is the number of prefetches actually used by the program, and "Missed prefetches" is the number of failed prefetches. We must note that the *prefetching accuracy* strongly depends on the execution flow of the program and the design of the data sets.

As explained at the beginning of this section, a superscalar CPU can perform multiple operations simultaneously. Therefore, the execution flow of the program must be designed so that the CPU workload is always close to the theoretical maximum. Throughput and latency are used to measure CPU workload.

Throughput refers to the number of results returned per unit of time, while *latency* refers to the units of time required to perform a single operation. It is easy to see that throughput and latency in superscalar CPUs are not directly related. This is because superscalar CPUs have many ports that can hold more than one operation in execution simultaneously, starting subsequent instructions on a free port while other ports are executing previously started instructions, so that the operations required to read and move results from registers to their destinations in the main memory (or vice versa) can be performed simultaneously with mathematical and logical operations (see also Figure 5.2).

Another important concept is the *resource pressure*, which is the ratio between the number of operations being performed and the number of operations that could be performed simultaneously. Low or zero pressure means that the CPU is working at maximum efficiency.

Metrics like resource pressure, throughput, and latency are used by static analysis tools (such as llvm-mca[3]) to measure (or predict) the efficiency of a program. For example, an average resource pressure greater than one can indicate a poorly optimized execution flow or too much time spent in a waiting state.

To optimize execution flow and consequently improve performance, modern CPUs use *out-of-order* execution (Hennessy and Patterson, 2019), which is the ability to reorder the sequence of program instructions so that the time spent in the waiting state is minimized. For example, in algorithm 11, operation 1 (OP 1) has a latency of 4 and writes its result to register R3; it is easy to see that if

[2] The *Prefetching accuracy* and other metrics related to the execution flow of a program can be measured through a profiler, e.g. *Linux Perf* (`https://perf.wiki.kernel.org/index.php/Main_Page`, see also Bakhvalov et al. (2020))

[3] llvm-mca - LLVM Machine Code Analyzer

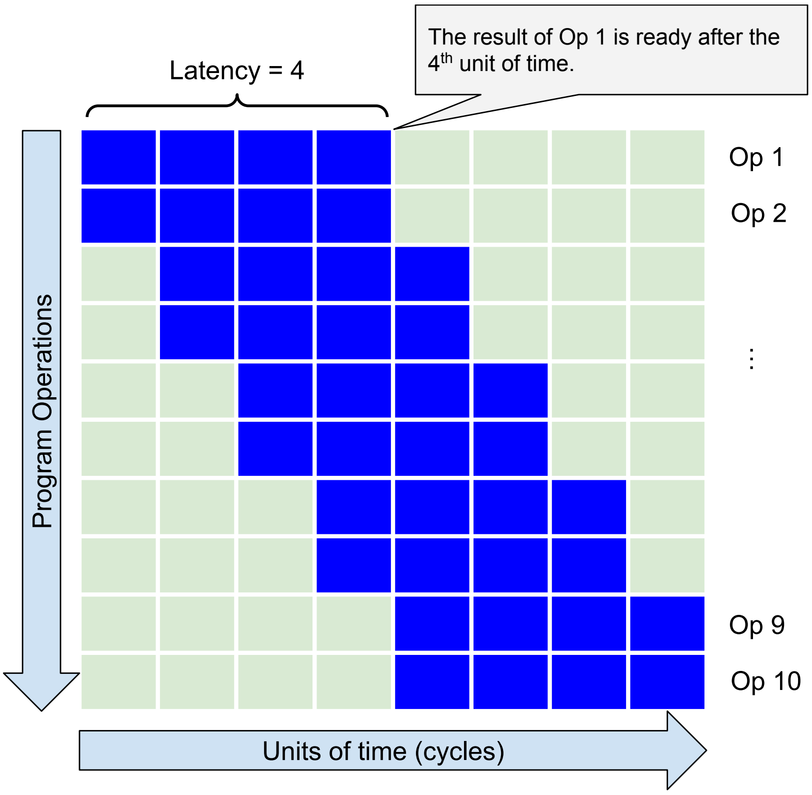

Figure 5.2. *Schematic representation of the execution flow (aka timeline) of a superscalar CPU with a throughput of 2 (it can start two operations per unit of time), 8 ports (it can hold 8 simultaneous operations in execution), and a latency of 4 (the result of each operation is returned after 4 units of time). Because of the latency, we can only use the results of the first or second operation by the 9^{th} or subsequent operations. If the result of operation 1 (or 2) is used before operation 9^{th}, the execution flow is suboptimal (hence the effective throughput is less than 2). On real CPUs, the actual latency varies depending on the operation and execution flow. For this reason, when implementing high-performance code, it is important to fragment the computation into its basic steps so that the compiler can apply the most appropriate optimizations and the CPUs maximizes the throughput.*

the CPU respects the in-order execution, it must wait for OP 1 to finish in order to execute OP 2, since it depends on the result of OP 1 (which is written to R3), although Operation 3, which is independent of the previews, could be executed at the same time as OP 1. In a superscalar CPU, the most convenient execution order is the second, where the execution order of OP 2 and 3 is swapped, and sent to two different ports. CPUs with out-of-order capability can detect dependencies between microoperations and automatically reorder the execution flow in a way that maximizes throughput, this is typically done by using Tomasulo's algorithm (Hennessy and Patterson, 2019, Chap. 3). In addition, compiler optimizers can detect dependencies between instructions and automatically reorder the execution flow to improve throughput (in some cases, this operation should also be done by the programmer).

```
Algorithm 11: Out of order execution flow
  In-order execution flow:
  1. ADD (R1, R2) → R3                    ▷ Latency = 4
  2. MUL (R3, R4) → R4                    ▷ Wait until the result in R3 is ready
  3. MUL (R6, R4) → R5                    ▷ It is independent from the previews ops

  Out-of-order execution flow:
  1. ADD (R1, R2) → R3                    ▷ Latency = 4
  3. MUL (R6, R4) → R5                    ▷ It can be executed simultaneously with op1
  2. MUL (R3, R4) → R4                    ▷ If last the waiting time reduces
```

5.2.2 SIMD operations (AVX)

One of the most powerful features available in modern CPUs for developing numerical applications are SIMD arithmetic operations (Single Instruction Multiple Data) or vector operations (AVX on x86 CPUs). The main advantages of SIMD operations over GPUs are that they do not require the huge amount of data transfer that is normally required on GPUs, and the second remarkable property is that they can take advantage of all the advanced features of CPUs mentioned above. These advantages make the use of SIMD operations on CPUs particularly suitable for memory-bound, unstructured, or highly fragmented applications. AVX extensions are successfully used in many numerical applications, such as Intel MKL, video editing (e.g., the x265 encoder), image processing, and so on.

On x86 CPUs, the use of AVX operations is generally exploited by using the `immintrin.h`[4] library.

With the AVX2 extensions in `float64`, it is possible to perform SIMD operations on vectorized registers of 4 elements (and 8 for the `float32` format), so we could expect an ideal speed-up of 4 (see also Figure 5.3), while on AVX512, the size of the vectorized registers is 8 (in `float64`).

[4] See also Intel® Intrinsics Guide

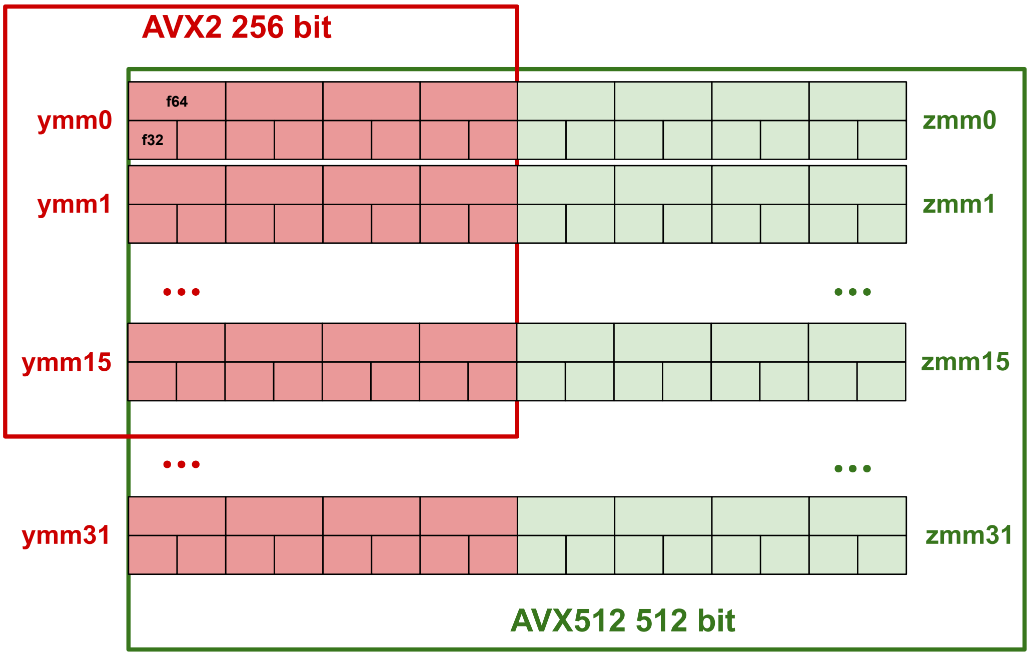

Figure 5.3. *Schematic representation of the AVX2 and AVX512 registers (labeled ymm and zmm, respectively). Most recent CPUs with AVX512 extensions have 32 vector registers, while those with AVX2 have 16. With AVX2, the length of a register is 256 bits so that SIMD operations can be performed on vectors of 4 elements in `float64` or 8 elements in `float32`. With AVX512, the length of a register is 512 bits, which doubles the size for SIMD operations.*

AVX instructions include elementary arithmetic operators, comparisons, reductions (greatly extended in AVX512), type conversion, shuffling, and only in AVX512: convolution accelerator, and masks. A class of operations that are very useful for operations on complex numbers, available in AVX (and thus useful for calculating R^{II} matrices), are the *fused multiplication and addition* (FMA), i.e. $\text{FMA}(\mathbf{a},\mathbf{b},\mathbf{c}) = \pm\mathbf{a}*\mathbf{b}\pm\mathbf{c}$ (some CPU models feature 4 FMA units, so the CPU can hold up to 4 FMA operations in execution simultaneously)[5].

A method to estimate the maximal peak performance (in FLOPS) of a CPU is given in the following equation (Bakhvalov et al., 2020):

$$\text{Peak\_FLOPS} = \#\text{Cores} \times \frac{\#\text{Bits(SIMD vector)}}{\#\text{Bits(Float Word)}} \times 2 \times \text{Clock}\,, \tag{5.2}$$

where 2 is the number of FLOP of an FMA operation. It should be noted that the

[5] Hardware support for FMA operators is also provided on NVidia GPUs.

above equation is incomplete as it does not take into account the latency and the configuration of the CPU ports; for this reason, in this work, we preferred to use the performance specifications published by the manufacturer, which in practice have proved to be more reliable.

5.3 Memory Hierarchy and Bandwidth

Perhaps the most critical feature of modern computer systems is memory management (cache, HBM, DRAM, and hard drive). The impact of memory and especially the *memory bandwidth* (i.e. the data transfer rate between the memory to/from the CPU) is of high relevance in the majority of numerical algorithms, including the computation of the emission vector. This arises from the fact that since about 2001, the computational capacity of CPUs and GPUs grows faster than the memory bandwidth (e.g. Solihin, 2015; Clapp et al., 2015). The result of this evolution is that the memory bandwidth (or equivalently, latency or access time) is, in the majority of cases, the most important technical specification to be considered in the design of algorithms and computer systems (see also Sect. 5.6). This is because the computational capacity of CPUs or GPUs to process data is far superior to the capacity of the memory channels to transfer the required data. In superscalar CPUs and GPUs, memory access time is typically inversely proportional to physical distance and capacity. For example, hard drives with large capacity that are physically far from the CPU will have a low transfer rate. On the opposite, the L1 cache, which has a small capacity and is physically close to the CPU, shows a very high transfer rate.

In computer systems, the memory is, typically, organized in a hierarchy of three layers of cache, normally embedded in the CPU packaging, the external main memory DRAM and the hard drive. The cache is used to store the portions of data or programs that are most likely to be used (this is generally done through the prefetch and the speculative execution). The access times for the three levels of cache and DRAM are reported in Table 5.1.

The read/write activity to the main memory (DRAM) is typically made through different parallel memory channels, whose number can vary from 2 in consumer CPUs up to 12 in server or HPC CPUs. The addressing of memory pages is normally interleaved to the channels so that the *words* with addresses N, $N+mW_L\dots$ for $m <$ `#Channels` are physically stored in different memory slots and can be transferred to the CPU in parallel (in modern CPUs, the word length W_L is of 64 bits). This highlights the importance of designing the data management and algorithms where the *locality of the data* is respected (see also the roofline model

Memory component	Latency [cycles]	time [ns]
L1 Cache	4	1
L2 Cache	10 - 25	5 - 10
L3 Cache	40	20
DRAM	>200	100

Table 5.1. *Typical latency of memory components in modern computer systems, source (Ch. 4 Bakhvalov et al., 2020). Comparison of latencies and times emphasizes the importance of developing programs that allow efficient data prefetching.*

in Sect. 5.6, and Sect. 5.7).

HBM and the Fugaku supercomputer

The most recent CPU models equipped with *High Bandwidth Memory* (HBM) technology can achieve up to 1000 GBytes/sec of bandwidth. One of the most interesting consequences of the use of HBM is the *Fugaku supercomputer* (see: www.top500.org), where HBM is the main technology that allows this *CPU-only* machine to outperform supercomputers equipped with GPUs. The CPU model used in the Fugaku is the *Fujitsu A64FX* which is based on the ARM architecture and has 48 cores. This CPU, equipped with 512-bit SIMD units, shows a peak performance of 2.7 TFLOPS[6] in `float64`, which is comparable to that of Intel Xeon CPUs with a similar number of cores and AVX512. Therefore, the only technology that allows the Fugaku supercomputer to outperform GPU-based supercomputers is the very high memory bandwidth of 1000 GBytes/sec (i.e. HBM) provided by the CPUs used in this machine.

HBM has also recently been introduced in the Intel Xeon Max CPU series. Finally, HBM is used as the internal memory of GPUs (e.g. the NVidia A100), and in many applications the main advantage of GPUs is given primarily by the internal bandwidth, which is usually very high (see Sect. 5.6, and Tab. 5.3).

[6] Fujitsu Presents Post-K CPU Specifications

5.4 GPU - Graphic Processor Unit

5.4.1 Introduction

The main idea of GPU (Graphic Processor Unit, or SIMT Single Instruction Multiple Threads) is based on the fact that numerous numerical applications consist of relatively simple functions applied to each node of large and structured datasets. GPUs were originally developed for gaming applications and video acceleration support, in which the vast majority of numerical operations consist of texture manipulation (e.g., image anti-aliasing, projection, etc. ...) and geometric operations in homogeneous coordinate systems on very large meshes (that is, linear operations on 4×4 matrices). Only later they were used in scientific computing and other numerical applications.

5.4.2 GPU in scientific computing

In scientific computing, applications with similar characteristics include time-dependent finite differences or Monte Carlo integration methods. All these applications have in common that the execution flow of the program is identical (and non-divergent) at each node of a structured data set. Therefore, these operations can be performed simultaneously on parallel and synchronized ALUs (a.k.a. CUDA Core, Vector Lane, Thread Processor, Stream Processors) sharing the same program counter (PC). This means that the same operation of the same program is performed on each ALU of the GPU (i.e. the ALUs are controlled by the same PC) by operating on different data. Obviously, in this class of applications, the access to the data should be relative to the node, so that the indices of the vectors containing the data can be calculated by simple arithmetic operations and do not need to be extracted by maps, lookup tables, or search algorithms.

It is easy to conclude that the main architectural idea of GPUs is to have a large number of ALUs sharing the same Program Counter and other logic components so that the same program is executed in a synchronized manner on a set of ALUs by operating on different data.

With GPUs, the main goal is to organize the workload of a program into a grid of threads, where each thread is associated with an absolute grid coordinate that is used to retrieve data. Threads are thus grouped into blocks (WARPs on NVidia GPUs) and executed in parallel. Due to these characteristics, a GPU can achieve very high throughput when operating under optimal conditions.

Difference between SIMD (on CPU), and SIMT (on GPU):

- In SIMD, variables are declared as vectors and the program must explicitly call vectorized operations.
- In SIMT, variables are declared as scalars, and the GPU logic automatically spawns operations on scalars into vectorized operations.

5.4.3 Architecture of NVidia GPUs

A GPU is designed to spawn a *scalar program* (kernel) into a very large number of threads executed in parallel on a large set of vectorized processors. The main processing unit at the center of the mechanisms allowing this massive and automatic vectorization is the *Streaming Multiprocessor* (hereafter SM, Figure 5.4). An Nvidia GPU is a grid of SMs that is designed to perform this vectorization efficiently.

An SM consists of 2 or 4 GPU-SIMD units of 32 `float32` or 16 `float64`[7] elements. Each element of a GPU-SIMD unit is called *Thread Processor* (hereafter TP, a.k.a. Stream Processors, or CUDA core). It is important to note that all TPs in an SM share the same *Program Counter* (PC), *Shared Memory*, and *L1 cache*. We must observe that in Hennessy and Patterson (2019) the SM is called *Multithreaded SIMD Processor* highlighting the numerous similarities of an SM with a SIMD unit. In addition, the TPs in a GPU-SIMD unit also appear to share the same *Register File* (although NVidia has not yet disclosed any technical descriptions of how the Register File is designed and managed by the SM, it is quite obvious that TPs simply operate on one index of vectorized registers). On most recent GPUs, the GPU-SIMD units are alternatively an equal number of *Tensor Core*. In general, an SM can be said to be a set of GPU-SIMD units that operate on vectorized registers of 16 or 32 elements, or a TPU that operates on 4×4 tiles (see also 5.5).

Finally, a GPU consists of a set of SMs that share the same L2 cache (Figure 5.5), internal memory, and PCI port (Newer GPUs, such as the NVidia A100, have two independent L2 cache blocks). Since the L1 and L2 caches are shared, the

[7] Hardware support for the `float64` format is limited to GPU models dedicated to high-performance computing or data centers. Consumer GPU models (e.g. NVidia GTX series) only provide hardware support for single precision (`float32`) floating-point numbers.

total amount of cache memory per TP is usually small[8] and access to the cache is concurrent. GPU memory is accessed through a series of parallel memory channels, the number of which varies from 4 in consumer GPUs to 12 in HPC GPUs (NVidia A100 has 12 memory channels).

On GPUs, the hardware is optimized to handle the massive concurrency generated by threads on L1 and L2 cache and internal memory and to control the workload of different SMs by assigning the correct set of threads to each. GPUs do not have out-of-order or speculative execution (like superscalar CPUs). Therefore, these types of optimizations are left to the compiler (at the cost of a significant loss in execution efficiency).

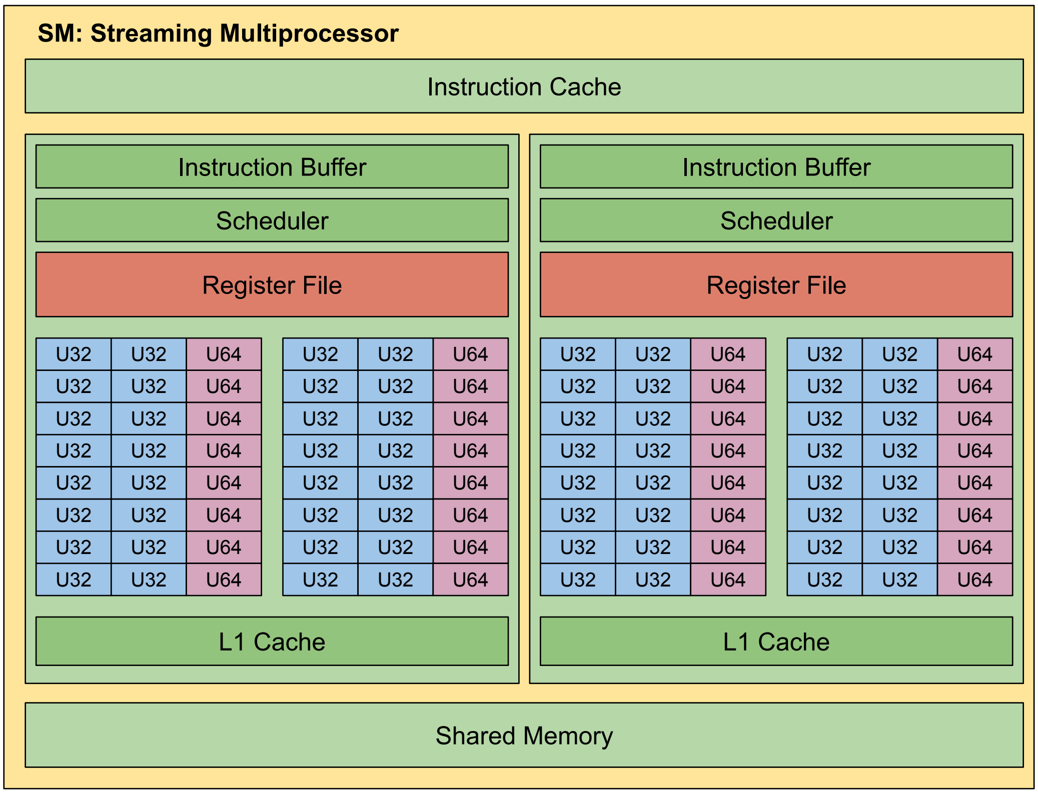
SM: Streaming Multiprocessor
Instruction Cache
Instruction Buffer
Instruction Buffer
Scheduler
Scheduler
Register File
Register File
U32 U32 U64 U32 U32 U64 U32 U32 U64 U32 U32 U64
U32 U32 U64 U32 U32 U64 U32 U32 U64 U32 U32 U64
U32 U32 U64 U32 U32 U64 U32 U32 U64 U32 U32 U64
U32 U32 U64 U32 U32 U64 U32 U32 U64 U32 U32 U64
U32 U32 U64 U32 U32 U64 U32 U32 U64 U32 U32 U64
U32 U32 U64 U32 U32 U64 U32 U32 U64 U32 U32 U64
U32 U32 U64 U32 U32 U64 U32 U32 U64 U32 U32 U64
U32 U32 U64 U32 U32 U64 U32 U32 U64 U32 U32 U64
L1 Cache
L1 Cache
Shared Memory

Figure 5.4. *Schematic representation of a Streaming Multiprocessor of an NVidia GPU*

[8] In the NVidia A100 there are in total 40 Mb of L2 cache and 3456 `float64` TP. Which results in $\sim$ 11 Kb of L2 cache per TP. While in the Intel Golden Cove Microarchitecture, there are $2 + 1$ ports with AVX512 (SIMD) capabilities for a total of 16 vectorized operations per core, considering 2Mb of L2 cache per core this results in $\sim$ 125 Kb of L2 cache per `float64` vector element. Moreover, on CPUs, the L2 cache is not shared between cores.

On an NVidia GPU, threads are executed in WARPS. A WARP consists of a set of at most 32 threads. All threads in a WARP are controlled by the same PC, i.e. their execution is synchronized, or in other words, all threads in a WARP execute the same instruction at the same time, but on different data. The resources of each WARP are limited by the sizes of the register file, shared memory, and the L1 cache. If WARPs require more resources than are physically provided by the SM, the GPU scheduler, to satisfy the requirements reduces the number of WARPs that can be executed simultaneously on an SM. This operation greatly reduces GPU parallelism and typically occurs when a complex program is executed (it should be noted that in some cases reducing the number of threads per SM can improve the efficiency and consequently the throughput of a GPU (e.g. Lashgar et al., 2012)). In the case of a divergent execution flow within a WARP, the scheduler of the SM executes one branch at a time so that all threads with a different execution flow enter a wait state (the classic example is the if-else statement). Finally, the execution of a WARP does not terminate until all threads belonging to it have finished.

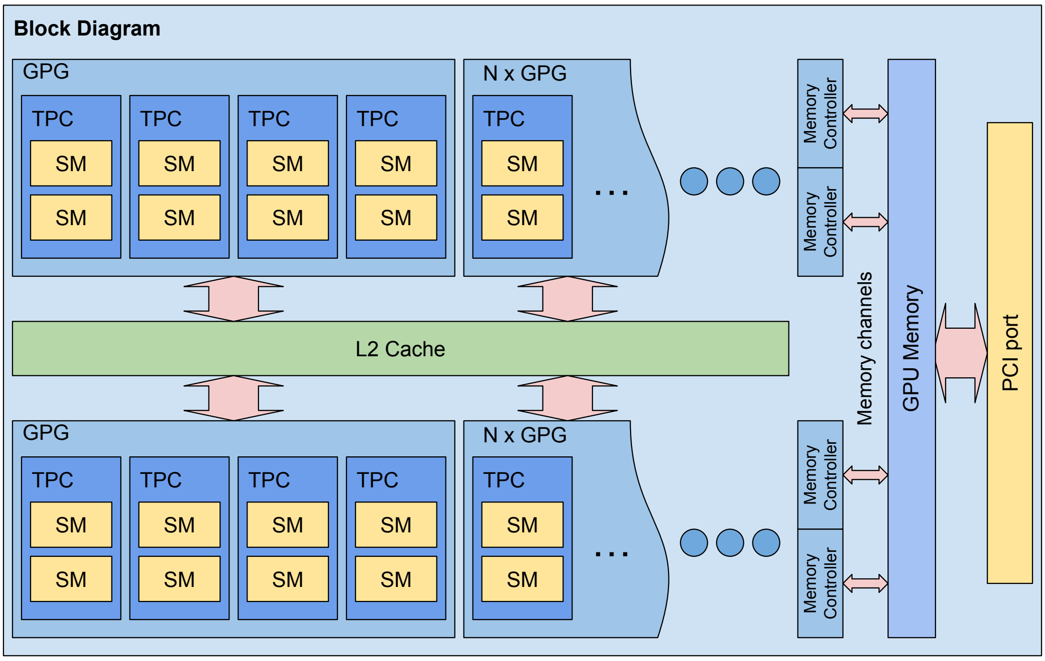

Figure 5.5. *Schematic representation of an NVidia GPU. TPC stands for Texture Processor Cluster, and GPC stands for Graphics Processing Cluster.*

On NVIDIA GPUs, threads are grouped into *Threading Blocks* (hereafter TB), each of which is a set of at most 1024 threads. Finally, threads belonging to a

TB are packed into different WARPs. The GPU scheduler organizes the execution flow so that all WARPs owned by a given TB are executed on the same SM, allowing threads in the same TB to share the same *Shared Memory* (see also figure 5.6). It is important to note that WARPs running on different SMs do not share the same PC.

This architecture allows the scheduler to load and execute a WARP in an SM while other WARPs are running on other SMs. Consequently, the execution of different WARPs, owned by different TBs, is independent of each other, which means that execution times can be different and execution flows between different WARPs running on different SMs can diverge.

The *shared memory* is a low-latency memory that can be directly referenced by applications in the CUDA source code and used as a local and fast memory within a TB, and SMs access to it through a multichannel bus (Ding and Williams, 2019).

A CUDA kernel (i.e., thread) is executed by specifying the number of threading blocks and the number of threads per block (recall that the number of threads per WARP is fixed [32] or decided by the scheduler). The exact coordinate of a thread in the grid can be retrieved through some built-in variables assigned by the scheduler, i.e:

`threadIdx.x`: is the id of the current thread (inside a block).
`blockIdx.x`: is the id of the current threading block.
`blockDim.x`: is the number of threads in a block.
`gridDim.x`: is the number of blocks in a grid.

Where the absolute index of a thread in the grid is given by:

```
int tid = threadIdx.x + blockIdx.x * blockDim.x;
```

The primary limitation of a GPU is its access to the main memory (DRAM) through the PCI, which is typically slow and limits the classes of algorithms that can be efficiently implemented on a GPU (see also Section 5.6).

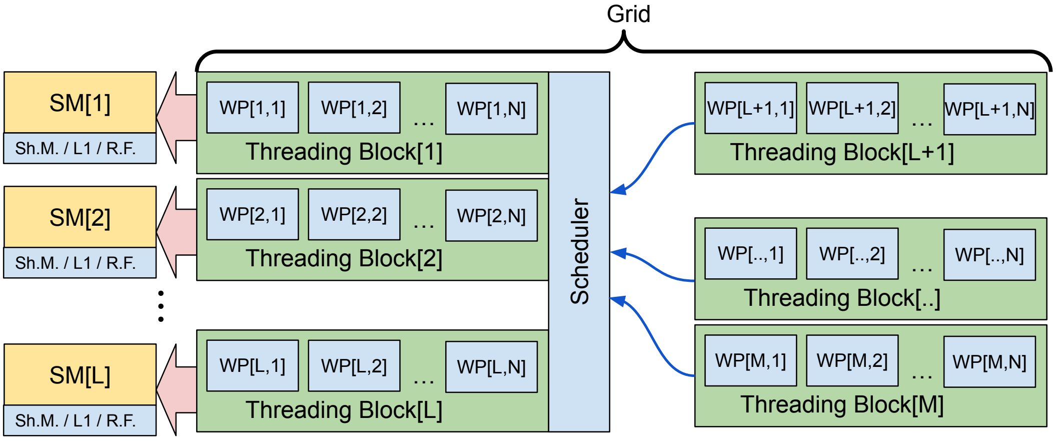

Figure 5.6. *Schematic representation of the execution flow of an NVidia GPU with L SMs. Each threading block has N WARPs (WP) which are executed on the same SM by sharing the shared memory (Sh.M.), L1 cache, and the register file (R.F.).*

5.5 TPU - Tensor Processor Unit

Probably the most interesting (upcoming) evolution in microprocessor functionality in recent years is the introduction of the so-called TPU (Tensor Processor Unit) or MMU (Matrix Multiplication Unit). Manufacturers are gradually introducing TPUs in their CPUs, and GPUs architectures, e.g. Apple's M-Architecture (Neural Engine) and Google's mobile (and cluster) CPUs (both based on ARM architecture), NVidia has introduced the Tensor Core in its latest GPU models, Intel has introduced the AMX extensions (Advanced Matrix Extensions) in the 4th generation Xeon CPUs, and finally AMD the CDNA architecture. We must also note that these tensors (AI) accelerators are frequently implemented in FPGA or ASIC processors and are also growing the idea of analog computers[9].

In practice, TPUs consist of CPUs equipped with matrix registers (two-dimensional registers or *tiles*) and the hardware capability to directly perform the matrix Multiply-Accumulate operation (i.e., $C = AB + C$ where $A, B, C \in \mathbb{R}^{N\times N}$ and $N \in \{4, 8, 16\}$) between tiles with machine instructions, and possibly other operations on matrices.

Although TPUs are currently used almost exclusively in deep learning applications, in practice they can be used in other numerical applications as well. This makes them very attractive for the development of new scientific computing

[9] The Femtojoule Promise of Analog AI

applications.

The main advantage of CPUs equipped with TPUs is that they can outperform GPUs in applications that traditionally perform better on GPUs (allowing for significant source code simplification). On the GPU side, the use of Tensor Core features allows a peak performance that is far superior to the approach based on the SIMT paradigm on the same GPU model. The disadvantage of TPUs is that they are currently limited to the `float16` and `float32` formats. But probably the biggest disadvantage is that the technical specifications released by the vendors are currently minimal or null (for example, Apple has not yet released any technical specifications for its Neural Engine extensions) and the software support is still incomplete.

5.6 Roofline model

The *roofline model* (a.k.a. roofline analysis), first introduced in Williams et al. (2009), is a very effective visual analytical tool for comparing the measured *throughput* (a.k.a. performance) of an algorithm with its theoretical maximum performance on a given computer system. The main advantage of the roofline model is that it makes it possible to perform the *performance analysis* of an algorithm starting from the technical specifications of the computer system, namely the CPU and DRAM models, and the time and space complexity of the algorithm itself. This makes it easy to determine the execution efficiency of a program, it allows us to decide a priori which computer system (architecture) is the most appropriate to execute an algorithm, and it also allows us to easily understand which types of optimization strategies are best suited to improve the performance.

5.6.1 Operational intensity

The effective performance of an algorithm (or program) is highly dependent on the amount of input data required to achieve the result and on the size of the result itself (output) combined with the number of arithmetic and logic operations necessary to complete it.

For this motivation, we must distinguish between programs that have a high data transfer rate and those that have a low data transfer rate. Typical examples of programs that require a high data transfer rate are operations on matrices or vectors, such as transposition and multiplication of matrices, the dot product, etc. On the other hand, programs that require a low transfer rate are iterative algorithms, such as Monte Carlo methods or Taylor expansions. In practice, it is necessary to measure how many arithmetic operations are necessary to complete the algorithm given a data set with a specific size. This can be done using the idea of *operational intensity* (a.k.a. *arithmetic intensity*).

The *operational intensity* is defined as the ratio between the total number of FLOP (floating point operations) and the size of the input and output data sets (measured in bytes) necessary to complete the algorithm. Given a program P_N where the sizes of the input and output data sets and the total numbers of operations are a function of N the *operational intensity*, $\mathsf{I}_{\mathsf{OP}}(\cdot)$, is defined as:

$$\mathsf{I}_{\mathsf{OP}}(P_N) = \frac{\mathrm{F}(P_N)}{\mathrm{B}(P_N)} \qquad \left[\frac{\text{FLOP}}{\text{Byte}}\right], \tag{5.3}$$

where $\mathrm{F}(P_N)$ are the FLOPs performed by the program for an input data set of size N, and $\mathrm{B}(P_N)$ is the total size in bytes of the input and output data sets (e.g. Williams et al., 2009; Ofenbeck et al., 2014; Hennessy and Patterson, 2019; Bakhvalov et al., 2020). More generally, it can be said that the operational intensity is asymptotically proportional to the ratio between the *time complexity*, and *space complexity* (see also Sipser, 1996, chapters 7 and 8).

A program is said to be *compute-bound* if its operational intensity is large, while it is said to be *memory-bound* if its operational intensity is small. The boundary between these two classes of applications and their properties depends on the computer system and will be discussed in the following sections.

Examples of operational intensity

Operational intensity of the dot product

A dot product of two real-valued vectors of size N requires N multiplications and $N-1$ additions. Therefore, the total number of arithmetic operations is $2N-1$. The total amount of input and output data is $2N+1$, consisting of the sizes of the two input vectors and the result, which when multiplied by the size in bytes of a floating point number (e.g. 8 bytes for a `float64` number) gives the total amount of bytes transferred. Thus, the operational intensity of the dot product is:

$$\mathrm{I}_{\mathrm{OP}}(\mathrm{Dot}_N) = \frac{2N-1}{\mathrm{sizeof}(\texttt{float})(2N+1)}. \tag{5.4}$$

It is straightforward to show that for large N the operational intensity of the dot product converges asymptotically to $\mathrm{sizeof}(\texttt{float})^{-1}$. Therefore, it is, in any case, a memory-bound operation.

Operational intensity of dense square matrix multiplication

A dense matrix multiplication consists of N^2 dot products. Then the total number of operations is $N^2(2N-1)$. The total amount of data transferred is $3N^2$. Therefore, the resulting operational intensity is:

$$\mathrm{I}_{\mathrm{OP}}(\mathrm{MatMult}_N) = \frac{2N^3-N^2}{\mathrm{sizeof}(\texttt{float})\ 3N^2}, \tag{5.5}$$

where we can observe that it grows linearly with N. Thus, the multiplication of dense matrices is not necessarily a memory-bound operation, since for sufficiently large N it is a compute-bound operation.

Note about matrix multiplication: It is important to note that the operational intensity in equation (5.5) is ideal. This is because the amount of memory transferred is highly dependent on the effective implementation of the algorithm. For example, the "school book algorithm" consisting of N^2 dot products has an effective operational intensity much closer to that of the dot product than to the ideal one given in equation (5.5), and it is for this reason that the development of algorithms for dense matrix multiplication is difficult (see also Goto and Geijn, 2008).

Operational intensity of dense matrix transposition

Matrix transposition is a limit case of a mathematical operation with an operational intensity that is equal to zero, since no floating-point operations are performed. For this reason, the development of efficient matrix transposition algorithms is challenging in any case, since it is an operation in which only data transfer is performed and it is impossible to respect the locality of the data. Thus, in numerical applications, the matrix transposition must be limited to the strict minimum necessary.

5.6.2 Roofline function

The roofline $\mathsf{R_{LINE}}(\cdot)$ is a function that expresses the fact that the maximum throughput, measured in floating-point operations per second (FLOPS), of a program with a characteristic operational intensity is limited by both the bandwidth (the transfer rate between the CPU and DRAM, measured in bytes/second) and the theoretical peak performance of the CPU (measured in FLOPS), which is defined as follows:

$$\mathsf{R_{LINE}}(\mathsf{I_{OP}}(P_N)) = \min\{\mathsf{FLOPS_{MAX}},\quad \mathsf{MBW_{MAX}} \times \mathsf{I_{OP}}(P_N)\}, \tag{5.6}$$

where $\mathsf{FLOPS_{MAX}}$ is the (theoretical) peak floating point operations per second of the given CPU, $\mathsf{MBW_{MAX}}$ is the maximal memory bandwidth in bytes per second between the CPU and DRAM, and $\mathsf{I_{OP}}(P_N)$ is the operational intensity (Williams et al., 2009).

In a roofline plot, the *ridge point* is where $\mathsf{FLOPS_{MAX}} = \mathsf{MBW_{MAX}} \times \mathsf{I_{OP}}(P_N)$, which corresponds to the intersection between the horizontal line expressing the maximum CPU performance (i.e., $y = \mathsf{FLOPS_{MAX}}$) and the linear function $y = \mathsf{MBW_{MAX}} \times \mathsf{I_{OP}}(P_N)$ expressing the maximum performance constrained by the bandwidth.

A program is said to be *memory-bound* if its operational intensity is less than the abscissa of the ridge point; otherwise, it is said to be *compute-bound*. Or, in other words, if $\mathsf{I}_{\mathsf{OP}}(P_N)$ is to the right of the ridge point, its execution flow could not enter in a waiting state, while if $\mathsf{I}_{\mathsf{OP}}(P_N)$ is to the left of the ridge point, the execution flow of the program is constrained by the memory transfer rate, and it is necessary to take into account the memory bandwidth and, to prevent CPU waiting time, the memory management is the primary aspect to be considered.

5.6.3 Example of rooflines on CPU

This section aims to provide an illustrative example of roofline analysis applied to three real computer systems: a Lenovo ThinkPad laptop, a "fat" node from the USI-HPC cluster[10] (a small HPC cluster), and a Cray XC40 compute node from the Piz Daint supercomputer[11] at the Swiss National Supercomputing Center (CSCS). The main specifications of these three computer systems are listed in Table 5.2. The maximum memory bandwidth (BW) was measured using the *STREAM* benchmark[12], while the peak performances are those reported by the manufacturer (i.e., Intel[13]).

Name	CPU model	#CPU	Peak GFLOPS	BW [Gbytes/sec]
Lenovo ThinkPad	i7-9750H	1	249.6	25
fat USI-HPC	E5-2650 v3	2	2 × 368	70.5
Cray XC40	E5-2695 v4	2	2 × 604.8	97.6

Table 5.2. *Specifications of the considered computer configurations.*

[10] USI HPC Cluster
[11] Piz Daint - supercomputer
[12] STREAM: Sustainable Memory Bandwidth in High Performance Computers
[13] Compliance Metrics for Intel® Microprocessors

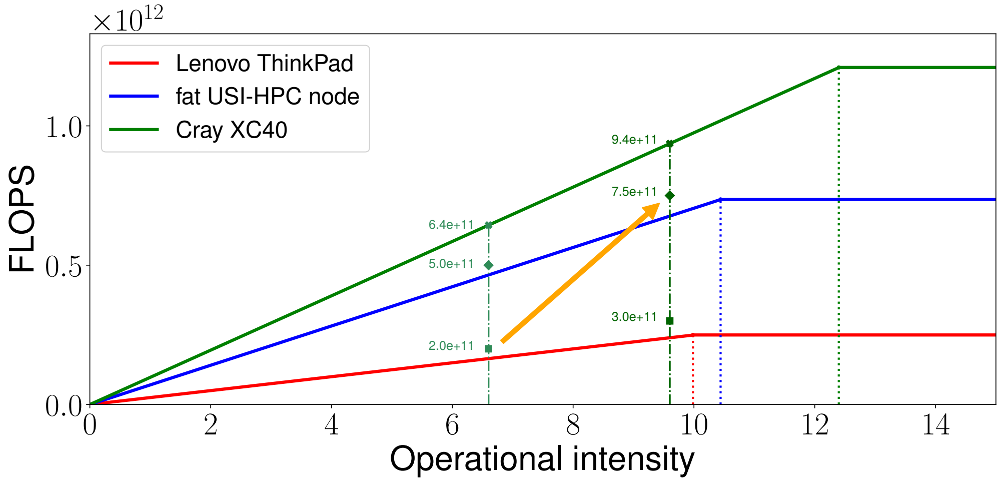

Figure 5.7. *Example roofline for three CPU configurations. A Lenovo ThinkPad laptop equipped with a Core i7-9750H (red), a "fat" compute node from the USI HPC cluster (blue), and a Cray XC40 compute node from the CSCS Piz Daint supercomputer (green). The dashed green vertical lines show two examples of possible results of a roofline analysis on a Cray XC40. The arrow indicates the ideal direction of optimization (note: the roofline plots are typically reported in logarithmic scales).*

In figure 5.7 are the plots of the rooflines of the considered computer systems. Where the vertical dotted lines are the abscissae of the ridge points (in the plot are the angles) of the three different computer systems.

The left vertical dashed line shows the performance range of a hypothetical program with an operational intensity of $\mathsf{I}_{\mathsf{OP}}(P_N) = 6.6$ on the Cray XC40. The lower left marked point shows the measure of a suboptimal implementation, with an efficiency of only 31% with respect to the theoretical peak performance for an operational intensity of 6.6, that is 640 GFLOPS (i.e., the marked point on the green roofline). A first possible optimization strategy is to improve the performance of the algorithm only by optimizing the quality of the code (e.g. by applying SIMD operations) so that the efficiency of the program increases to 78%. A second possible approach, which is not always applicable, is to modify the algorithm in order to increase the operational intensity, namely, improving or redesigning memory management. The dashed vertical line on the right is the range of performance for a program with an operational intensity of 9.6, where we immediately observe that its theoretical peak performance has grown proportionally as a function of the operational intensity itself. The arrow represents an ideal optimization strategy in which an improvement of the memory management strategy and SIMD operations are applied.

Finally, if the program is compute-bound, i.e. its operational intensity is on the right of the ridge points (dotted line), its maximum performance is limited only by the peak performance of the CPU, so that, ideally, the only necessary optimization is improving the execution flow (e.g. applying SIMD operations).

Remark. *The maximal and ideal theoretical speedup of a memory-bound program is given by the ratio of the bandwidths of the analyzed computer systems.*

5.6.4 Roofline model on GPU

On GPUs the performance is constrained by three variables: **a)** the peak performance of the GPU, **b)** the bandwidth of the internal memory (GPU memory), and **c)** the bandwidth of the PCI, which is the data transfer channel between the DRAM (host) and the GPU memory (device). The importance of PCI bandwidth in GPUs is that all input and output data must flow through it and that it greatly affects the overall efficiency of a program. Consequently, for GPUs, it is necessary to define a lower roofline for programs completely dependent on input and output data and an upper roofline for programs with no input and output (this second class of programs is clearly unrealistic).

Based on this motivation, the effective roofline of a GPU program is bounded between the lower and upper rooflines according to the ratio of the amount of input and output data (that must be transferred through the PCI port) to the amount of data generated by the program in GPU memory. Or, more generally, GPUs are suitable for programs with high operational intensity or for programs where the data transfer through the PCI is low. In Ding and Williams (2019) the roofline model on has been extended to an instruction-centric approach that is more appropriate for integer-only algorithms on GPUs.

Example: rooflines of the NVidia A100 GPU

In this example, we analyze the NVidia A100 GPU installed in the USI HPC cluster by applying the roofline model.

First, we need the technical specifications necessary to define the roofline function of the analyzed GPU, which are **1)** peak performance (measured in FLOPS), **2)** GPU memory bandwidth, and **3)** input and output (PCI) bandwidth (measured in [Gbytes/sec]). It is important to note that the input and output bandwidths of a PCI port can be different, so it is necessary to measure this quantity in both cases. In Table 5.3 we report the results for the NVidia A100.

NVidia A100		
Peak Performance (in `float64`)	9700	[GFLOPS]
Nr. of Streaming Multiprocessor	128	
GPU Memory Bandwidth	1365	[Gbytes/sec]
GPU Memory Channels	12	
PCI Bandwidth IN	8.5*	[Gbytes/sec]
PCI Bandwidth OUT	5*	[Gbytes/sec]

Table 5.3. *Technical specifications of an NVidia A100 GPU installed in the GPU nodes of the USI HPC cluster. Peak performance, number of memory channels, and streaming multiprocessor are taken from the NVidia A100 white paper*[14]*. Bandwidths were measured using benchmarks. Note(*): The low measured PCI bandwidth of the NVidia A100, which is 32 Gbytes/sec from the tech specs, is because the GPU nodes in the USI cluster have PCI 2.0 with a nominal bandwidth of 8 Gbytes/sec.*

Second, given the technical specifications, we made the roofline plot (Figure 5.8), where we plotted the upper and lower bounds of the GPU and, for comparison, the roofline of a Cray XC40 compute node. The first thing that becomes clear from the plot in Figure 5.8 is the confirmation that the GPU is convenient for compute-bound applications (i.e., applications where it is possible to minimize memory transfers over the PCI port). The second is that the GPU is never advantageous for memory-bound programs. For example, computing the dot product is always faster on CPUs because the time spent on moving the arrays is too large to justify the use of the GPU (see also the example 5.6.1) (it should be noted that by using the CUDA stream, it is possible to perform the data transfer at the same time as a kernel is running, but this does not increase the operational intensity of a program).

We must also note that in the case of the dot product (or more in general of memory-bound applications), the GPU is more advantageous when the vectors are generated directly in the GPU memory (namely, it is not necessary to use the PCI channel). However, we must point out that this speed-up is mainly a consequence of the very high bandwidth of the internal GPU memory and *not* of the parallel processing capacity (i.e. the number of Stream Multiprocessors, see also remark 5.6.3, and Sect. 5.3). For dense matrix multiplications, where

[14] NVIDIA A100 Tensor Core GPU Architecture.

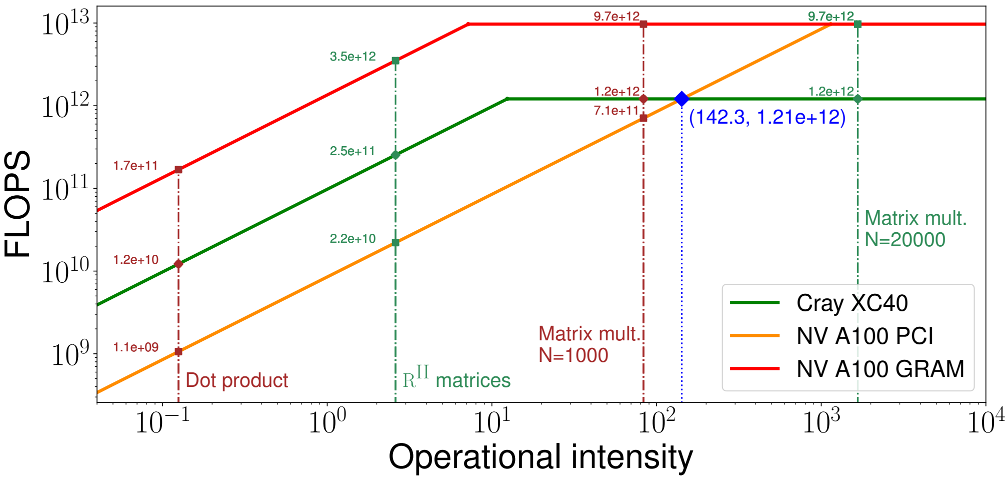

Figure 5.8. *Lower and upper bound rooflines, for the* `float64` *format, of the NVidia A100 GPU (installed in the USI HPC GPU nodes), orange and red lines, respectively, compared to a CPU-only compute node (Cray XC40), green line. Vertical dashed lines represent different operational intensities of different algorithms, from left to right a) Dot product with* $\mathrm{I}_{\mathrm{OP}}(\mathrm{Dot}_{\mathrm{N}}) = 1/8$*,* R^{II} *matrices with* $\mathrm{I}_{\mathrm{OP}}(R^{\mathrm{II}}) \approx 2.6$*, dense matrix multiplication for* $N = 1000$ *with* $\mathrm{I}_{\mathrm{OP}}(\mathrm{MatMult}_{10^3}) = 83.3$*, and dense matrix multiplication for* $N = 2 \times 10^4$ *with* $\mathrm{I}_{\mathrm{OP}}(\mathrm{MatMult}_{2\times10^4}) = 1666.6$*. The blue vertical line expresses the operational intensity above which the use of the NVidia A100 GPU is, in any case, more advantageous with respect to the Cray XC 40 (i.e.* $\mathrm{I}_{\mathrm{OP}}(P_N) > 142.3$*), which corresponds to a dense matrix multiplication with size* $N = 1716$ *(see also example 5.6.1). Finally, the area highlighted in yellow is the region where GPU programs should ideally operate.*

the operational intensity increases linearly with N, GPUs are convenient for sufficiently large matrices. In the cases discussed here, the GPU (NVidia A100) becomes suitable for operational intensities greater than 142, corresponding to a `float64` dense matrix multiplications where $N > 1716$ (if compared to a Cray XC40 compute node). It should be emphasized that implementing efficient matrix multiplication is usually difficult (e.g. Goto and Geijn, 2008) and maximum performance on both the CPU and GPU can only be achieved if the algorithm is able to adapt (or is optimized) to the technical specification of the CPU or GPU models.

From the roofline analysis of the NVidia A100, we can confirm that GPUs are generally suitable for compute-bound algorithms that operate on medium to large datasets.

For example, error backpropagation in deep neural networks (DNN) is well suited to GPUs because of the large number of iterations required to train a DNN, and, the operational intensity grow with the size of the batches from the training sets, which makes it a compute-bound procedure. Similar considerations can be made for Monte Carlo methods, which typically work very well on GPUs.

5.6.5 Application of roofline model on matrix multiplication

So far, we discussed the roofline model only from a theoretical point of view, taking into account the technical specifications of the computer system under consideration and some sample programs and their operational intensity. In this section, however, we compare the predictions made using the roofline model with the performance measured by a benchmark based on multiplication over dense matrices.

The benchmark consists of two applications: dense matrix multiplications performed using **a)** Intel MKL BLAS[15] on the CPU and **b)** NVidia cuBLAS[16] on the GPU.

With this benchmark, we can measure the impact of the memory bandwidth of the PCI port on the GPU performance, and we can validate the prediction made in Figure 5.8, according to which the GPU becomes more advantageous when the operational intensity crosses a certain threshold (which is 142 in the example considered).

The main goal of the benchmark is to perform matrix multiplication between two dense `float64` matrices stored in DRAM and write the result into a third matrix, also stored in DRAM. The tests were performed on square matrices of sizes: $N = 5 \dots 20000$. The respective operational intensities were calculated with Equation (5.5). To avoid clock instabilities, we measured the time for 100 repetitions of the matrix multiplication of the given sizes. As computer systems, we used an NVidia A100 (see Table 5.3) installed in the GPU nodes of the USI HPC cluster and a Cray XC40 (see Table 5.2) compute node from the CSCS Pitz Daint supercomputer. The results of this benchmark are shown in Figure 5.9, where it is confirmed that the CPU is more suitable for operations with low operational intensity, while GPUs become more suitable for high operational intensities. Furthermore, this benchmark shows that the theoretical prediction according to

[15] Developer Reference for Intel® oneAPI Math Kernel Library - C

[16] cuBLAS, the CUDA Basic Linear Algebra Subroutine library.

which the NVidia A100 is, in any case, more advantageous for $\mathsf{I}_{\mathsf{OP}}(P_N) > 142$ is correct (see also Figure 5.8).

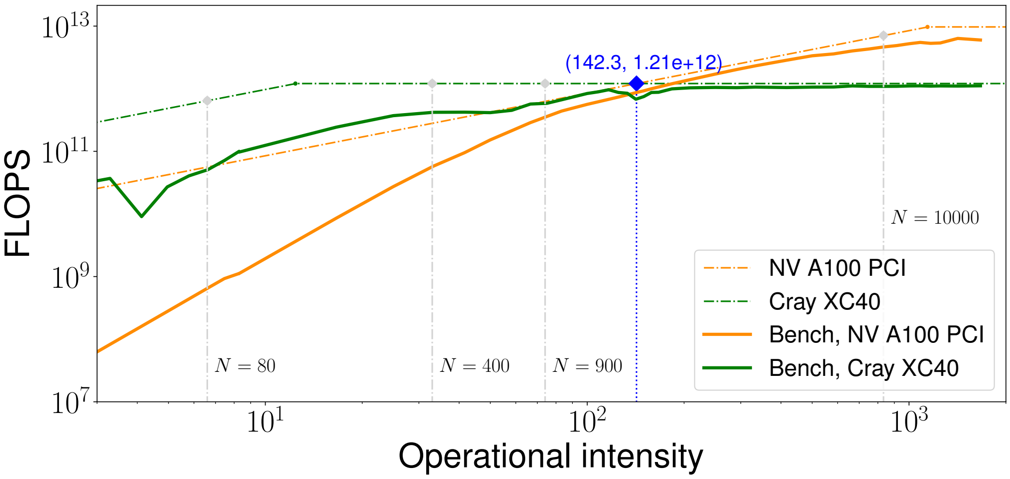

Figure 5.9. *Roofline models for Cray XC 40 and NVidia A100 (dashed green and orange lines) compared to the performance measured for dense matrix multiplication (solid green and orange lines, see also Example 5.6.1). Vertical gray lines are some matrices sizes for the corresponding operational intensities.*

5.6.6 Conclusions

The examples given in this section demonstrate the simplicity and effectiveness of the roofline model, which provides an immediate understanding of how a computer system responds to a program and which optimization strategies are most appropriate to be applied. In fact, it makes it possible to define *a priori* the theoretical peak performance of an algorithm and decide which computer architectures are best suited to execute it. We must also emphasize that with the roofline model, the decision of the computer system architecture is primarily made starting from the property and an analysis of the algorithm, and not from the generic specifications of CPUs or GPUs models.

Finally, although in this section we only discussed the cases of CPUs and GPUs, roofline analysis can easily be applied to HPC systems by considering the data transfer made by the node-to-node communication performed through MPI and the bandwidth of this channel.

5.7 Data-structures

The design of optimal data structures is probably the most important task necessary to develop algorithms that can properly exploit the features of CPUs or GPUs to maximize the throughput of the program (especially if the algorithm is memory-bound). All of the data structures we introduce in this section are designed so that all algorithms can respect the locality of the data and thus easily exploit SIMD operations.

In this section, we will first introduce the main concepts necessary to define the data structures and then present the two generic algorithms that can be used to efficiently compute the R^{II} matrices.

5.7.1 Definitions

We define the set T of all triples of combinations of *tensorial indices for polarimetry* $T = \left\{K_k, K'_k, Q_k\right\}_{k=1}^{19}$ where

$$\begin{aligned} &K \in 0 \ldots 2 \\ &K' \in 0 \ldots 2 \\ &Q \in \left[-\min\left(K, K'\right) \ldots \min\left(K, K'\right)\right], \end{aligned} \tag{5.7}$$

enumerated with the index $k \in 1 \ldots 19$ (which is constant for all two-level atomic models (see Sect. 1.4.3, and Bommier, 1997a)).

Definition 1. *In a computer program, the major index "n" of an N-dimensional array $A \in \mathbb{R}^{\times_{i=1\ldots N} D_i}$ is the index such that the values $A_{\ldots,l,n}$ and $A_{\ldots,l,n+1}$ are **adjacent in memory** (e.g. Knuth, 1968). In this chapter, the major index is always indicated as the last in the subscript (**row-major** formalism).*

The minor index "w" is the index such that the distance in memory between the elements $A_{w,l,\ldots}$ and $A_{w+1,l,\ldots}$ is maximal. Here, the minor index is always indicated as the first element of the subscript.

Definition 2. *With **flatten** we mean the operation of collapsing into 1 dimension 2 or more dimensions of an N-dimensional array. For example, if in a 3D matrix $A \in \mathbb{R}^{D_1 \times D_2 \times D_3}$ we flatten dimensions 2 and 3, we express it as a 2D matrix $\tilde{A} \in \mathbb{R}^{D_1 \times (D_2 D_3)}$. It is straightforward to note that if the matrix A is row-major, then after flattening the last two dimensions, $\tilde{A}$ is also row-major.*

Definition 3. *A **structured matrix** is an N-dimensional matrix in which the size of each row is constant in all dimensions. Thus, to access the elements of the array,* *it is sufficient to know the size of each dimension. An* ***unstructured matrix*** *is an N-dimensional matrix where the size of the rows is not constant in at least one dimension. Therefore, it is necessary to use a lookup table to access the elements.*

All multidimensional arrays described in this chapter are physically stored in 1D arrays. In the case of structured arrays, the actual index is calculated from the size of each dimension, given the set of N indices. In the case of unstructured arrays, a lookup table is used. To take full advantage of SIMD operations, the real and imaginary parts of complex values are stored in two separate arrays.

5.7.2 Data-structure for the quantities $\mathcal{R}_Q^{\text{II},KK'}$

The redistribution function $\mathcal{R}_Q^{\text{II},KK'}$ depends on the scattering angle Θ, the outgoing and incoming reduced frequencies u and u', respectively, and the *tensorial indices for polarimetry* $\{KK'Q\}$ given in equation (5.7), and it is calculated at the coordinates appearing on a frequency grid $\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$ (see also Section 4.2.4 on page 74). Therefore, the values of $\mathcal{R}_Q^{\text{II},KK'}$ are stored in a three-dimensional unstructured matrix with the indices d and p referring to the incoming and outgoing frequencies, respectively, and the major index k referring to the tensorial indices for polarimetry (we must note that the adjacency in memory is with respect to the index k).

Accordingly, we define the complex matrix $\left[\mathcal{R}_Q^{\text{II},KK'}\right]^{\Theta}_{d,p,k}$ (where the superscript indicates the dependence on the scattering angle). Since this matrix is unstructured, a lookup table must be used to convert the indices d and p into the offsets needed to access the cells in which the desired values are actually stored.

5.7.3 Data-structure for the phase scattering matrices $\mathcal{P}_Q^{KK'}$

The phase scattering matrix $\mathcal{P}_Q^{KK'} \in \mathbb{C}^{4\times 4}$ is a function of the incoming and outgoing directions and the tensorial indices for polarimetry (index k), and each element of it couples the outgoing and incoming Stokes parameters, which are denoted by the indices i and j, respectively (i.e. $i, j \in 1\ldots4$). As a result, these matrices are stored in a 3D structured row-major complex matrix with indices k, i, j, where j is the major index (so, in this case, the adjacency in memory with respect to the incoming Stokes parameter j).

Considering that the quantities $\mathcal{P}_Q^{KK'}$ have to be computed for all pairs $\boldsymbol{\Omega}'$ and $\boldsymbol{\Omega}$ of the angular grid S (see also Section 1.8.2 on page 33), it could be convenient to store these values in a 5-dimensional matrix, where the two minor indices refer to the directions from the angular grid (for the sake of simplicity, we omit to specify the indices of the directions in this discussion). Therefore, we define a 3D complex structured matrix $\left[\mathcal{P}^{KK'}_{Q}\right]^{\Omega',\Omega}_{k,i,j} \in \mathbb{C}^{19\times4\times4}$ with a constant dimension (where the superscript indicates the dependence on the incoming and outgoing directions).

5.7.4 Data structure and calculation for R^{II} matrices.

The values of R^{II} depend on the outgoing and incoming directions, frequencies, and Stokes parameters (as in the case of $\mathcal{P}^{KK'}_{Q}$, see 5.7.3). Therefore, each element of the matrix can be calculated with:

$$[R^{\mathrm{II}}]^{\Omega',\Omega}_{d,p,i,j} = \mathrm{Re}\left(\sum_{k=1}^{19}\left[\mathcal{R}^{\mathrm{II},KK'}_{Q}\right]^{\Theta}_{d,p,k}\left(\left[\mathcal{P}^{KK'}_{Q}\right]^{\Omega',\Omega}\right)^{\mathsf{T}}_{j,i,k}\right) \qquad \forall\, i,j=1\ldots4, \tag{5.8}$$

where the most convenient method to store R^{II} is to define a real unstructured matrix with indices d, p, i, and j such that the index referring to the incoming Stokes parameter (i.e., j) is the major one. Recall that, by definition, the imaginary part vanishes in the resulting matrix, so we can ignore it (see also Eq. (1.12)).

From equation (5.8) it is easy to observe that all the elements i, j of R^{II} are dot products between the values indexed by k from $\mathcal{R}^{\mathrm{II},KK'}_{Q}$ and $\mathcal{P}^{KK'}_{Q}$, so that the whole operation can be reduced to a linear operation on matrices and vectors. It is important to note that the operation in the equation (5.8) is not convenient, since, in order to respect the locality of the data, it is necessary to perform a matrix transposition (see also 5.6.1).

The transposition of $\mathcal{P}^{KK'}_{Q}$ can be avoided by reducing the operation in (5.8) to a matrix multiplication. If we *flatten* the indices referring to the frequencies d and p in the index $\tilde{\nu}$ and the indices referring to the Stokes parameters i and j in the index $\tilde{s}$, all matrices appearing in the equation (5.8) become 2-dimensional structured matrices, i.e,

$$\begin{aligned}
\left[\widetilde{R^{\mathrm{II}}}\right]^{\Omega',\Omega} &\in \mathbb{R}^{N_{\mathrm{II}}\times16}\\
\left[\widetilde{\mathcal{R}^{\mathrm{II},KK'}_{Q}}\right]^{\Theta} &\in \mathbb{C}^{N_{\mathrm{II}}\times19}\\
\left[\widetilde{\mathcal{P}^{KK'}_{Q}}\right]^{\Omega',\Omega} &\in \mathbb{C}^{19\times16},
\end{aligned} \tag{5.9}$$

where $N_{\text{II}} = \sum_{d=1}^{N_\nu} N_\beta$ is the total number of R^{II} matrices generated for a given pair of directions $\boldsymbol{\Omega}'$ and $\boldsymbol{\Omega}$, with $N_\beta = |\mathcal{G}_d^\beta|$ is the number of quadrature nodes required to perform the numerical integration on the incoming frequencies, at one outgoing frequency and directions, given the scattering angle (see also Section 4.1 on page 67).

Thus, by using the definitions 5.9, the operation in the equation (5.8) can be transformed into a complex matrix multiplication, i.e,

$$\left[\widetilde{R^{\text{II}}}\right]^{\boldsymbol{\Omega}',\boldsymbol{\Omega}} = \text{Re}\left(\left[\widetilde{\mathcal{R}_Q^{\text{II},KK'}}\right]^{\Theta} \cdot \left[\widetilde{\mathcal{P}_Q^{KK'}}\right]^{\boldsymbol{\Omega}',\boldsymbol{\Omega}}\right), \tag{5.10}$$

where its real part is calculated with:

$$\begin{aligned}\left[\widetilde{R^{\text{II}}}\right]^{\boldsymbol{\Omega}',\boldsymbol{\Omega}} =& \text{Re}\left(\left[\widetilde{\mathcal{R}_Q^{\text{II},KK'}}\right]^{\Theta}\right) \cdot \text{Re}\left(\left[\widetilde{\mathcal{P}_Q^{KK'}}\right]^{\boldsymbol{\Omega}',\boldsymbol{\Omega}}\right) \\ &- \text{Im}\left(\left[\widetilde{\mathcal{R}_Q^{\text{II},KK'}}\right]^{\Theta}\right) \cdot \text{Im}\left(\left[\widetilde{\mathcal{P}_Q^{KK'}}\right]^{\boldsymbol{\Omega}',\boldsymbol{\Omega}}\right).\end{aligned} \tag{5.11}$$

The possibility to reduce the calculation of R^{II} matrices to a matrix multiplication allows us to use highly optimized libraries (e.g. Intel MKL BLAS[17]) to build them.

Since the ordering of the data in the array used to store the resulting structured matrix $\left[\widetilde{R^{\text{II}}}\right]^{\boldsymbol{\Omega}',\boldsymbol{\Omega}}$ is equivalent to that of the unstructured one, $[R^{\text{II}}]^{\boldsymbol{\Omega}',\boldsymbol{\Omega}}$, it is not necessary to reorganize the elements of the resulting array after performing the complex matrix multiplication in the equation (5.9). In fact, to access the elements of $[R^{\text{II}}]$, we can use the indices d, p, i, j and the corresponding lookup table. In other words, this method allows us to compute the resulting elements of an unstructured data set by applying a dense matrix multiplication operator on rectangular and structured matrices.

The main disadvantage of reducing the equation (5.9) is that the use of complex matrix multiplication in BLAS (i.e. `zgemm`) (e.g. Blackford et al., 2002) does not allow us to exclude the imaginary part of the result, which by definition vanishes in the resulting R^{II} matrices. The second problem is that `zgemm` assumes that real and imaginary parts are stored in interleaved matrices (i.e., real and imaginary parts are adjacent in memory), whereas we use distinct matrices (which is generally the most efficient strategy for performing operations on complex numbers). These two problems can be solved by using the BLAS interface for real-only arrays. Thus, the reduction in equation (5.9) can be done

[17] Developer Reference for Intel® oneAPI Math Kernel Library - C

by performing two multiplications and one subtraction between real matrices (as in (5.11)) (with the disadvantage that two separate calls of a BLAS routine (i.e. `dgemm`) are required).

5.8 Operational intensity for computing a batch of R^{II} matrices

The operational intensity (see also eq. (5.3)) for building a batch with size N_{II} of R^{II} matrices is:

$$\mathsf{I}_{\mathsf{OP}}\left(R^{\text{II}}_{N_{\text{II}}}\right) = \frac{N_{\text{II}}(4\times 16\times 19)}{\text{sizeof}(\texttt{float64})[16\times 19\times 2 + N_{\text{II}}(19\times 2+16)]}, \tag{5.12}$$

where the numerator is the number of arithmetic operations required to compute one R^{II} matrix multiplied by the batch size (that is, N_{II}). The denominator is the total amount of data transferred in bytes (considering the `float64` format) needed to construct all matrices in the batch, where the first term is the size of a matrix $\left[\widetilde{\mathcal{P}^{KK'}_{Q}}\right]^{\Omega',\Omega}$ and the second is the size of the input values $\left[\widetilde{\mathcal{R}^{\text{II},KK'}_{Q}}\right]^{\Theta}$ plus the size of the output matrix, multiplied by N_{II}. It must be noted that the operational intensity of $R^{\text{II}}_{N_{\text{II}}}$ for large N_{II} asymptotically converges to:

$$\lim_{N_{\text{II}}\to\infty} \mathsf{I}_{\mathsf{OP}}\left(R^{\text{II}}_{N_{\text{II}}}\right) = \frac{76}{26} \approx 2.8. \tag{5.13}$$

Consequently, the computation of R^{II} matrices in the `float64` format is a memory-bound operation on most of the currently available computer systems (see also Section 5.6.3). And as a consequence, the *memory bandwidth* and the design of *data structures* are the primary aspects to be considered in the development of numerical methods.

This result shows also that the computation of emission vectors is primarily a massive set of small, unstructured, *memory-bound* problems that require a sequence of complex algorithms and data structures. For these reasons, we concluded that the use of CPUs combined with SIMD operations is the first choice for solving these problems (see also Section 5.6). Moreover, a highly optimized implementation on CPUs is a reference point for possible implementation on GPUs.

5.9 Direct calculation of the emission vector

Once the R^{II} matrices are ready, the subsequent step is the calculation of the emission vector at all diffused frequencies ν_d and Stokes parameters $i \in 1 \ldots 4$, and one outgoing direction $\mathbf{\Omega}$, i.e.:

$$\left[\varepsilon^{\ell,II}\right]^{\mathbf{\Omega}}_{d,i} = \sum_{m=1}^{N_\Omega} [w]^S_m \left(\sum_{p=1}^{N_\beta} [w]^\Theta_{d,p} \sum_{j=1}^{4} [R^{\text{II}}]^{\mathbf{\Omega}',\mathbf{\Omega}}_{d,p,i,j} \, [\mathbf{I}]^{\mathbf{\Omega}'}_{d,p,j} \right) \quad \forall \ \nu_d \in U, \ i = 1 \ldots 4, \tag{5.14}$$

where S is the angular quadrature grid with weights $[w]^S$. The emission vector can be stored in a structured real matrix where the number of rows is equal to the size of the frequency grid N_ν (see also Section 1.8.2 on page 33), that is:

$$\left[\varepsilon^{\ell,II}\right]^{\mathbf{\Omega}} \in \mathbb{R}^{N_\nu \times 4}. \tag{5.15}$$

The quadrature weights $[w]^\Theta$ are stored in an unstructured real matrix, where each row contains the weights necessary to perform the quadrature on the incoming frequencies, its size is equal to the size of the frequency grid $\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$, so that the lookup table necessary to access its elements is the one associated with the frequency grid. Finally, the incoming radiation field $[\mathbf{I}]^{\mathbf{\Omega}'}$ is stored in a 3D unstructured real matrix, where the major row stores the values of $\mathbf{I}$ in the four Stokes parameters. Since in the main problem the values of $\mathbf{I}$ are stored with respect to the main frequency grid U, in order to perform the calculation as described in equation 5.14, it is necessary to interpolate the incoming field for all nodes in all frequency grids $\mathrm{F}^{\text{Grid}}_{(U,\Theta,\mathbf{r})}$ of the problem.

It is straightforward to observe that the innermost summation in the equation (5.14) is a matrix-vector multiplication between the R^{II} matrices and the incoming radiation field. It is worth noting that the structure of the calculation of the equation (5.14) respects the locality of the data with respect to the data structures we defined in the preview sections. Unlike what we saw with R^{II}, in this case, it is impossible to calculate the results in batches by applying linear operators.

5.10 Explicit use of the SIMD operators in the computation of R^{II}

If we use the flattened redefinition of matrices from equation (5.9), the computation of a R^{II} matrix can be reduced to a weighted sum of 1D arrays, i.e.

$$\left[\widetilde{R^{\mathrm{II}}}\right]^{\boldsymbol{\Omega}',\boldsymbol{\Omega}}_{\tilde{\nu},1:16} = \sum_{k=1}^{19} \mathrm{Re}\left(\left[\widetilde{\mathcal{R}^{\mathrm{II},KK'}_{Q}}\right]^{\Theta}_{\tilde{\nu},k}\left[\widetilde{\mathcal{P}^{KK'}_{Q}}\right]^{\boldsymbol{\Omega}',\boldsymbol{\Omega}}_{k,1:16}\right), \tag{5.16}$$

where the notation 1 : 16 denotes a subvector that starts at index 1 and ends at 16 (also known as *slice operator*).

The sum in Equation (5.16) can be performed using the SIMD operations available in modern CPUs (e.g., AVX on the x86 architecture, see Section 5.2.2).

Since the vectorized registers of SIMD operations in the x86 architecture have a size of 4 in the case of AVX2, or 8 in the case of AVX512, and the elements of the R^{II} matrices are adjacent in memory, it is possible to execute the expression inside the parentheses in 4 or 2 steps, respectively.

The programming strategies that can be applied to exploit AVX operations on the x86 architecture are

- Intrinsic vectorization (macro assembler): The implementation is done by providing all necessary SIMD instructions and optimization strategies in the source code (using a macro assembler).

- Compiler vectorization: The code is written so that the compiler optimizer can easily recognize and vectorize the operations.

- Use of C/C++ vector extensions: The use of SIMD instructions is exploited by using a C/C++ dialect. This is a hybrid approach between the two previous strategies.

5.10.1 Introduction to the effective implementation

The main implementation for building a R^{II} matrix following the method in Equation (5.16) is in the C/C++ code in the Listing 5.1.

```
void compute_one_RII(double* R_II,
                     const double* R_KKpQ_real,
                     const double* R_KKpQ_imag,
                     const double* P_KKpQ_real,
                     const double* P_KKpQ_imag,
                     const int mu_tilde) {

    // offsets used to access the data in the arrays
    const int offset_RII = 16;
    const int offset_RK = 19;
    const int offset_PK = 16;

    for(int k = 0; k < 19; ++k){
        for(int ij = 0; ij < 16; ++ij){

        // ij is the index of the element in the slice
        // of the R_II matrix.

            R_II[mu_tilde * offset_RII + ij] +=
                R_KKpQ_real[mu_tilde * offset_RK + k] *
                        P_KKpQ_real[k * offset_PK + ij] -
                R_KKpQ_imag[mu_tilde * offset_RK + k] *
                        P_KKpQ_imag[k * offset_PK + ij];
        }
    }
}
```

Listing 5.1. *Simple C code for the calculation of one R^{II} matrix given the values of $\mathcal{P}_Q^{KK'}$ and $\mathcal{R}_Q^{\mathrm{II},KK'}$.*

In order to make a preliminary analysis of the code, we must consider that the size of a quadrature grid at u' (which depends on Θ, u, a and other physical parameters) can vary from ~ 130 to ~ 2500. As a consequence, the batch size (N_{II}) varies from ~ 13000 to ~ 25000 in the case of a frequency grid with $N_\nu \approx 100$ elements (which is the size typically used for the Ca I 4227). In general, the largest batches occur in the line core when $\Theta = \pi$ (the limit case described in Section 2.4 on page 43), while the smallest appear when Θ is close to 0.

In practical implementations, there are two possibilities: **1)** use batches as described in Section 5.7.4, where, given a pair of directions, all R^{II} matrices are calculated in a single large batch by including all outgoing frequencies contained in a frequency grid (i.e., U, see also Section 1.8.2). **2)** iterating on the outgoing frequencies in U following the structure of Algorithm 6 on page 70.

Consequently, the operational intensity for the computation of a batch varies from $\mathsf{I}_{\mathsf{OP}}\left(R^{\text{II}}_{130}\right) = 2.5$ up to $\mathsf{I}_{\mathsf{OP}}\left(R^{\text{II}}_{25000}\right) = 2.8$ with an average value of 2.6. As a result, the computation of R^{II} matrices in practical applications shows an operational intensity that is sufficiently close to its asymptotic value in the majority of cases.

Using large bathes has the advantage that all the R^{II} matrices are calculated with a unique call of a highly optimized routine. While the advantage of using small batches is that the CPU can hold all R^{II} values in the L2 cache so that they can be accessed quickly in the next stage of the algorithm.

Description of the code: In the Listing 5.1 we can observe that the arrays `R_KKpQ_real`, `R_KKpQ_imag`, `P_KKpQ_real` and `P_KKpQ_imag` are arguments of the procedure and their contents are transferred to the CPU cache and registers during the calculation by putting the execution flow of the program into a waiting state. Data transfer is generally the most time-consuming activity in memory-bound applications, and many strategies have been proposed to reduce this time. One of the most notable results is that of Goto and Geijn (2008), which suggests preemptively moving data into small blocks of memory (aka software prefetching) to ensure that it can be stored in the CPU's L2 cache before computation begins. This operation is possible for the $\left[\widetilde{\mathcal{P}^{KK'}_{Q}}\right]$ matrix, which has a constant size and can be easily copied into the L2 cache by the prefetching mechanism implemented in superscalar CPUs.

We should also recall that modern CPUs, thanks to cache prefetching, can (speculatively) transfer data and executables from slow memory (DRAM) to the CPU cache before the program uses them (e.g. Solihin, 2015). Data locality is a strategy that increases prefetch accuracy (see Equation (5.1)) because the CPU normally prefetches successive adjacent memory cells (of the arrays currently in use).

The second issue is the calculation of 1D indices by relative offsets, which are used to convert matrix indices into the pointers needed to access the memory cells used to store data. The calculation of these relative indices can be neglected from a computational point of view since most of these indices are implicit and there is no need to calculate them (for reasons, related to the vectorization, that will be clarified in the following sections).

5.10.2 Intrinsic vectorization

The first and more direct approach to vectorize the calculation of Equation (5.16) is the use of *intrinsic functions*. Namely, the implementation of methods that directly use the AVX operations of the x86 architecture (note: since during the development of this project we had access only to CPUs based on the Haswell microarchitecture and equipped with AVX2, in this and the following examples we will refer only to this architecture).

Intrinsic vectorization is based on the `immintrin.h` library[18], which provides data types and functions for vector operations using AVX instructions and registers (the library is available for GCC, Clang, Intel, and Microsoft compilers).

The advantage of using intrinsics is that code optimization is entirely at the developer's discretion, rather than relying solely on the compiler's automatic optimizer. This makes it possible to write optimal code even in situations where the compiler is unable to optimize the program. The main disadvantage of this approach is that the code is not portable to CPU architectures other than the target CPU model. In addition, the behavior of intrinsics may change on different compilers.

Intrinsic operations can be divided into five classes of operations: **a)** load and store, **b**) arithmetic, **c)** compare, **d)** convert, and **e)** shuffle.

> In AVX2, vector variables of four `float64` elements are of type `_m256d`.
>
> Intrinsic functions use the following naming convention:
> `_mm<vsize>_<intrin_op>_<suffix>`
>
> where `<vsize>` is the size of the vector register in bits (all functions in AVX2 are prefixed with `_mm256`), `<intrin_op>` is the name of the operation, in in/out operations the last letter `u` means *unaligned memory*. The first letter in `<suffix>` denotes the data model (`p` stands for packed or vector operation), and the second letter or numbers denote the data type (e.g., `d` and `s` mean `float64` and `float32`, respectively).
>
> In the AVX512, the `float64` type is `_m512d` and the functions are prefixed with `_mm512`.
>
> For further details on the naming convention, see the Intel documentation[19].

[18] Intel Intrinsics Guide

[19] Intrinsic: Naming and Usage Syntax

The two major classes of intrinsic operations are:

- The *load and store* functions are used to set the register values by copying the main memory to them, setting constant values, and copying the contents of registers to memory. The functions _mm256_load_pd and _mm256_loadu_pd for aligned and unaligned memory, respectively, and are used to load a vector of four float64 values from the DRAM.

  The functions to set constants values are _mm256_set_pd, _mm256_set1_pd, and _mm256_setzero_pd.

  Finally, the _mm256_store_pd or _mm256_storeu_pd functions are used to move the register values to the main memory.

- The main *Arithmetic* operations are _mm256_add_pd, _mm256_sub_pd, _mm256_mul_pd and _mm256_div_pd.

  The most interesting arithmetic operations are FMA (Fused Multiplying Addition), which performs operations in the form $\mathbf{r} = \pm\mathbf{a} * \mathbf{b} \pm \mathbf{c}$. In the specific problem of calculating the R^{II} matrices, FMAs are used to perform and speed up complex multiplications. In our case, where we calculate only the real part of the complex numbers, we use the function _mm256_fmsub_pd, which operates $\mathbf{r} = \mathbf{a} * \mathbf{b} - \mathbf{c}$, and then we will also discuss the use of the function _mm256_fnmadd_pd, which operates $\mathbf{r} = -\mathbf{a} * \mathbf{b} + \mathbf{c}$ (Note that immintrin.h provides a rich collection of FMA operations that can be used in a wide variety of scenarios).

The code that exploits the AVX2 functions is based on the idea of associating a vector register to each row of the matrix R^{II} and updating them iteratively.

Considering that each element (i, j) of the matrix R^{II} is a weighted sum of the corresponding elements in the matrices $\mathcal{P}_Q^{KK'}$ and the quantities $\mathcal{R}_Q^{\text{II},KK'}$ with respect to the tensorial indices for polarimetry (index k), we start by storing the real and imaginary parts of the latter quantities in two vector registers (see lines 17 and 38 of the listing 5.2). In the case of $\mathcal{P}_Q^{KK'}$ we copy the rows of the matrix into the registers, while in the case of $\mathcal{R}_Q^{\text{II},KK'}$ we broadcast the values into the corresponding registers.

The R^{II} matrix is updated by the macro: __compute_RII_row (line 16 of the 5.2 listing), which updates one row of the matrix using intrinsic operations.

We force the compiler to avoid unrolling the loop over k in order to produce a more compact and efficient assembly. In other words, the CPU does not have to spend time transferring the part of the (unrolled) program to be executed from the main memory to the L1 cache (in the experiments we performed, the unrolled code was effectively slower).

The disadvantage of this code is that, in general, compilers do not optimize intrinsics.

```
#include <immintrin.h>

void kernel_R_II_up_avx2(double *R_II_re,
                         const double *P_re,
                         const double *P_im,
                         const unsigned int P_off_k,
                         const double *rii_re,
                         const double *rii_im) {

    __m256d R_II_re_1 = _mm256_setzero_pd();
    __m256d R_II_re_2 = _mm256_setzero_pd();
    __m256d R_II_re_3 = _mm256_setzero_pd();
    __m256d R_II_re_4 = _mm256_setzero_pd();

#define __compute_RII_row(R_II_re_row, P_indx)\
    {\
        const __m256d P_re_k =\
            _mm256_loadu_pd(&P_re[off_k + P_indx]);\
        const __m256d P_im_k =\
            _mm256_loadu_pd(&P_im[off_k + P_indx]);\
\
        R_II_re_row =\
            _mm256_fmadd_pd(rii_im_k, P_im_k, R_II_re_row);\
        R_II_re_row =\
            _mm256_fnmadd_pd(rii_re_k, P_re_k, R_II_re_row);\
    }

#pragma unroll(0)
    for (unsigned int k = 0; k < 19; ++k) {
        const unsigned int off_k = P_off_k * k;

        const __m256d rii_re_k =
            _mm256_broadcast_sd(&rii_re[k]);
        const __m256d rii_im_k =
            _mm256_broadcast_sd(&rii_im[k]);

        __compute_RII_row(R_II_re_1, 0);
        __compute_RII_row(R_II_re_2, 4);
        __compute_RII_row(R_II_re_3, 8);
        __compute_RII_row(R_II_re_4, 12);
    }

    _mm256_store_pd(&R_II_re[0], R_II_re_1);
    _mm256_store_pd(&R_II_re[4], R_II_re_2);
    _mm256_store_pd(&R_II_re[8], R_II_re_3);
    _mm256_store_pd(&R_II_re[12], R_II_re_4);
}
```

Listing 5.2. *Simple C code for the calculation of one R^{II} by using the intrinsic (despite it is assumed that the memory is unaligned, for improving the performances it is recommendable to use "aligned memory" and modify the code accordingly).*

5.10.3 Compiler vectorization

This optimization strategy is based on the ability of the most advanced compilers to identify the underlying structure of the algorithms in the source code and automatically apply highly effective optimizations (GCC[20] and Intel[21] compilers are currently the most advanced). Typically, compiler optimizers can detect elementary operations between arrays, simple numerical conditions, and reductions. In addition, they can detect multiplications between complex numbers, so that the *FMA* instructions are automatically applied.

Compilers can also automatically vectorize loops, so that vectorial math operations, such as the dot product, are automatically recognized and replaced by an optimal assembly taking advantage of SIMD operations and applying the reduction algorithm.

To benefit from the vectorization capabilities of compilers, it is usually necessary to write a standard C program[22] in which we divide the algorithms into small chunks and access the elements stored in arrays using only the index of the innermost for-loop statement, allowing the optimizer to easily retrieve the main structure of the algorithm and automatically apply the most appropriate vectorization (we should observe that the most recent compilers can recognize the structure of algorithms even in poorly written code).

In the listing 5.3 we show the C code for the computation of a R^{II}, developed in a way to take advantage of the compiler's vectorization capabilities.

[20] GCC, the GNU Compiler Collection

[21] Intel compiler

[22] Support for vectorization is also provided for Fortran and C++. In the examples given in this section, we use C because the applications associated with this project are developed in C++ and the performance-critical methods are implemented in C.

```
void kernel_R_II_up_cv(double *R_II_re,
                       const double *P_re,
                       const double *P_im,
                       const unsigned int P_off_k,
                       const double *rii_re,
                       const double *rii_im){
#pragma GCC unroll(0)
    for (unsigned int k = 0; k < 19; ++k) {
        const double rii_re_k = rii_re[k];
        const double rii_im_k = rii_im[k];

        const unsigned int off_k = P_off_k * k;
        const double *P_re_k = &P_re[off_k];
        const double *P_im_k = &P_im[off_k];
#pragma GCC unroll(0)
        for (unsigned int ij = 0; ij < 16; ij++) {
            const double P_re_i = P_re_k[ij];
            const double P_im_i = P_im_k[ij];
            R_II_re[ij] += rii_re_k * P_re_i -
                            rii_im_k * P_im_i;
        }
    }
}
```

Listing 5.3. *Calculation of one R^{II} exploiting the compiler vectorization.*

In the implementation, we first assign two variables to store the real and imaginary parts of the quantities $\mathcal{R}_Q^{\text{II},KK'}$ (i.e. `rii_re_k`, and `rii_im_k`), that the compiler automatically broadcast in two vector registers of four constants values (i.e., `ymm1` and `ymm2`). Second, we define two pointers to the real and imaginary parts of the current $\mathcal{P}_Q^{KK'}$ matrix.

As explained earlier, using relative pointers allows one to use only the index `ij` to access the elements of the matrix, so that the compiler optimizer can recognize the structure of the algorithm and vectorize the innermost loop.

Given these assumptions, the compiler can generate an assembly that executes the innermost loop in four steps, i.e., the resulting assembly code updates one row of the R^{II} matrix on each iteration. In addition, the compiler uses the following strategy to update the current row:

$$\text{R\_II\_re[i:i+3]} = (\text{R\_II\_re[i:i+3]} + \text{rii\_re\_k} * \text{P\_re\_k}) - \text{rii\_im\_k} * \text{P\_im\_k},$$

where $i \in \{0, 4, 8, 12\}$. In the resulting assembly the above calculation is carried out using two FMA operations:

1) tmp = R_II_re[i:i+3] + rii_re_k ∗ P_re_i `_mm256_fmadd_pd`
2) R_II_re[i:i+3] = tmp − rii_im_k ∗ P_im_i `_mm256_fnmadd_pd`,

so that the resulting assembly code is compact and efficient. In the Listing 5.4 we report only the part of the assembly code related to the innermost loop generated by the compiler (GCC 12.1, x86 Haswell microarchitecture).

```
.L3:
    vmovupd ymm0, YMMWORD PTR [r12+rax]
    vfmadd213pd     ymm0, ymm1, YMMWORD PTR [rdi+rax]
    vfnmadd231pd    ymm0, ymm2, YMMWORD PTR [rdx+rax]
    vmovupd YMMWORD PTR [rdi+rax], ymm0
    add     rax, 32
    cmp     rax, 128
    jne     .L3
```

Listing 5.4. *GCC 12.1 - x86 Haswell generated assembly code of the innermost loop in the source code 5.3*

The disadvantage of the generated assembly is that the temporary values of the R^{II} rows are not stored in vector registers, but directly in main memory, so theoretically the intermediate result must be moved to DRAM after each iteration, putting the CPU into a wait state. It is important to remember that during the execution of the program, the R^{II} temporary values, although they refer to a DRAM address, are not stored there, but in the worst case, they are stored in the L2 (or L3) cache and moved to DRAM when the computation is finished (thanks to the speculative execution strategies of the CPU).

5.10.4 Use of C/C++ vector extension

Modern compilers support non-standard C/C++ vector extensions (dialects) that introduce vectorized operators and data types into the language. Using vector extensions allows writing programs that implicitly use vector registers without the need of using special libraries (e.g. the Intel Intrinsic); therefore, the application is portable on different CPU architectures. The main disadvantage of using the C/C++ vector extension is that it is a poorly documented and compiler-dependent feature[23](we must note that GCC, Intel and Clang (LLVM) correctly compile the source code in 5.5). Furthermore, the functionalities supported by vector extensions are very limited when compared with intrinsic operations.

The source code in 5.5 is the function used to compute one R^{II} matrix using

[23] GCC Vector Extensions

the GCC vector extensions (C/C++).

```
#include <cstddef>
void kernel_R_II_up_vec(double *R_II_re,
                        double *R_II_im,
                        const double *const P_re,
                        const double *const P_im,
                        const int P_off_id,
                        const double *const rii_re,
                        const double *const rii_im){
#define _VL_ 4
    typedef double vec
    __attribute__((vector_size(_VL_ * sizeof(double)),
                   aligned(sizeof(double))));

    const vec *const P_re_v = (const vec *)P_re;
    const vec *const P_im_v = (const vec *)P_im;
    vec row0_re = {0.0, 0.0, 0.0, 0.0},
        row1_re = {0.0, 0.0, 0.0, 0.0},
        row2_re = {0.0, 0.0, 0.0, 0.0},
        row3_re = {0.0, 0.0, 0.0, 0.0};

    for (ptrdiff_t k = 0; k < 19; ++k) {
        const ptrdiff_t base = 4 * k;
        const double rii_re_v = rii_re[k];
        const double rii_im_v = rii_im[k];

        row0_re += P_re_v[0 + base] * rii_re_v -
                   P_im_v[0 + base] * rii_im_v;
        row1_re += P_re_v[1 + base] * rii_re_v -
                   P_im_v[1 + base] * rii_im_v;
        row2_re += P_re_v[2 + base] * rii_re_v -
                   P_im_v[2 + base] * rii_im_v;
        row3_re += P_re_v[3 + base] * rii_re_v -
                   P_im_v[3 + base] * rii_im_v;
    }

    vec *R_II_re_v = (vec *)R_II_re;

    R_II_re_v[0] = row0_re;
    R_II_re_v[1] = row1_re;
    R_II_re_v[2] = row2_re;
    R_II_re_v[3] = row3_re;
}
```

Listing 5.5. *Calculation of one R^{II} exploiting the C vector extensions.*

The strategy used in the above code is equivalent to the one used in the function based on Intrinsics 5.2, Namely, we used a vector of four elements for each row of the R^{II} matrix (variables `row0_re`, ...), which the compiler automatically converts to vector registers and updates on each iteration over `k`.

The key feature of the code 5.5 is the type `vec`, which is a vector of `double` with a fixed size of four, declared by the keyword `__attribute__`. Variables `P_re_v` and `P_im_v` are arrays of `vec`, so that each element of `P_re_v` or `P_im_v` refers to a vector (`vec`) of four elements.

It should be noted that the resulting assembly (compiled with GCC 12.1) is not optimal. In fact, the updating of each row of R^{II} (`row0_re, ...`) is carried out in 3 steps:

1) tmp = P_im_v[i + base] * rii_im_v `__mm256_mul_pd`
2) tmp = P_re_v[i + base] * rii_re_v − tmp `__mm256_fmsub_pd`
3) rowi_re = rowi_re + tmp `__mm256_add_pd`,

where $\text{i} \in \{0, 1, 2, 3\}$, while if we compile the code with Clang, the compiler will use the optimal strategy, i.e., the update is performed in two steps.

```
vmulpd ymm2, ymm1, YMMWORD PTR -96[rcx].
vfmsub231pd ymm2, ymm0, YMMWORD PTR -96[rdx].
vaddpd ymm5, ymm5, ymm2
```

Listing 5.6. *GCC 12.1 - x86 Haswell assembly code for updating one row in the listing 5.5*

The advantage of using vector extensions is that each row of R^{II} is stored in a vector register and copied to the target array after the loop has finished, without putting the CPU into an unnecessary wait state or referring to a DRAM address.

5.10.5 Static analysis

Currently, there are many static analysis tools; most of the time, these applications are specialized in detecting errors in program semantics, inefficient code, and the main components of the algorithm. For example, compiler optimizers use *structural static analysis* to identify the structure of the algorithm in order to automatically apply the most appropriate optimization strategy (see also Section 5.10.3). In this section, we will focus on a static analysis tool dedicated to analyzing the performance of both source code and assembly. This second class of static analysis tools simulates the execution of a program on a given CPU (i.e., microarchitecture), and as a result, generates a near-realistic execution flow.

In this work, we use `llvm-mca`. From the official manual page[24]: “`llvm-mca` is a performance analysis tool that uses information available in LLVM (e.g. scheduling models) to statically measure the performance of machine code on a given CPU.” In other words, given the assembly, `llvm-mca` produces a rich output with many details about the efficiency of the program. Probably one of the most useful is the ability to generate the timeline of the execution flow (see also figure 5.2), which is reported in the listing 5.7.

In the timeline 5.7 it can be seen that the first iteration takes 26 cycles to complete and that the execution of the second one starts before all the operations of the first iteration have been completed, ending at cycle 36, i.e. the second and subsequent iterations take 10 clock cycles because the CPU is capable of executing two iterations of the same loop in parallel by distributing the flow of instructions on different ports (thanks to the instruction level parallelism mechanisms). This behavior shows that the program is highly optimized and able to use CPU resources correctly. In the timeline, `Index` is the number of clock cycles, `D` means the instruction is dispatched, `e` the instruction is in execution, and `E` the execution is complete. While `R` means that the execution is retired, i.e. the result of the instruction is moved to user space by the retirement unit (i.e. it is ready to be used). The retirement unit checks if the speculative and out-of-order executions were successful and, if so, releases the result. Finally, "=" and "−" are waiting states.

[24] llvm-mca - LLVM Machine Code Analyzer

```
    Timeline view:
                    0123456789
Index     0123456789          012345

[0,0]     DeER .    .    .    .    .   mov       1st cycle
[0,1]     DeeeeeeeER.    .    .    .   vbroadcastsd ...
[0,2]     DeeeeeeeER.    .    .    .   vbroadcastsd ...
[0,3]     DeE------R.    .    .    .   add      r8, ...
[0,4]     .DeeeeeeeeeeeeER    .    .   vfmadd231pd  ...
[0,5]     .DeE-----------R    .    .   add      r9, ...
[0,6]     . D====eeeeeeeeeeeeER    .   vfnmadd231pd ...
[0,7]     . DeE---------------R    .   lea      ecx ...
[0,8]     .  DeeeeeeeeeeeeE---R    .   vfmadd231pd  ...
[0,9]     .  D=====eeeeeeeeeeeeER  .   vfnmadd231pd ...
[0,10]    .   DeE---------------R  .   lea      ecx ...
[0,11]    .   D=eeeeeeeeeeeeE---R  .   vfmadd231pd  ...
[0,12]    .    D=====eeeeeeeeeeeeER.   vfnmadd231pd ...
[0,13]    .    DeE----------------R.   lea      ecx ...
[0,14]    .    DeE----------------R.   add      eax ...
[0,15]    .    .DeeeeeeeeeeeeE----R.   vfmadd231pd  ...
[0,16]    .    .D=====eeeeeeeeeeeeER   vfnmadd231pd ...
[0,17]    .    . DeE---------------R   cmp      r11 ...
[0,18]    .    . D=eE--------------R   jne  .L2     ...
[1,0]     .    . DeE---------------R    mov       2nd cycle
[1,1]     .    . DeeeeeeeE---------R    vbroadcastsd ...
[1,2]     .    .  DeeeeeeeE--------R    vbroadcastsd ...
[1,3]     .    .  DeE--------------R    add  r8,     ...
[1,4]     .    .  D===eeeeeeeeeeeeER    vfmadd231pd  ...
[1,5]     .    .   DeE-------------R    add  r9,     ...
```

Listing 5.7. *Timelime of the first and the beginning of the second iterations of the innermost loop in Listing 5.2 generated with* `llvm-mca`. *Compiler: GCC 12.2 - x86 Haswell microarchitecture*

5.11 Benchmarks and results

The purpose of this section is to use a benchmark to evaluate the effectiveness of the various strategies to calculate the R^{II} matrices introduced and discussed in this chapter. The benchmark consists of 7 different tests.

The first two benchmarks are designed to test the strategy based on matrix multiplication to build the R^{II} matrices (see Section 5.7.4). These tests take advantage of the `BLAS` routines, especially the use of the `cblas_*gemm` functions. In the first test, we used the `cblas_zgemm` to compute the multiplication between complex matrices. Thus, the calculation of equation (5.9) is carried out in its real and imaginary parts[25]. In the second test, we used two calls of the routine cblas_dgemm[26] to separately calculate the two terms of equation (5.11) and to perform the subtraction (note that: cblas_dgemm calculates $C = \alpha AB + \beta C$).

The second series of benchmarks is based on the dedicated functions we introduced in the preview sections: **1)** Intrinsic 5.10.2, **2)** Compiler vectorization 5.10.3 and **3)** Vector extension 5.10.4. In the case of the vector extension, with the objective of using all available vector registers of the CPU (16 in the Haswell microarchitecture), we considered three variants of the function. The first is the one reported in Section 5.10.4, while in the other two we modified the original function to compute the 2 and 4 R^{II} matrices at each call.

These tests were performed on the "fat" nodes of the USI HPC cluster and the Cray XC40 compute nodes of the Pitz Daint supercomputer at CSCS (see Table 5.2). For benchmarks based on BLAS routines, only the Intel MKL implementation was used, since the actual performance of other implementations is insufficient (and therefore not necessary to be reported).

The main function of the benchmark repeats 10000 times the computation of a batch of R^{II} matrices of size $N_{\text{II}} = 20000$ (the total size of the benchmark is close to the size of a real-world problem). To reproduce the realistic scenario where all CPUs of a compute node are used, we ran as many MPI processes as the number of physical processors (i.e., 36 on Cray XC40 and 20 on USI HPC). The processes are synchronized (via MPI barriers) so that all test functions are executed simultaneously. This benchmark setup reproduces the computational scenario we have implemented to solve the 3D radiative transfer problem (Benedusi et al., 2023), where there is one MPI process on each processor, each of which computes the emission vectors of a spatial point.

In tables 5.4 and 5.5 (where we report the results of the benchmarks), it can be seen that the dedicated implementations of the function achieved an efficiency of more than 90%, significantly better than that obtained using the Intel MKL implementation of BLAS. Since the application is memory bound, the bandwidth of the CPUs plays a key role in defining peak performance. As a result, the increase in FLOPS on the Cary XC40 compared to the USI HPC is not proportional to the number of processors, but to the bandwidth (which is greater on the Cary XC40, see remark: 5.6.3); this fact, based on a theoretical prediction, confirms that the

[25] Note that cblas_zgemm assumes that the real and imaginary parts of a number are contiguous in memory. Therefore, given the design of the data structure we have developed, the method of this test is not usable in the effective calculation of the R^{II} matrices.

[26] See: CBLAS gemm documentation for Intel MKL

Test [Cray XC40]	Mean GFLOPS	95% CI	Efficiency [%]
`zgemm MKL`	175.05	± 0.12	62.96
`dgemm MKL`	177.47	± 0.09	63.83
`cv`	253.89	± 0.30	91.31
`vec`	258.83	± 0.09	93.09
`vec_j2`	259.21	± 0.09	93.23
`vec_j4`	258.84	± 0.11	93.09
`avx2`	258.60	± 0.09	93.01

Table 5.4. *Benchmark results on **Cray XC40** nodes from the Daint supercomputer at the Swiss National Supercomputing Center. The sample size is 100. The `zgemm` and `dgemm` tests are those based on MKL BLAS. The test `cv` is the one based on compiler vectorization (see 5.10.3). The tests `vec`, `vec_j2` and `vec_j4` rely on vector extension (see 5.10.4) with 1, 2 and 4 R^{II} matrices computed at each step, respectively. Finally, `avx2` are the results based on the intrinsic implementation (see 5.10.2). The second column of the table shows the average performance of the compute node expressed in GFLOPS, followed by the 95% confidence interval. Finally, the last column shows the efficiency, which is the percentage ratio between the achieved performance and the roofline of the considered compute node as a function of the operational intensity of the program, i.e.* $\mathsf{I}_{\mathsf{OP}}\left(R^{\text{II}}_{2\times 10^4}\right) = 2.8$ *(see Eq. (5.13) and Sect. 5.6).*

Test [USI HPC]	Mean GFLOPS	95% CI	Efficiency [%]
`zgemm MKL`	156.36	± 0.15	62.18
`dgemm MKL`	165.66	± 0.65	65.88
`cv`	166.40	± 0.40	66.17
`vec`	240.63	± 0.84	95.69
`vec_j2`	241.55	± 0.60	96.06
`vec_j4`	235.68	± 0.57	93.72
`avx2`	237.74	± 0.67	94.54

Table 5.5. *Same as table 5.4 but reports results achieved on **fat-nodes** of the USI HPC with a sample size of 80.*

application is optimal.

We also find that methods based on intrinsic and vector extensions are generally more efficient than those based on compiler vectorization. In particular, in the table 5.5, the `cv` method shows a loss of efficiency, which could be attributed to a software problem with the compiler or to a memory problem.

From the results reported in this section, we can conclude that the main procedure used to compute the R^{II} matrices makes optimal use of CPU resources.

5.12 Alternatives strategies

In this final section, we present two alternative strategies (not yet implemented at the time of this writing) that arise from the analysis presented in this chapter and that improve the operational intensity and, as a result, effective performance of the main approach.

5.12.1 Direct calculation of the emission vector

In Equation (5.14) we show the complete structure necessary to compute the emission vector directly (i.e., without explicitly performing matrix multiplication). In terms of computational cost, this approach could be interesting because it allows us to avoid storing the values of the R^{II} matrices in an external array, so that the matrices are built and consumed directly in the registers of the CPU (without the need to reference and move the data to a memory page in DRAM). However, the main disadvantage of this approach is that it allows the calculation to be performed on one output frequency at a time (consequently, the batch size is limited to an average of $N_\beta \approx 150$).

The operational intensity (see Sect. 5.6.1) of Equation (5.14), considering the `float64` format, is

$$\mathsf{I}_{\mathrm{OP}}\left(R^{\mathrm{II}}_{N_\beta}\right) = \frac{N_\beta\,(4 \times 16 \times 19 + 16 + 12 + 4 + 4)}{\mathrm{sizeof}(\texttt{float64})\left(2 \times 16 \times 19 + N_\beta\,(19 \times 2 + 4 + 1) + 4\right)}, \quad (5.17)$$

where the numerator is the number of FLOPS required to construct a R^{II} matrix, compute the dot product between that matrix and the incoming radiation field, and finally perform the weighted sum required to compute the quadrature of the emissivity. In the denominator, there is the amount of data transferred as a function of the sizes of the input matrices, the interpolated Stokes parameters, the quadrature weights, and the resulting Stokes vector. Therefore, without the need to store the R^{II} matrices in DRAM, the total amount of data transferred is less than that considered in Section 5.10.1. For an average value of $N_\beta \approx 150$, the resulting operational intensity of the equation (5.14) is $\mathsf{I}_{\mathrm{OP}}\left(R^{\mathrm{II}}_{150}\right) = 3.3$. Therefore, in the typical realistic case, we can expect a performance improvement of $\sim 17\%$ compared to the methods introduced in section 5.10. With the limitation that the batch size N_β is a constraint imposed by the problem parameters, and on average it is small. As a consequence, due to the higher fragmentation of the execution flow, a global loss of efficiency is possible.

The listing 5.8 shows the optimized C/C++ code (based on the compiler vectorization 5.10.3) needed to compute the contribution to the emission vectors for an incoming and outgoing pair of directions (i.e. $\boldsymbol{\Omega}'$, $\boldsymbol{\Omega}$) at an outgoing frequency, given the quantities $\left[\widetilde{\mathcal{R}^{\text{II},KK'}_{Q}}\right]^{\Theta}$ and $\left[\widetilde{\mathcal{P}^{KK'}_{Q}}\right]^{\boldsymbol{\Omega}',\boldsymbol{\Omega}}$, namely, it computes only the expression in brackets of the equation (5.14).

The output array `EpsV` has a size of 4 and must be added to the final value of $\left[\varepsilon^{\ell,II}\right]$, hence the iteration over the diffused frequencies and directions must be performed with external loops (see also Algorithm 6 on page 70). As a result, the function in Listing 5.8 must be called $N_{\Omega}{}^{2} \times N_{\nu}$ (which is 921600 times if we consider the typical problem of the Ca I 4227 Å line) when calculating all emission vectors at one spatial point.

```
void kernel_R_II_up_cv_quad(double *EpsV,
                            const double *P_re,
                            const double *P_im,
                            const unsigned int P_off_k,
                            const double *rii_re,
                            const double *rii_im,
                            const double *ww,
                            const double *I,
                            unsigned int size_up) {
  EpsV[0] = 0.0;
  EpsV[1] = 0.0;
  EpsV[2] = 0.0;
  EpsV[3] = 0.0;

#pragma GCC unroll(1)
  for (unsigned int up_i = 0; up_i < size_up; up_i++) {
    double R_II_re[] = {0.0, 0.0, 0.0, 0.0,
                        0.0, 0.0, 0.0, 0.0,
                        0.0, 0.0, 0.0, 0.0,
                        0.0, 0.0, 0.0, 0.0};

    const double wl[] = {ww[up_i], ww[up_i],
                         ww[up_i], ww[up_i]};

    double EpsVl[] = {0.0, 0.0,
                      0.0, 0.0};

#pragma GCC unroll(1)
    for (unsigned int k = 0; k < 19; ++k) {
      const double rii_re_k = rii_re[k];
      const double rii_im_k = rii_im[k];

      const unsigned int off_k = P_off_k * k;
      const double *P_re_k = &P_re[off_k];
      const double *P_im_k = &P_im[off_k];

#pragma GCC unroll(1)
      for (unsigned int ij = 0; ij < 16; ij++) {
        const double P_re_i = P_re_k[ij];
        const double P_im_i = P_im_k[ij];
        R_II_re[ij] += rii_re_k * P_re_i - rii_im_k * P_im_i;
      }
    }

    const double w = ww[up_i];
    const double *I_rel = &I[up_i * 4];

    const double *R_II_re_4 = &R_II_re[4];
    const double *R_II_re_8 = &R_II_re[8];
    const double *R_II_re_12 = &R_II_re[12];

#pragma GCC unroll(1)
    for (unsigned int i = 0; i < 4; i++) {
      EpsVl[0] += (R_II_re[i] * I_rel[i]);
      EpsVl[1] += (R_II_re_4[i] * I_rel[i]);
      EpsVl[2] += (R_II_re_8[i] * I_rel[i]);
      EpsVl[3] += (R_II_re_12[i] * I_rel[i]);
    }

#pragma GCC unroll(1)
    for (unsigned int i = 0; i < 4; i++) {
      EpsV[i] += wl[i] * EpsVl[i];
    }
  }
}
```

Listing 5.8. *C/C++ code for the calculation of the emission vectors given the values of $\mathcal{P}_Q^{KK'}$, $\mathcal{R}_Q^{\mathrm{II},KK'}$, the quadrature weights* `ww` *and an array storing the interpolated values of the incoming radiative field (i.e.,* `I`*).*

5.12.2 Computation in batches of pair of directions

In Chapter 4 we introduced a method that exploits the repetitions of the scattering angles appearing in some angular grid (i.e. Cartesian product, and Lebedev's rules) to minimize the number of calculations.

This alternative method, which also exploits the repetitions of the scattering angles, is based on the fact that each flattened matrix $\left[\widetilde{\mathcal{R}_Q^{\mathrm{II},KK'}}\right]^{\Theta}$ is associated with all the phase scattering matrices $\left[\widetilde{\mathcal{P}_Q^{KK'}}\right]^{\boldsymbol{\Omega}',\boldsymbol{\Omega}}$ where the pair of directions $\boldsymbol{\Omega}'$, $\boldsymbol{\Omega}$ generate the given scattering angle Θ (see also Eq. (5.9)).

If we concatenate all these matrices in a unique matrix

$$\left[\widetilde{\mathcal{P}_Q^{KK'}}\right]^{\mathrm{Batch}} \in \mathbb{C}^{19\times(16B)},$$

where B is the batch size (i.e. the number of pairs of direction associated with a given scattering angle), and a Batch is the set of all pairs of discrete directions such that

$$\Theta = \left|\arccos\left(\boldsymbol{\Omega}_n \cdot \boldsymbol{\Omega}'_m\right)\right| \qquad \text{where :} \quad \boldsymbol{\Omega}_n, \boldsymbol{\Omega}'_m \in S\,.$$

This allows computing all the R^{II} matrices associated with the given scattering angle with unique matrix multiplication:

$$\left[\widetilde{R^{\mathrm{II}}}\right]^{\mathrm{Batch}} = \mathrm{Re}\left(\left[\widetilde{\mathcal{R}_Q^{\mathrm{II},KK'}}\right]^{\Theta} \cdot \left[\widetilde{\mathcal{P}_Q^{KK'}}\right]^{\mathrm{Batch}}\right),$$

where

$$\left[\widetilde{R^{\mathrm{II}}}\right]^{\mathrm{Batch}} \in \mathbb{R}^{N_{\mathrm{II}}\times 16B}.$$

It is straightforward to conclude that the operational intensity (derived from Eq. (5.12)) of the above matrix multiplication is:

$$\mathrm{I}_{\mathrm{OP}}\left(R^{\mathrm{II}}_{N_{\mathrm{II}},\mathrm{Batch}}\right) = \frac{N_{\mathrm{II}}(B \times 4 \times 16 \times 19)}{\mathrm{sizeof}(\texttt{float64})[B \times 16 \times 19 \times 2 + N_{\mathrm{II}}(19 \times 2 + B \times 16)]},$$

whose limit is a function of B, i.e.

$$\lim_{N_{\mathrm{II}}\to\infty} \mathrm{I}_{\mathrm{OP}}\left(R^{\mathrm{II}}_{N_{\mathrm{II}},\mathrm{Batch}}\right) = \frac{76B}{19+8B}$$

which, in turn, converge to an asymptotic upper bound operational intensity of 9.5 for $B \to \infty$. This approach could theoretically be about 3 times faster (on the same computer system, and by performing the same operations) than the main approach described above. And on some CPUs or GPUs, models could be compute-bound.

5.13 Conclusions

In this chapter, thanks to the roofline model and the study of superscalar CPUs (and GPUs), we made a comprehensive analysis of the most time-consuming function of the entire computational method for calculating the emission vectors (namely the calculation of the R^{II} matrices given the values of $\mathcal{R}_{Q}^{\text{II},KK'}$).

Thanks to the analysis of the algorithm and the available computer systems, we concluded that the first choice (and perhaps the most convenient) is to implement the algorithm on CPU since it is a memory-bound procedure and fragmented into a very large number of unstructured and small sub-problems.

The implemented strategies were validated through dedicated benchmarks, where we demonstrated a quasi-optimal efficiency.

Moreover, the methodology proposed in this Chapter (based on the roofline model) allowed us to propose two alternative strategies which can be used to further improve the performances.

Chapter 6

Quantitative analysis of the CRD approximation for the R^{III}

This chapter is based on the paper:
Assessment of the CRD approximation for the observer's frame R^{III} redistribution matrix Riva et al. (2023).

Abstract

Context: Approximated forms of the R^{II} and R^{III} redistribution matrices are frequently applied to simplify the numerical solution of the radiative transfer problem for polarized radiation, taking partial frequency redistribution (PRD) effects into account. A widely used approximation for R^{III} is to consider its expression under the assumption of complete frequency redistribution (CRD) in the observer's frame ($R^{\mathrm{III-CRD}}$). The adequacy of this approximation for modeling the intensity profiles has been firmly established. By contrast, its suitability for modeling scattering polarization signals has only been analyzed in a few studies, considering simplified settings.

Aims: In this work, we aim at quantitatively assessing the impact and the range of validity of the $R^{\mathrm{III-CRD}}$ approximation in the modeling of scattering polarization.

Methods: We first present an analytic comparison between R^{III} and $R^{\mathrm{III-CRD}}$. We then compare the results of radiative transfer calculations, out of local thermodynamic equilibrium, performed with R^{III} and $R^{\mathrm{III-CRD}}$ in realistic one-dimensional

atmospheric models. We focus on the chromospheric Ca I line at 4227 Å and on the photospheric Sr I line at 4607 Å.

Results: The $R^{\text{III-CRD}}$ approximation provides accurate results for the Ca I 4227 Å line. Only when velocities are included, some appreciable discrepancies can be found, especially for lines of sight close to the disk center. The approximation performs well also for the Sr I 4607 Å line, especially in the absence of magnetic fields or when a micro-turbulent field is included. However, some appreciable errors appear when deterministic magnetic fields or bulk velocities are considered.

Conclusions: Our results show that the $R^{\text{III-CRD}}$ approximation is suitable for modeling the scattering polarization signals of strong chromospheric lines, both in the core and wings. The approximation is suitable for photospheric lines as well, guaranteeing accurate results, especially in the presence of micro-structured magnetic fields.

6.1 Introduction

Significant scattering polarization signals are observed in several strong resonance lines of the solar spectrum, such as H I Ly-α (Kano et al., 2017), Mg II k (Rachmeler et al., 2022), Ca II K, Ca I 4227 Å, and Na I D_2 (e.g., Stenflo et al., 1980; Stenflo and Keller, 1997; Gandorfer, 2000, 2002). These signals, which are characterized by broad profiles with large amplitudes in the line wings, encode a variety of information on the thermodynamic and magnetic properties of the upper layers of the solar atmosphere (e.g., Trujillo Bueno, 2014; Trujillo Bueno et al., 2017). A correct modeling of these profiles requires solving the radiative transfer (RT) problem for polarized radiation in non-local thermodynamic equilibrium (non-LTE) conditions, taking partial frequency redistribution (PRD) effects (i.e., correlations between the frequencies of incoming and outgoing photons in scattering processes) into account (e.g., Faurobert-Scholl, 1992; Stenflo, 1994; Holzreuter et al., 2005; Belluzzi and Trujillo Bueno, 2012).

A powerful formalism to describe PRD phenomena is that of the redistribution function (e.g., Hummer, 1962; Mihalas, 1978), which is generalized to the redistribution matrix in the polarized case (e.g., Domke and Hubeny, 1988; Stenflo, 1994; Bommier, 1997a,b). For resonance lines, the redistribution matrix is given by the sum of two terms, commonly labeled as R^{II} and R^{III} according to the notation introduced by Hummer (1962). The R^{II} matrix describes scattering processes that are coherent in frequency in the atomic reference frame, while R^{III} describes scattering processes that are totally incoherent in the same frame (e.g., Bommier, 1997a,b).[1] The linear combination of R^{II} and R^{III} allows taking frequency redistribution effects due to elastic collisions with neutral perturbers into account (e.g., Bommier, 1997b). In the observer's frame, the Doppler shifts due to the thermal motions of the atoms are responsible for further frequency redistribution effects. The Doppler effect actually induces a complex coupling between the frequencies and propagation directions of the incident and scattered radiation, which makes the evaluation of both R^{II} and R^{III}, as well as the solution of the whole non-LTE RT problem, notoriously challenging from the computational standpoint. For this reason, approximate expressions of the redistribution matrices in the observer's frame, in which such coupling is loosened, have been proposed and extensively used. In this context, a common choice is to use the angle-averaged (AA) expression of R^{II}, hereafter $R^{\text{II}-\text{AA}}$ (e.g., Mihalas, 1978; Rees and Saliba, 1982; Bommier, 1997b; Leenaarts et al., 2012), and the expression of R^{III} obtained under the assumption that the scattering processes described by this matrix are totally incoherent *also* in the observer's frame, hereafter $R^{\text{III}-\text{CRD}}$ (e.g., Mihalas, 1978; Bommier, 1997b; Alsina Ballester et al., 2017). The latter is also referred to as the assumption of complete frequency redistribution (CRD) in the observer's frame.

The impact and the range of validity of the $R^{\text{II}-\text{AA}}$ approximation in the modeling of scattering polarization has been discussed by several authors (e.g., Faurobert, 1987b, 1988b; Nagendra et al., 2002; Nagendra and Sampoorna, 2011; Sampoorna et al., 2011b; Anusha and Nagendra, 2012; Sampoorna and Nagendra, 2015; Sampoorna et al., 2017; Nagendra et al., 2020; del Pino Alemán et al., 2020; Janett et al., 2021a). These studies showed that the use of $R^{\text{II}-\text{AA}}$ can introduce significant (and hardly predictable) inaccuracies in the modeling of the line-core signals, while it seems to be suited for modeling the wing lobes and their magnetic sensitivity through the magneto-optical (MO) effects. By contrast, little effort has been directed to determine the suitability of the $R^{\text{III}-\text{CRD}}$ assumption when modeling scattering polarization. Although this approximation has no true physical justification, it proved to be suitable for modeling the intensity profiles of spectral lines (e.g., Chapter 13 of Mihalas, 1978, and references therein). However, Bommier (1997b) pointed out that it may lead to appreciable errors when polarization phenomena are taken into account. In that work, the author considered a 90° scattering process of an unpolarized beam of

[1] In this work, the terms *coherent* and *totally incoherent* are used in the sense that the frequencies of the incident and scattered radiation are fully correlated and completely uncorrelated, respectively.

radiation in the presence of a weak magnetic field, and compared the polarization of the scattered radiation calculated considering the exact expression of R^{III} and the $R^{\text{III-CRD}}$ approximation, finding appreciable differences between the two cases.[2] The exact expression of R^{III} has been used in the non-LTE RT calculations with isothermal one-dimensional (1D) atmospheric models by Sampoorna et al. (2011b) and Supriya et al. (2012) for the non-magnetic case, Nagendra et al. (2002) and Supriya et al. (2013) in the weak field Hanle regime and, more recently, Sampoorna et al. (2017) in the more general Hanle-Zeeman regime. In this last paper, the authors analyzed the suitability of the $R^{\text{III-CRD}}$ approximation for modeling scattering polarization, considering an ideal spectral line. They first concluded that the use of $R^{\text{III-CRD}}$ introduces some inaccuracies, especially in the line wings, when considering optically-thin self-emitting slabs in the presence of weak magnetic fields (i.e., fields for which the ratio Γ_B between the magnetic splitting of the Zeeman sublevels and the natural width of the upper level of the considered transition is in the order of unity). Moreover, these discrepancies are much smaller when stronger magnetic fields ($\Gamma_B \approx 100$) are considered (see Fig. 1 and Sect. 4 in Sampoorna et al., 2017). However, when considering an atmospheric model with a greater optical depth, they found that the use of $R^{\text{III-CRD}}$ does not seem to produce any noticeable effect on the emergent Stokes profiles already for weak fields (see Fig. 2 in Sampoorna et al., 2017). To the best of the authors' knowledge, the aforementioned works are the only ones reporting PRD calculations of scattering polarization performed using the exact expression of R^{III}.

Taking advantage of a new solution strategy for the non-LTE RT problem for polarized radiation, tailored for including PRD effects (see Benedusi et al., 2022), we provide a quantitative analysis of the suitability of the $R^{\text{III-CRD}}$ approximation, considering more general and realistic settings than in previous studies. In particular, we show the results of non-LTE RT calculations of the scattering polarization signals of two different spectral lines (i.e., Ca I 4227 Å and Sr I 4607 Å) in 1D models of the solar atmosphere, both semi-empirical and extracted from 3D magneto-hydrodynamic (MHD) simulations, and accounting for the impact of realistic magnetic and bulk velocity fields.

Comparisons between R^{III} and $R^{\text{III-CRD}}$. Section 6.3 provides some general considerations on the role of R^{III} in the RT modeling of scattering polarization, and on the expected impact of $R^{\text{III-CRD}}$. Sections 6.4 and 6.5 report comparisons of non-LTE RT calculations of scattering polarization signals in realistic 1D settings,

[2] It must be observed that the author applied the CRD approximation after introducing in R^{III} other simplifications suited for the weak-field regime (see Sect. 4 of Bommier, 1997b).

performed with R^{III} and $R^{\text{III-CRD}}$. Finally, Section 6.6 provides remarks and conclusions.

6.2 Comparison of R^{III} and $R^{\text{III-CRD}}$

In this section, we first present an analytical comparison between the general angle-dependent expression of the R^{III} matrix (hereafter denoted as R^{III}) and its $R^{\text{III-CRD}}$ approximation. We then review the reasons why the $R^{\text{III-CRD}}$ approximation allows for a significant simplification of the problem from a computational point of view, and also provide a presentation of the challenges faced when considering R^{III}, focusing on algorithmic aspects.

6.2.1 Analytic considerations

In the formalism of the irreducible spherical tensors for polarimetry (see Chapter 5 of LL04), the R^{III} redistribution matrix in the observer's reference frame (A.35) is given by the product between the scattering phase matrix $\mathcal{P}_Q^{KK'} \in \mathbb{C}^{4\times 4}$ (A.18), which only depends on $\mathbf{r}$, $\mathbf{\Omega}$, and $\mathbf{\Omega}'$, and the redistribution function $\mathcal{R}_Q^{\text{III},KK'} \in \mathbb{C}$ (A.36), which also depends on ν and ν'. This factorization also holds for the $R^{\text{III-CRD}}$ matrix, leaving unchanged the $\mathcal{P}_Q^{KK'}$ matrix, while replacing the redistribution function $\mathcal{R}_Q^{\text{III},KK'}$ with the approximation $\mathcal{R}_Q^{\text{III-CRD},KK'}$ given by (A.41). A fundamental difference between $\mathcal{R}_Q^{\text{III},KK'}$ and $\mathcal{R}_Q^{\text{III-CRD},KK'}$ is that the former depends on the scattering angle $\Theta = \arccos(\mathbf{\Omega}\cdot\mathbf{\Omega}')$ (i.e., the angle between the directions $\mathbf{\Omega}$ and $\mathbf{\Omega}'$), while the latter does not.

To analyze the dependence on Θ, we start considering the simpler expressions that $\mathcal{R}_Q^{\text{III},KK'}$ and $\mathcal{R}_Q^{\text{III-CRD},KK'}$ assume in the absence of magnetic fields ($B = 0$). In this case, the magnetic shifts $u_{M_u M_\ell}$ (see Eq. (A.26)) are zero, and the sums over the magnetic quantum numbers M appearing in (A.36) and (A.42) can be performed analytically. If we additionally assume that there are no bulk velocities ($\mathbf{v}_b = 0$) or, without loss of generality, we evaluate the redistribution matrix in the comoving frame (see Sect. 1.8.2), the Doppler shifts u_b (see Eq. (A.26)) also vanish, and the dependence on the propagation directions $\mathbf{\Omega}$ and $\mathbf{\Omega}'$ is only through the scattering angle Θ. Expressing the functions in terms of the reduced frequencies u and u' (see Eq. (A.25)), in the limit of $B = 0$ and $\mathbf{v}_b = 0$, one finds:

$$\mathcal{R}_Q^{\text{III},KK'}(\mathbf{r},\mathbf{\Omega},\mathbf{\Omega}',u,u') \xrightarrow[B=0,\,\mathbf{v}_b=0]{} \delta_{KK'}\,\tilde{\mathcal{R}}^{\text{III},K}(\mathbf{r},\Theta,u,u'),$$

with

$$\tilde{\mathcal{R}}^{\text{III},K}(\mathbf{r},\Theta,u,u') = \big(\beta^K(\mathbf{r}) - \alpha(\mathbf{r})\big)\,W^K(J_\ell,J_u)\,\tilde{\mathcal{I}}(\mathbf{r},\Theta,u,u'). \tag{6.1}$$

The quantities β^K and α are given by (A.4) and (A.5), respectively, in the limit of no magnetic fields, while W^K is defined as (see Eq. (10.17) of LL04)

$$W^K(J_\ell, J_u) = 3\,(2J_u + 1) \begin{Bmatrix} 1 & 1 & K \\ J_u & J_u & J_\ell \end{Bmatrix}^2 ,$$

where J_ℓ and J_u are the total angular momenta of the lower and upper level, respectively, and the operator in curly brackets is the Wigner's 6-j symbol. The quantity $\tilde{\mathfrak{I}}$ takes different analytical forms depending on the value of Θ:

- if $\Theta \in (0, \pi)$ (i.e., if $\mathbf{\Omega}' \neq \mathbf{\Omega}, -\mathbf{\Omega}$)

$$\tilde{\mathfrak{I}}(\mathbf{r}, \Theta, u, u') = \frac{1}{\pi^2 \sin(\Theta)} \int \mathrm{d}y \exp\left(-y^2\right) \times H\left(\frac{a(\mathbf{r})}{\sin(\Theta)}, \frac{u + y\cos(\Theta)}{\sin(\Theta)}\right) \phi\left(a(\mathbf{r}), u' + y\right), \tag{6.2}$$

- if $\Theta = 0$ (i.e., forward scattering $\mathbf{\Omega}' = \mathbf{\Omega}$)

$$\tilde{\mathfrak{I}}(\mathbf{r}, \Theta = 0, u, u') = \frac{1}{\pi^{5/2}} \int \mathrm{d}y \exp\left(-y^2\right) \times \phi\left(a(\mathbf{r}), u + y\right) \phi\left(a(\mathbf{r}), u' + y\right), \tag{6.3}$$

- if $\Theta = \pi$ (i.e., backward scattering $\mathbf{\Omega}' = -\mathbf{\Omega}$)

$$\tilde{\mathfrak{I}}(\mathbf{r}, \Theta = \pi, u, u') = \frac{1}{\pi^{5/2}} \int \mathrm{d}y \exp\left(-y^2\right) \times \phi\left(a(\mathbf{r}), u - y\right) \phi\left(a(\mathbf{r}), u' + y\right). \tag{6.4}$$

In the previous equations, H is the Voigt profile (i.e., the real part of the Faddeeva function, see Chapter 5 of LL04) and ϕ the Lorentzian profile (i.e., the real part of the function φ defined in Eq. (A.34)). Similarly, it can be shown that

$$\mathcal{R}_Q^{\text{III-CRD},KK'}(\mathbf{r}, \mathbf{\Omega}, \mathbf{\Omega}', u, u') \xrightarrow[B=0,\, \mathrm{v}_b=0]{} \delta_{KK'} \tilde{\mathfrak{R}}^{\text{III-CRD},K}(\mathbf{r}, u, u'),$$

with

$$\tilde{\mathfrak{R}}^{\text{III-CRD},K}(\mathbf{r}, u, u') = \left(\beta^K(\mathbf{r}) - \alpha(\mathbf{r})\right) W^K(J_\ell, J_u) \times \frac{1}{\pi} H(a(\mathbf{r}), u) H(a(\mathbf{r}), u').$$

Recalling that the Voigt profile is, by definition, a convolution between a Gaussian and a Lorentzian distribution (e.g., Chapter 5 of LL04), from Eq. (6.2), it can be easily seen that for $\Theta = \pi/2$

$$\begin{aligned} \tilde{\mathcal{I}}(\mathbf{r},\Theta=\pi/2,u,u') &= \frac{1}{\pi^2} H\left(a(\mathbf{r}),u\right) \int \mathrm{d}y \exp\left(-y^2\right) \phi\left(a(\mathbf{r}),u'+y\right) \\ &= \frac{1}{\pi} H\left(a(\mathbf{r}),u\right) H\left(a(\mathbf{r}),u'\right) . \end{aligned}$$

This shows that the approximate $\tilde{\mathcal{R}}^{\text{III-CRD},K}$ function corresponds to the exact $\tilde{\mathcal{R}}^{\text{III},K}$ for $\Theta = \pi/2$, namely,

$$\tilde{\mathcal{R}}^{\text{III-CRD},K}(\mathbf{r},u,u') = \tilde{\mathcal{R}}^{\text{III},K}(\mathbf{r},\Theta=\pi/2,u,u') . \tag{6.5}$$

For scattering angles $\Theta \neq \pi/2$, the functions $\tilde{\mathcal{R}}^{\text{III},K}$ and $\tilde{\mathcal{R}}^{\text{III-CRD},K}$ are generally different, and this difference increases as Θ approaches the two limiting cases of forward and backward scattering (i.e., $\Theta = 0$ and $\Theta = \pi$, respectively). This can be clearly seen in Fig. 6.1, where $\tilde{\mathcal{R}}^{\text{III},0}$ is plotted as a function of u' for different values of u and Θ. In particular, we compare the profiles for $\Theta = 0$ and π, to that for $\Theta = \pi/2$, which, as shown above, corresponds to $\tilde{\mathcal{R}}^{\text{III-CRD},0}$. For any value of u, the function $\tilde{\mathcal{R}}^{\text{III-CRD},0}$ shows a relatively broad profile (full width at half maximum of about 2), centered at $u' = 0$. The amplitude of the peak is maximum at $u = 0$ (left panel) and quickly decreases by various orders of magnitude already at $u \approx 3$ (middle panel). The behavior of the function $\tilde{\mathcal{R}}^{\text{III},0}$ is much more complex. For $u = 0$ (left panel) and $\Theta = 0$ or π, the profile is centered at $u' = 0$ and it is much sharper and with a larger amplitude than that of $\tilde{\mathcal{R}}^{\text{III-CRD},0}$. For $u \approx 3$ (middle panel) and $\Theta = 0$ (resp. $\Theta = \pi$), the function is characterized by a broad profile, similar in amplitude and width to that of $\Theta = \pi/2$ (which is equivalent to $\tilde{\mathcal{R}}^{\text{III-CRD},0}$, see Eq. (6.5)), but with its maximum slightly shifted to positive (resp. negative) values of u'. Additionally, it shows a secondary sharp peak centered at $u' = u$ (resp. $u' = -u$). For $u \approx 9$ (right panel), in the case of $\Theta = 0$ (resp. $\Theta = \pi$), the secondary peak becomes negligible, while the main one is very similar to that of $\tilde{\mathcal{R}}^{\text{III-CRD},0}$. Noting that the dependence on K is limited to the factors β^K and W^K (see Eq. (6.1)), it is possible to state that $\tilde{\mathcal{R}}^{\text{III-CRD},K}$ and $\tilde{\mathcal{R}}^{\text{III},K}$ are *de facto* equivalent in the wings, independently of the value of Θ. By contrast, they differ in the core and near wings, where the magnitude of the discrepancies strongly depends on Θ. In the presence of a magnetic field, the redistribution function $\mathcal{R}_Q^{\text{III},KK'}$ (A.36) is given by a linear combination of the functions $\mathcal{I}_{(M_u M_\ell),(M'_u M'_\ell)}$ (A.37) – (A.39). These functions are fully analogous to $\tilde{\mathcal{I}}$ (6.2) – (6.4), the only difference being that the second and third functions in the integrand (i.e., the profiles depending

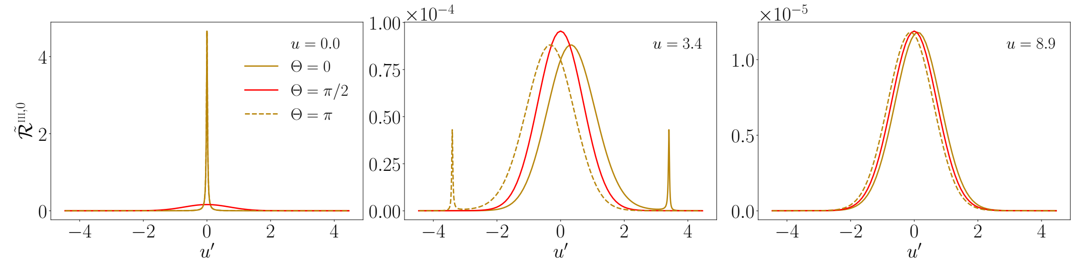

Figure 6.1. *Comparison of $\tilde{\mathcal{R}}^{\text{III},0}$ as a function of u' for three different scattering angles Θ (see legends), for $u = 0$ (left panel), $u = 3.4$ (middle panel), and $u = 8.9$ (right panel). The function is calculated considering a transition between levels with total angular momenta $J_\ell = 0$ and $J_u = 1$, and a damping parameter $a = 0.01$. The factor $(\beta^0 - \alpha)$ is calculated setting $\alpha = 0$ and using the value of β^0 corresponding to the Ca* I *4227 Å line at a height of 300 km in the FAL-C atmospheric model.*

on the scattered and incident radiation, respectively) are shifted by $u_{M_u M_\ell}$ and $u_{M'_u M'_\ell}$, respectively. It can be shown that also in the presence of magnetic fields (but still neglecting bulk velocities), the following relation holds:

$$\mathcal{R}_Q^{\text{III-CRD},KK'}(\mathbf{r}, u, u') = \mathcal{R}_Q^{\text{III},KK'}(\mathbf{r}, \Theta = \pi/2, u, u')\,.$$

When $\Theta \neq \pi/2$, $\mathcal{R}_Q^{\text{III},KK'}$ (as function of u') differs in general from $\mathcal{R}_Q^{\text{III-CRD},KK'}$. As in the unmagnetized case, the difference is marginal for large values of u (i.e., $u > 6$) independently of the magnetic field strength, while it can be very significant in the core and near wings, especially when Θ is close to 0 or π. When approaching these limit cases, the curves for $\mathcal{R}_Q^{\text{III},KK'}$ can show very sharp peaks and the contribution of the various Zeeman components, split in frequency by the magnetic field, can eventually be resolved. As an illustrative example, Fig. 6.2 shows the component $\mathcal{R}_{-2}^{\text{III},22}$ plotted as a function of u', for $u = 0.76$, $B = 30\,$G, and different values of Θ. We note that this component has a non-zero imaginary part since $Q \neq 0$. As for the unmagnetized case, the curve for $\Theta = \pi/2$, which corresponds to the CRD approximation, shows a broad profile centered at $u' = 0$. As Θ departs from $\pi/2$, the corresponding profiles show increasingly large differences from the previous one. In particular, for Θ approaching 0 (resp. π), the position of the maximum moves from $u' = 0$ towards $u' = u$ (resp. $u' = -u$) , while the width of the profile becomes sharper and the amplitude larger, in both the real and imaginary parts. As the profiles become sharper, the Zeeman components become visible, producing small lobes of opposite signs with respect to the central peak in the real part (left panels) and small substructures in the central peak in the imaginary part (right panels).

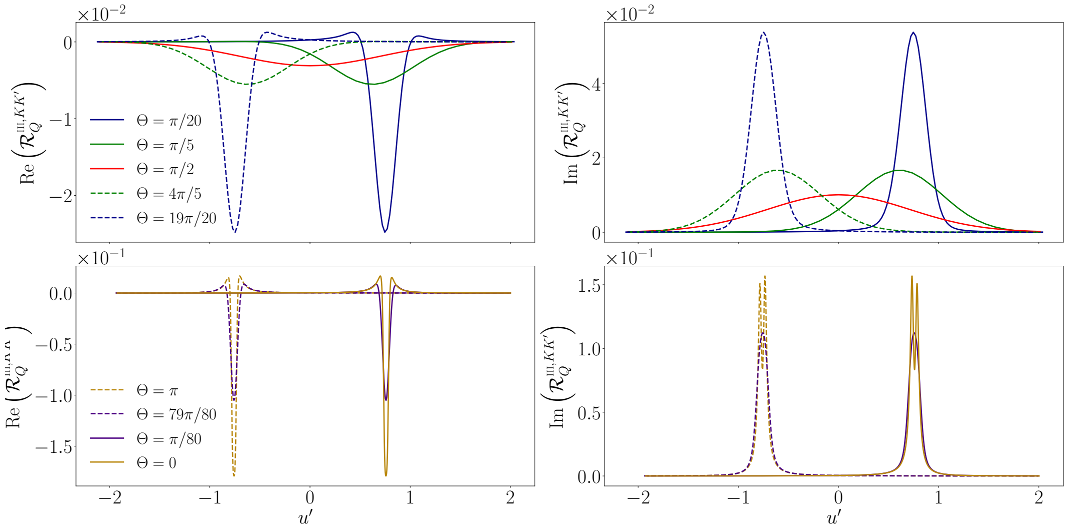

Figure 6.2. *Real (left column) and imaginary (right column) parts of* $\mathcal{R}_Q^{\text{III},KK'}$ *as a function of* u'*, for different scattering angles (see legend). We consider the component with* $K = K' = 2$ *and* $Q = -2$*. The function is evaluated at* $u = 0.76$*, including a magnetic field of 30 G. The other parameters are the same as in Fig. 6.1.*

In summary, the $R^{\text{III-CRD}}$ approximation should be suitable in the far wings of the spectral lines (see right panel of Fig. 6.1), while it can introduce some inaccuracies in the core and near wings, mainly due to scattering processes with Θ close to 0 or π, for which $R^{\text{III-CRD}}$ and R^{III} differ significantly.

6.2.2 Computational considerations on $R^{\text{III-CRD}}$

The $R^{\text{III-CRD}}$ approximation is based on the assumption that scattering processes are totally incoherent *also* in the observer's frame, which in turn implies that in a scattering event, absorption and re-emission can be treated as completely independent processes. Consistently with this picture, Eq. (A.41) shows that in $R^{\text{III-CRD}}$ the joint probability of absorbing radiation with frequency ν' and re-emitting radiation with frequency ν is simply given by the product of two generalized profiles $\Phi_Q^{KK'}$ (A.42). Following the approach discussed in Sect. 1.8.2, we evaluate the emission vector in the comoving frame, where no bulk velocities are present. In this case, the generalized profiles do not depend on the propagation directions $\mathbf{\Omega}$ and $\mathbf{\Omega}'$, and from Eqs. (A.35), (A.41), and (A.18), one finds that the emission coefficients corresponding to $R^{\rm III-CRD}$ are given by

$$
\begin{aligned}
\varepsilon_i^{\rm III-CRD}(\mathbf{r},\boldsymbol{\Omega},\nu) &= k_L(\mathbf{r}) \\
&\quad\times \int_{\mathbb{R}_+} \mathrm{d}\nu' \oint \frac{\mathrm{d}\boldsymbol{\Omega}'}{4\pi} \sum_{j=1}^{4} R_{ij}^{\rm III-CRD}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu')\, I_j(\mathbf{r},\boldsymbol{\Omega}',\nu') \\
&= \frac{k_L(\mathbf{r})}{\Delta\nu_D^2(\mathbf{r})} \sum_{K,K'=0}^{2} \sum_{Q=-K_{\rm min}}^{K_{\rm min}} \sum_{K''=|Q|}^{2J_u} \Big(\beta_Q^{K''}(\mathbf{r}) - \alpha_Q(\mathbf{r})\Big) \\
&\quad\times \sum_{Q'=-K}^{K} \sum_{Q''=-K'}^{K'} (-1)^{Q'}\, \mathcal{T}^{K'}_{Q'',i}(\boldsymbol{\Omega})\, \overline{\mathcal{D}}^{K'}_{QQ''}(\mathbf{r})\, \mathcal{D}^{K}_{QQ'}(\mathbf{r}) \\
&\quad\times \Phi_Q^{K''K'}(\mathbf{r},\nu) \int_{\mathbb{R}_+} \mathrm{d}\nu'\, \Phi_Q^{K''K}\big(\mathbf{r},\nu'\big)\, J^K_{-Q'}\big(\mathbf{r},\nu'\big)\,, \qquad (6.6)
\end{aligned}
$$

where $i = 1,\ldots,4$, $K_{\rm min} = \min(K,K')$, and $\Delta\nu_D$, $\mathcal{T}^K_{Q,i}$, and $\mathcal{D}^K_{QQ'}$ are the Doppler width (in frequency units), the polarization tensor, and the rotation matrices, respectively (see Appendix A.2 for more details). Finally, J^K_Q is the radiation field tensor, defined as (see Eq. (5.157) of LL04)

$$
J^K_Q(\mathbf{r},\nu) = \oint \frac{\mathrm{d}\boldsymbol{\Omega}}{4\pi} \sum_{j=1}^{4} \mathcal{T}^K_{Q,j}(\mathbf{r},\boldsymbol{\Omega})\, I_j(\mathbf{r},\boldsymbol{\Omega},\nu)\,.
$$

Equation (6.6) shows that the dependencies on the frequency and propagation direction of the incoming and outgoing radiation are completely decoupled. This allows the implementation of simple and fast computational algorithms: once the values of J^K_Q are obtained from the formal solution of the RT equation, it is possible to independently compute the values of $\Phi_Q^{K'K}$ for the frequencies of the incoming and outgoing radiation and combine them only during the final calculation of the emission coefficient. Using the big-O notation, the time complexity (e.g., Sipser, 1996) of the algorithm scales as $\mathrm{O}\big(N_\nu^d\big)$, where $d \in (1,2)$ grows monotonically with N_ν. For the typical size of the frequency grids needed to synthesize one (or a few) spectral lines (i.e., $N_\nu \approx 100$), d can be considered close to 1. This kind of time complexity is justified because the calculation of the generalized profile is slow (complex algorithms are required), while the subsequent combination of them is fast as it only implies products of complex numbers.

In numerical applications considering the $R^{\rm III-CRD}$ approximation, it is common practice to perform the integration over the frequencies of the incoming radiation in (6.6) using the frequency grid from the problem discretization (see Sect. 1.8.2) without further refinements, and applying the trapezoidal rule. This methodology has the advantages of a decreased *time-to-solution* (TTS) and implementation simplicity. On the other hand, it should be pointed out that the accuracy of the results can be improved by using denser and more specific grids for the frequencies of the incoming radiation without any major loss of overall performance. This is because, even if a finer frequency grid is used in (6.6), in a PRD setting the TTS for evaluating the total emissivity remains mainly dominated by the contribution of the angle-dependent R^{II} redistribution matrix.

6.2.3 Computational considerations on R^{III}

The evaluation of $\boldsymbol{\varepsilon}^{\text{III}}$ considering the exact expression of R^{III} (see equations in Appendix A.2.2) is computationally challenging for the following main reasons: **a)** it involves a 4-dimensional integration, leading to a very large number of evaluations of the integrands in Eqs. (A.37) – (A.39). This issue is otherwise known as the curse of dimensionality; **b)** the integration variables (ν', $\boldsymbol{\Omega}'$, and y) cannot be algebraically decoupled; **c)** already for simple atomic transitions, the total number of combinations of magnetic quantum numbers coupled with the tensorial polarimetric indices is high, which adds another layer of complexity to the 4 dimensions of the overall problem; **d)** the integrands include the *Faddeeva* function (Faddeeva and Terent'ev, 1961) whose evaluation requires a large TTS (e.g., Oeftiger et al., 2016); **e)** the integrands show a very complex behavior with respect to the various integration variables and parameters, thus requiring the use of very fine, unstructured, and case-dependent grids.

Our overall approach for calculating $\boldsymbol{\varepsilon}^{\text{III}}$ is to first evaluate the integral over y, followed by that over the frequencies ν', and finally, the one over the propagation direction $\boldsymbol{\Omega}'$. The analytic expressions of $\mathcal{J}$ (A.37) – (A.39) show that in the absence of bulk velocities (or if the redistribution matrix is evaluated in the comoving frame) the coupling between the propagation directions of the incoming and outgoing radiation occurs through the scattering angle Θ. The integrand in $\mathcal{J}$ shows a complex behavior with respect to the integration variable y, especially when the scattering angle approaches 0 and π. In these cases, the integrand becomes close to a Lorentz distribution or its square. These functions are not easy to numerically integrate due to the presence of sharp peaks and extended wings. Indeed, the convergence rate for the numerical quadrature of the Lorentz distribution (and its square) is generally slow. Furthermore, the computation of a single $\mathcal{J}$ (see (A.37)) can require up to thousands of evaluations of the Faddeeva function (accounting for more than 70% of the TTS). In order to perform the quadratures over y, we applied an adaptive quadrature method based on the Gauss-Kronrod approach (e.g., Kronrod, 1965; Piessens et al., 2012). The advantage of this method is that it is capable to infer automatically the behavior of the integrand by achieving very high accuracy with a relatively low number of function evaluations.

In general, the time complexity for the computation of $\boldsymbol{\varepsilon}^{\text{III}}$ is $\mathrm{O}\big(N_\Omega^2 N_\nu^3\big)$. The cubic contribution given by the number of frequency grid points is due to the fourth dimension induced by y. It must be noted that the number of grid points needed to adequately perform the quadrature of the integral over y is generally larger than N_ν.

For the angular integral in (1.4), it is convenient to apply a quadrature rule characterized by a regular angular grid, because in this case, the effective number of different scattering angles is significantly lower than the total pairs of directions. The results reported in the next sections were obtained with the quadrature method described in Sect. 1.8.2, with 12 Gauss-Legendre inclinations and 8 equally spaced azimuths. In this case, the total number of scattering angles is limited to 200. In our algorithm, the quantities $\mathcal{J}$ are pre-computed for the whole set of different scattering angles corresponding to the chosen angular grid, thus avoiding repeating the calculation of $\mathcal{J}$, which is rather expensive, for different pairs of directions having the same scattering angle.

The quantity $\mathcal{J}$ depends on two pairs of magnetic quantum numbers, (M_u, M_ℓ) and (M_u', M_ℓ'), where the labels u and ℓ indicate the upper and lower level, respectively. Equation (A.36) shows that the expression of R^{III} actually includes four quantities $\mathcal{J}$, which differ for the values of the magnetic quantum numbers and are coupled inside six nested loops over such quantum numbers. It can be easily realized that in such loops several tuples of magnetic sublevels are repeated, which suggests the opportunity to increase the efficiency of the algorithm by pre-computing the quantities $\mathcal{J}$ for all possible combinations of magnetic quantum numbers, thus avoiding to re-calculate the same quantity $\mathcal{J}$ various times. The pre-computation of $\mathcal{J}$ as described above leads to a drastic reduction of the total number of calculations needed to compute $\boldsymbol{\varepsilon}^{\text{III}}$, which otherwise would be very significant due to the extra dimension in the integration. The pre-computed values of $\mathcal{J}$ are stored out-of-core (e.g. on disk) because they require a large footprint, and are accessed during the calculation of $\boldsymbol{\varepsilon}^{\text{III}}$ through a system of look-up tables. It must be observed that the quantity $\mathcal{J}$ and, consequently, the grids used for performing the numerical integration, also depends on the spatial point through the damping parameter a, the magnetic field and the Doppler width $\Delta\nu_D$. Our strategy thus calls for relatively large data storage capabilities that, however, remain manageable in the case of 1D applications.

6.3 Impact of R^{III} on spectral lines formation

The R^{III} redistribution matrix describes scattering processes during which the atom undergoes elastic collisions with neutral perturbers (mainly hydrogen and helium atoms in the solar atmosphere) that completely relax any correlation between the frequencies of the incident and scattered radiation, thus making the scattering totally incoherent. Informally speaking, due to such collisions, the atom does not keep any *memory* of the frequency of the incident photon. On the contrary, the R^{II} redistribution matrix describes scattering processes in which the atom is not subject to any elastic collisions, so that the frequencies of the incident and scattered radiation remain fully correlated (coherent scattering). A quantitative estimate of the relative weight of R^{III} with respect to R^{II} is provided by the branching ratio for R^{II} (see (A.15)) in the absence of magnetic fields (a.k.a. coherence fraction):

$$\tilde{\alpha}(\mathbf{r}) = \frac{A_{u\ell} + C_{u\ell}(\mathbf{r})}{A_{u\ell} + C_{u\ell}(\mathbf{r}) + Q_{\text{el}}(\mathbf{r})}\,,$$

where $A_{u\ell}$ is the Einstein coefficient for spontaneous emission, $C_{u\ell}$ is the rate of inelastic de-exciting collisions, and Q_{el} is the rate of elastic collisions with neutral perturbers. A value of $\tilde{\alpha}$ close to unity (corresponding to a very low rate of elastic collisions compared to the rates for spontaneous emission and collisional de-excitation) means that R^{II} dominates over R^{III}, while a value of $\tilde{\alpha}$ close to zero (corresponding to a very high value of Q_{el} compared to $A_{u\ell}$ and $C_{u\ell}$) means instead that R^{III} dominates with respect to R^{II}.

The impact of R^{III} is thus expected to be marginal in the core of strong spectral lines (i.e., lines showing broad intensity profiles with extended wings). Indeed, their line-core radiation generally originates from the upper layers of the solar atmosphere, where the number density of neutral perturbers (and thus the rate of elastic collisions) is relatively low. The relative weight of R^{III} can, however, become significant in the wings of such lines, as they usually form much lower in the atmosphere. On the other hand, it must be observed that the profiles entering the definition of R^{III} are all centered around the line-center frequency, and therefore the net contribution of this redistribution matrix to the emissivity in the line wings is generally marginal with respect to that of R^{II}. This can be seen as an analytical proof of the well-known fact that scattering processes are mainly coherent in the line wings (e.g., Mihalas, 1978). Consistently with this picture, Alsina Ballester et al. (2022) showed that in the wings of strong resonance lines, the contribution of R^{III} needs to be taken into account in order not to overestimate the weight of R^{II}, but its net contribution to scattering polarization is fully negligible. These considerations suggest that the exact analytical form of R^{III} should not be crucial for modeling the core and far wings of strong lines, and that the $R^{\text{III-CRD}}$ approximation should therefore be suitable in these spectral ranges. The most critical regime is that of the near wings, where strong resonance lines may show very significant scattering polarization signals. There, R^{III} may bring a non-negligible contribution, and it is harder to estimate a-priori the suitability of the $R^{\text{III-CRD}}$ approximation.

The relative weight of R^{III} is also non-negligible in the case of spectral lines forming in the deeper layers of the solar atmosphere (photosphere), where the number density of colliders is significant. On the other hand, these lines are generally weak in the intensity spectrum, showing narrow absorption profiles with a Doppler core and no wings. Since Doppler redistribution is generally very efficient in the line-core, the limit of CRD (i.e., to assume that all scattering processes are totally incoherent) has always been considered a good approximation for modeling both the intensity (e.g., Mihalas, 1978) and scattering polarization profiles of these lines (e.g., LL04).

In order to quantitatively verify these considerations, and assess the suitability of the $R^{\text{III-CRD}}$ approximation for modeling scattering polarization, we will model the intensity and polarization of two different spectral lines, namely a strong spectral line with extended wings forming in the upper layers of the solar atmosphere, and a weaker line forming deeper in the atmosphere. Excellent examples for these two typologies of spectral lines are, respectively, the Ca I line at 4227 Å and the Sr I line at 4607 Å. Both lines result from a resonant transition between the ground level of the considered atomic species, which in both cases has total angular momentum $J_\ell = 0$, and an excited level with $J_u = 1$. Both of them show conspicuous scattering polarization signals, which can be suitably modeled considering a two-level atom with an unpolarized and infinitely-sharp lower level. Figure 6.3 shows the variation of the coherence fraction $\tilde{\alpha}$ with height in the 1D semi-empirical model C of Fontenla et al. (1993, hereafter FAL-C) for the two considered spectral lines. In the same figure, we also plot the height at which the optical depth τ, in the frequency intervals of the considered spectral lines, is unity. It can be shown (e.g. Mihalas, 1978) that this height provides an approximate estimate of the atmospheric region from which the emergent radiation originates (formation height). We recall that the optical depth at the frequency ν along direction $\mathbf{\Omega}$ is defined as

$$\tau(s, \mathbf{\Omega}, \nu) = \int_0^s \eta_1(x, \mathbf{\Omega}, \nu)\, \mathrm{d}x \,, \tag{6.7}$$

where s is the spatial coordinate along direction $\mathbf{\Omega}$ (measured with respect to an arbitrary initial point), and η_1 is the absorption coefficient for the intensity

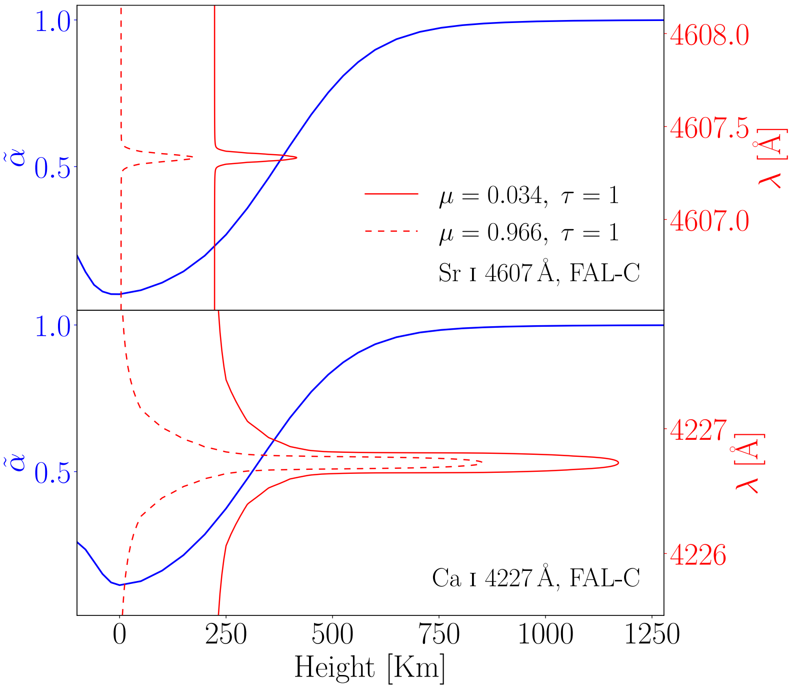

Figure 6.3. *Height variation of $\tilde{\alpha}$ (blue line) for Sr* I *4607 (top panel) and Ca* I *4227 (bottom panel) in the FAL-C model. The red and dashed red isolines show the height where the optical depth as a function of wavelength is unity (i.e. $\tau = 1$) for $\mu = 0.034$ and $\mu = 0.966$, respectively.*

(see also Janett et al., 2017). For calculating the formation height, the direction $\mathbf{\Omega}$ must point inwards in the atmosphere, and the initial point $s = 0$ is taken at the upper boundary. Since the absorption coefficient is largest at the line center and decreases moving to the wings, it can be immediately seen that the formation height is highest in the line core and decreases moving to the wings. From Eq. (6.7) it is also clear that, for a given frequency, the formation height is higher for a line of sight (LOS) close to the edge of the solar disk (limb) than for one at the disk center. Clearly, the formation height is higher for strong lines which have a large absorption coefficient (e.g., because the number density of atoms in the lower level of the transition is particularly large in the solar atmosphere) than for weak ones.

6.3.1 Ca I 4227 Å line

In the intensity spectrum, the Ca I 4227 Å line shows a very broad and deep absorption profile (e.g. Gandorfer, 2002) with an *equivalent width* of 1426 mÅ (Moore et al., 1966), i.e. one of the strongest spectral lines in the visible part of the solar spectrum. When observed in quiet regions close to the limb, this line shows a large scattering polarization signal with a sharp peak in the line core and broad lobes in the wings (e.g. Gandorfer, 2002). This signal, with its peculiar triplet-peak structure, has been extensively observed and modeled in the past (e.g., Stenflo et al., 1980; Faurobert-Scholl, 1992; Bianda et al., 2003; Anusha et al., 2011; Supriya et al., 2014; Carlin and Bianda, 2017; Alsina Ballester et al., 2018; Janett et al., 2021a; Capozzi et al., 2020; Jaume Bestard et al., 2021a, and references therein). In particular, it was clearly established that its broad wing lobes are produced by coherent scattering processes with PRD effects. We recall that the line-core peak is sensitive to the presence of magnetic fields via the Hanle effect. The Hanle critical field (i.e., the magnetic field strength for which the sensitivity to the Hanle effect is maximum) is $B_c \approx 25$ G. The wing lobes are sensitive to longitudinal magnetic fields of similar strength via the magneto-optical (MO) effects (Alsina Ballester et al., 2018).

The lower panel of Fig. 6.3 shows that in the FAL-C atmospheric model, the core of the Ca I 4227 Å line forms above 800 km (low chromosphere). At these heights, the number density of neutral perturbers is very low, the coherence fraction is thus very close to unity, and R^{II} dominates with respect to R^{III}. On the other hand, as we move from the core to the wings, the formation height quickly decreases to photospheric levels. At these heights, the coherence fraction is much lower than one and the weight of R^{III} is significant. On the other hand, for the reasons discussed above, its net impact in the line wings is expected to be relatively low. The most critical region is that of the near wings, where the Ca I 4227 Å line shows strong scattering polarization signals, and the net contribution from R^{III} can be non-negligible. The suitability of the $R^{\mathrm{III-CRD}}$ approximation in this spectral region can only be assessed numerically, and it will be analyzed in Sect. 6.4.

6.3.2 Sr I 4607 Å line

The Sr I 4607 Å line is a rather weak absorption line in the intensity spectrum, with an equivalent width of 36 mÅ only (see Moore et al., 1966). Nonetheless, this line shows a prominent scattering polarization signal at the solar limb, characterized by a sharp profile (e.g. Gandorfer, 2000). This signal has been also extensively observed and modeled in the past, especially in order to investigate, via the Hanle effect, the small-scale, unresolved magnetic fields that permeate the quiet solar photosphere (e.g., Stenflo and Keller, 1997; Trujillo Bueno et al., 2004; del Pino Alemán et al., 2018; del Pino Alemán and Trujillo Bueno, 2021; Dhara et al., 2019; Zeuner et al., 2020, 2022). The Hanle critical field for this line is $B_c \approx 23$ G. The limit of CRD has always been considered suited for modeling both the intensity and scattering polarization profiles of this line.

The upper panel of Fig. 6.3 shows that in the FAL-C atmospheric model, the core of this line forms in the photosphere, below 500 km. The curve for the coherence fraction shows that the weight of R^{III} is, as expected, significant at these heights. In the next section, we will explore the suitability of the $R^{\text{III}-\text{CRD}}$ approximation in the modeling of the scattering polarization signal of this line through full PRD RT calculations in the presence of both deterministic and non-deterministic (e.g. Stenflo, 1982, 1994; Landi Degl'Innocenti and Landolfi, 2004) magnetic fields. For the latter, we will consider *unimodal micro-structured isotropic* (MSI) magnetic fields, namely magnetic fields with a given strength and an orientation that changes on scales below the mean free path of photons, uniformly distributed over all directions. The expressions of RT quantities in the presence of MSI magnetic fields are exposed in Appendix A.5 (see also Sect. 4 of Alsina Ballester et al., 2017).

6.4 Numerical results: FAL-C atmospheric model

In this section and in the next one, we present the numerical results[3] of non-LTE RT calculations of the scattering polarization profiles of the Ca I 4227 Å and Sr I 4607 Å lines, performed with the numerical solution strategy described in Sect. 1.8.2. All calculations are performed using the general (angle-dependent) expression of the R^{II} redistribution matrix, while considering both the exact form of the R^{III} matrix and its $R^{\text{III}-\text{CRD}}$ approximation. The lower level population, which we keep fixed in the problem (see Sect. 1.8.1), is pre-computed with the RH code (Uitenbroek, 2001), which solves the nonlinear unpolarized non-LTE RT problem. For the Ca I 4227 Å line, we run RH using a model for calcium composed of 25 levels, including 5 levels of Ca II and the ground level of Ca III. For the Sr I 4607 Å line, we considered a model composed of 34 levels, including the ground

[3] All calculations involving R^{III} (in its angle-dependent formulation) were performed on the Piz Daint supercomputer of the Swiss National Supercomputing Center (CSCS) on Cray XC40 nodes (https://www.cscs.ch/computers/piz-daint). An XC40 compute node is equipped with two Intel® Xeon® E5-2695 v4 microprocessors at 2.10GHz (2x18 cores, 64/128 GB RAM).

level of Sr II. These preliminary computations also provided the rates for elastic and inelastic collisions, as well as the continuum quantities.

For a quantitative comparison between the emergent Stokes profiles obtained using R^{III} and $R^{\mathrm{III-CRD}}$, we plot the error defined by

$$\mathrm{Error}\,(\mathbf{a},\,\mathbf{b}) = \frac{\left|\mathbf{a}\,(\max\mathbf{a})^{-1} - \mathbf{b}\,(\max\mathbf{a})^{-1}\right|}{1 + |\mathbf{a}\,(\max\mathbf{a})^{-1}|}, \tag{6.8}$$

where **a** and **b** represent the values of a given Stokes parameter of the emergent radiation, for a given direction and all considered wavelengths, obtained considering R^{III} and $R^{\mathrm{III-CRD}}$, respectively. The maximum is calculated with respect to wavelength, over the considered interval. The error definition in Eq. (6.8) does not correspond to the standard relative error and it was introduced to prevent amplifying the discrepancies where **a** and **b** are close to zero and consequently to the numerical noise. Where the signals **a** and **b** are relevant, this definition provides a value that is smaller, by a factor of two in the worst case (i.e., where **a** takes its maximum value), than the usual relative error. This definition is thus justified as it provides the correct order of magnitude of the error where the signal is relevant while damping it when the signal becomes negligible.

In this section, we show calculations performed in the FAL-C atmospheric model (70 height nodes), in the presence of constant (i.e., height-independent) deterministic magnetic fields (i.e., magnetic fields having a well-defined strength and orientation at each spatial point), in the absence of bulk velocity fields. The LOS towards the observer is taken on the $x-z$ plane of the considered reference system (see Fig. 1.4) and is specified by $\mu = \cos\theta$, with θ the inclination with respect to the vertical. Typically, we present the emergent Stokes profiles for two specific directions: $\mu = 0.034$, which represents radiation coming from the solar limb (nearly horizontal LOS), and $\mu = 0.996$, which represents radiation coming from near the center of the solar disk (nearly vertical LOS).

6.4.1 Ca I 4227 Å line

We first consider the modeling of the Stokes profiles of the Ca I 4227 Å line, which is discretized with $N_\nu = 99$ unevenly spaced nodes. The characteristics of this spectral line are adequately reproduced by our calculations. The triple-peak structure in Q/I is evident in the non-magnetized model shown in Figure 6.4, while the magnetized case shown in Figure 6.6 displays a clear depolarization of the core as a result of the Hanle effect and a mild depolarization of the lobes due to the magneto-optical effects.

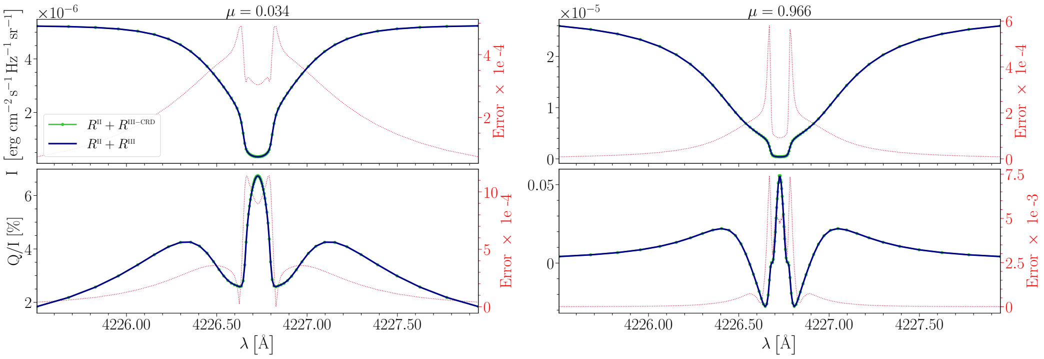

Figure 6.4. *Emergent Stokes I (top panels) and Q/I (bottom panels) profiles of the Ca* I *4227 Å line at $\mu = 0.034$ (left panels) and $\mu = 0.966$ (right panels) calculated in the FAL-C atmospheric model in the absence of magnetic fields. Calculations take into account PRD effects considering the exact expression of R^{III} (blue lines) and the $R^{\text{III-CRD}}$ approximation (green marked lines). The reference direction for positive Q is taken parallel to the limb. The red dotted lines report the error between R^{III} and $R^{\text{III-CRD}}$ calculations, given by Eq.* (6.8)*.*

Figure 6.4 shows the Stokes I and Q/I profiles for a non-magnetic scenario, where we observe no significant difference between R^{III} and $R^{\text{III-CRD}}$ calculations, with an error that is always smaller than 7.5×10^{-3}. In the absence of magnetic fields, the U/I and V/I profiles vanish and are consequently not shown.

Figure 6.5 shows the contributions from R^{II}, R^{III}, and $R^{\text{III-CRD}}$ to $|\varepsilon_I|$ and $|\varepsilon_Q|$ for the non-magnetized case as a function of height. For the sake of completeness, the figure also shows the contributions from continuum and line thermal emissivities to Stokes I and from continuum to Stokes Q (see blue lines). In this figure, we consider $\mu = 0.17$ and $\lambda = 4227.1$ Å, which corresponds to the wavelength at the maximum of the red Q/I lobe in Figure 6.4 (left bottom panel). The top panel of Fig. 6.5 shows that the contributions to ε_I from R^{III} and $R^{\text{III-CRD}}$ practically coincide at all heights, and that at the height where the optical depth is unity, they are very similar to that from R^{II}. By contrast, in the bottom panel, we see that the contribution to ε_Q of R^{II} dominates over that of R^{III} at all heights, and it can be thus considered the only relevant contributor to the formation of the Q/I wing lobe. In this case, the contribution from $R^{\text{III-CRD}}$ is different from that of R^{III}, but it remains negligible with respect to that of R^{II}. This explains why the computations of Fig. 6.4 do not show any appreciable differences between R^{III} and $R^{\text{III-CRD}}$ in the line wings.

Finally, Figure 6.6 displays all the Stokes profiles in the presence of a height-

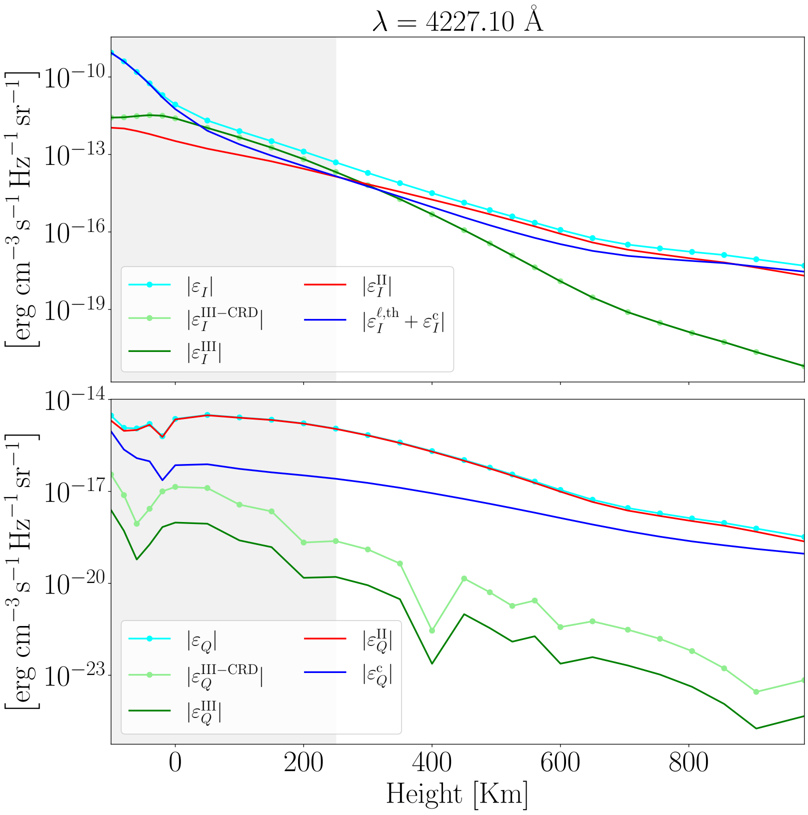

Figure 6.5. *Various contributions (see legend) to the emission coefficients for the Stokes parameters I (top panel) and Q (bottom panel) as a function of height in the FAL-C model, in the absence of magnetic fields. The emission coefficients are evaluated at the wavelength $\lambda = 4227.1$ Å, corresponding to the maximum of the Q/I lobe in the red wing of the line, for a LOS with $\mu = 0.17$. The shaded area in the panels highlights the atmospheric region where the optical depth at this wavelength and LOS is greater than 1. ε_X (where $X = I$ or Q) is the total emissivity, $\varepsilon_X^{\rm II}$, $\varepsilon_X^{\rm III}$, and $\varepsilon_X^{\rm III-CRD}$ are the contributions from $R^{\rm II}$, $R^{\rm III}$, and $R^{\rm III-CRD}$, respectively, while $\varepsilon_X^{\ell,\rm th}$ and $\varepsilon_X^{\rm c}$ are the contributions from the line thermal emissivity and continuum, respectively.*

independent, horizontal magnetic field with $B = 20\,$G. An overall good agreement between $R^{\rm III}$ and $R^{\rm III-CRD}$ settings is observed in all profiles. We note that the error is generally larger in the core and near wings of the Q/I, U/I, and V/I profiles. This conclusion remains valid for magnetic fields ranging from 10 G to 200 G. Indeed, these results show no significant differences and are therefore not reported.

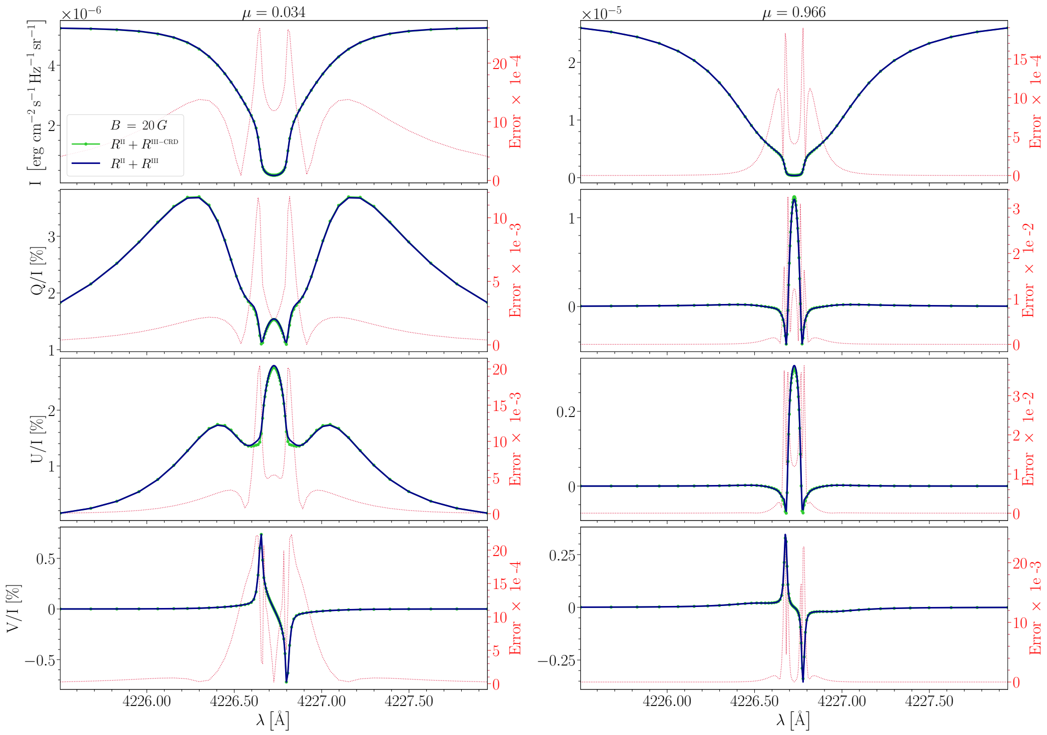

Figure 6.6. *Emergent Stokes I (first row), Q/I (second row), U/I (third row), and V/I (fourth row) profiles for the Ca* I *4227 Å line at $\mu = 0.034$ (left column) and $\mu = 0.966$ (right column) calculated in the FAL-C atmospheric model in the presence of a horizontal ($\theta_B = \pi/2$, $\chi_B = 0$) magnetic field with $B = 20$ G. Calculations take into account PRD effects considering the exact expression of $R^{\rm III}$ (blue lines) and the $R^{\rm III-CRD}$ approximation (green marked lines). The reference direction for positive Q is taken parallel to the limb. The red dotted lines report the error between $R^{\rm III}$ and $R^{\rm III-CRD}$ calculations, given by Eq. (6.8).*

6.4.2 Sr I 4607 Å line

We now consider the modeling of the Sr I 4607 Å line, which is discretized with $N_\nu = 130$ unevenly spaced nodes. Our non-LTE RT calculations adequately reproduce this *weak* line, showing a small absorption profile in the intensity spectrum without wings, and a prominent and sharp Q/I scattering polarization peak (see Fig. 6.7) .

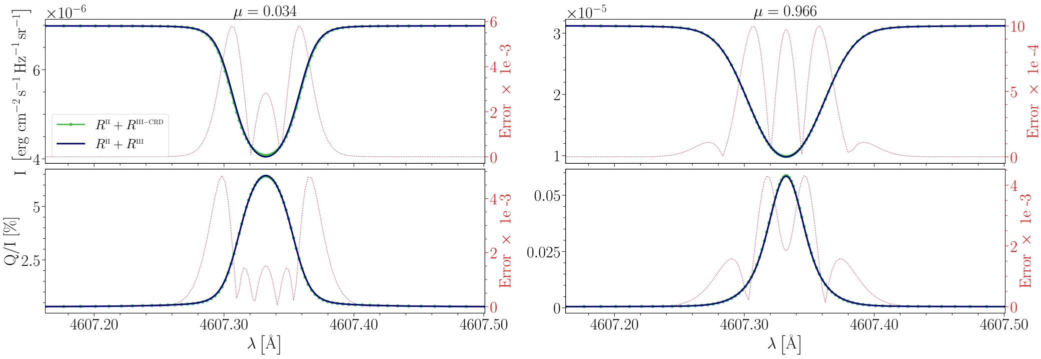

Figure 6.7. *Same as Fig. 6.4, but for the Sr* I *4607 Å line.*

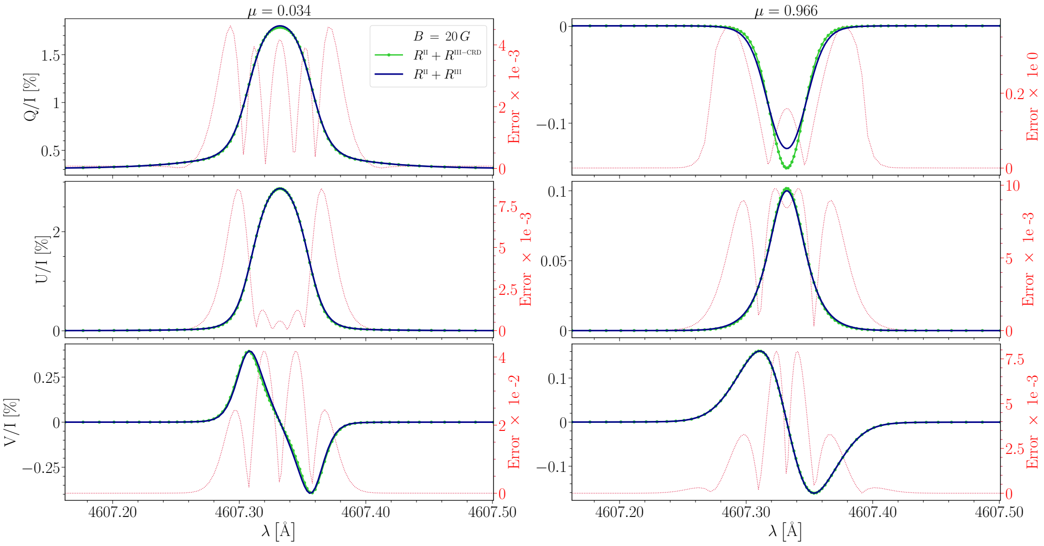

Figure 6.8. *Same as Fig. 6.6, but for the Sr* I *4607 Å line.*

In the I and Q/I profiles shown in Figure 6.7 for a non-magnetic scenario, we only note minor differences between $R^{\rm III}$ and $R^{\rm III-CRD}$ calculations. By contrast, we note some relevant discrepancies between $R^{\rm III}$ and $R^{\rm III-CRD}$ cases when a deterministic magnetic field is considered. Figures 6.8, 6.9, and 6.10 display the Q/I, U/I, and V/I profiles in the presence of a height-independent, horizontal magnetic field with $B = 20$ G, $B = 50$ G, and $B = 100$ G, respectively. We omit to report the intensity I profiles, because they are essentially identical to those exposed in Figure 6.7 for the non-magnetized case. We first note that, in all cases, there are no observable discrepancies between $R^{\rm III}$ and $R^{\rm III-CRD}$ calculations in the U/I and V/I profiles. By contrast, some relevant differences appear in the Q/I

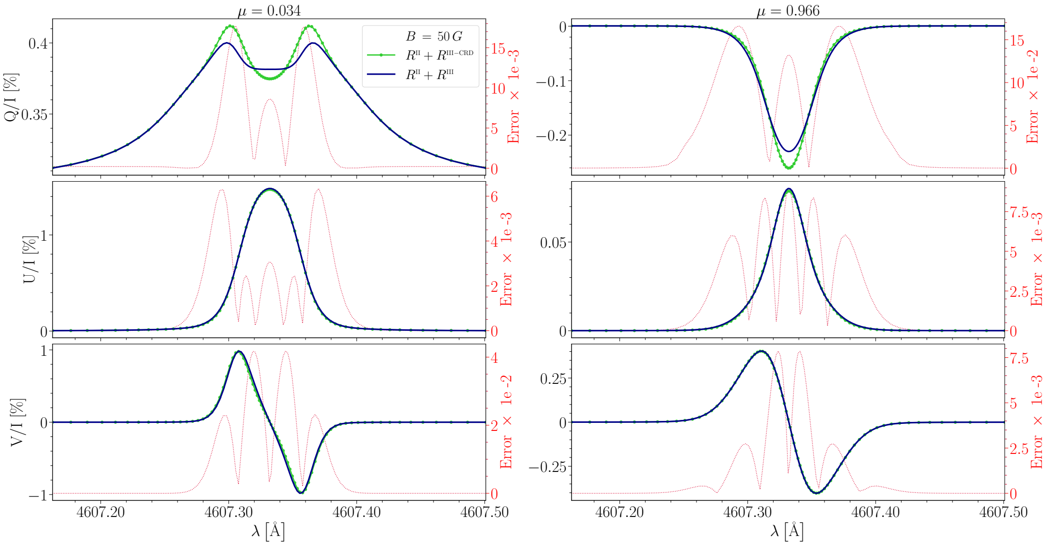

Figure 6.9. *Same as Fig. 6.8, but for $B = 50$ G.*

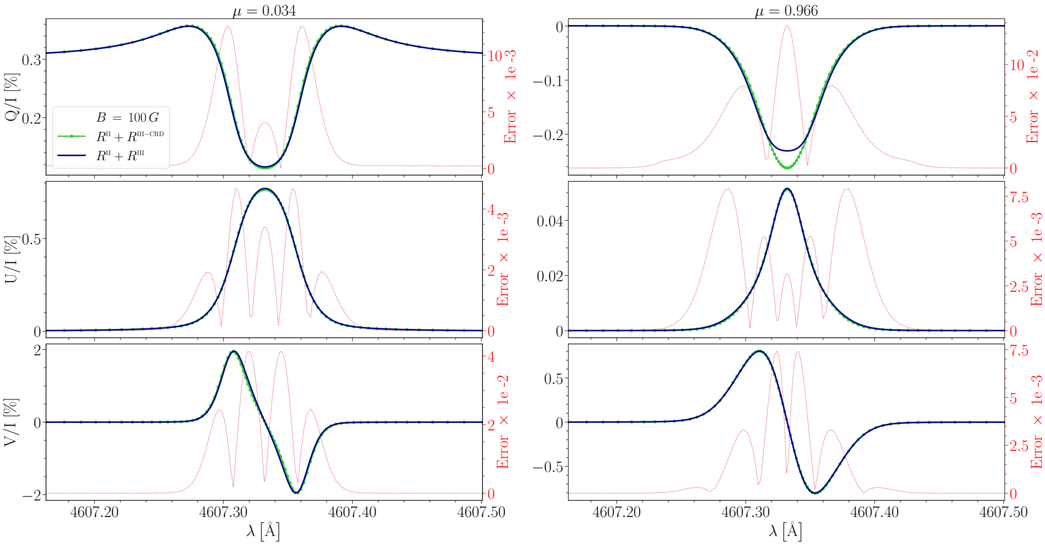

Figure 6.10. *Same as Fig. 6.8, but for $B = 100$ G.*

profiles, such as the one shown in the top-left panel of Figure 6.9. Moreover, $R^{\text{III}-\text{CRD}}$ calculations generally present larger Q/I line-core signals at $\mu = 0.966$ with respect to R^{III} ones. For completeness, Figure 6.11 displays the Q/I profile where we observed the maximal error, corresponding to the $B = 20$ G case at $\mu = 0.83$.

It is worth observing that for $\mu = 0.966$ and a magnetic field with $B = 20\,\mathrm{G}$ (i.e., a forward scattering Hanle effect geometry), the Ca I 4227 Å line shows a positive Q/I peak (see Fig.6.6), while the Sr I 4607 Å line a negative one (see Fig. 6.8). Bearing in mind that both spectral lines originate from a $J_\ell = 0 \rightarrow J_u = 1$ transition and have similar Hanle critical fields, we may suggest that this inversion is due to their different formation heights and properties. An in-depth analysis of this result goes beyond the scope of this paper and will be the object of a future investigation.

As anticipated, the Sr I 4607 Å line has been extensively exploited to investigate the small-scale unresolved magnetic fields that fill the quiet solar photosphere. For this reason, we have also analyzed the case of unimodal MSI magnetic fields. We recall that in the presence of such magnetic fields the signatures of Zeeman and magneto-optical effects vanish due to cancellation effects, and the only impact of the magnetic field is the depolarization of Q/I due to the Hanle effect, which depends on the field strength. Figure 6.12 shows the I and Q/I profiles for a height-independent MSI magnetic field with $B = 20$ G. No significant differences between R^{III} and $R^{\mathrm{III-CRD}}$ calculations are visible, and the error is always smaller than 5×10^{-3}. This result suggests that the $R^{\mathrm{III-CRD}}$ approximation provides accurate results when the problem is characterized by cylindrical symmetry, while it can introduce appreciable discrepancies when the direction of a non-vertical, deterministic magnetic field breaks this symmetry.

6.5 Numerical results: 1D atmospheric model from 3D MHD simulation

In Sect. 6.4, we limited our calculations to the semi-empirical FAL-C atmospheric model, possibly including spatially uniform magnetic fields, in the absence of bulk velocities. In this section, we compare the impact of R^{III} and $R^{\mathrm{III-CRD}}$ in PRD calculations in a 1D atmospheric model extracted from a 3D MHD simulation, which includes height-dependent magnetic and bulk velocity fields. As we have seen in previous sections, the magnetic field impacts the polarization profiles through the Hanle, magneto-optical, and Zeeman effects. On the other hand, bulk velocities generally introduce Doppler shifts as well as amplitude enhancements and asymmetries in the polarization profiles (e.g., Carlin et al., 2012, 2013; Jaume Bestard et al., 2021a). The joint impact of height-dependent magnetic fields and bulk velocities, together with the inherent thermodynamic structure of the atmospheric model, results in complex emergent Stokes profiles, in

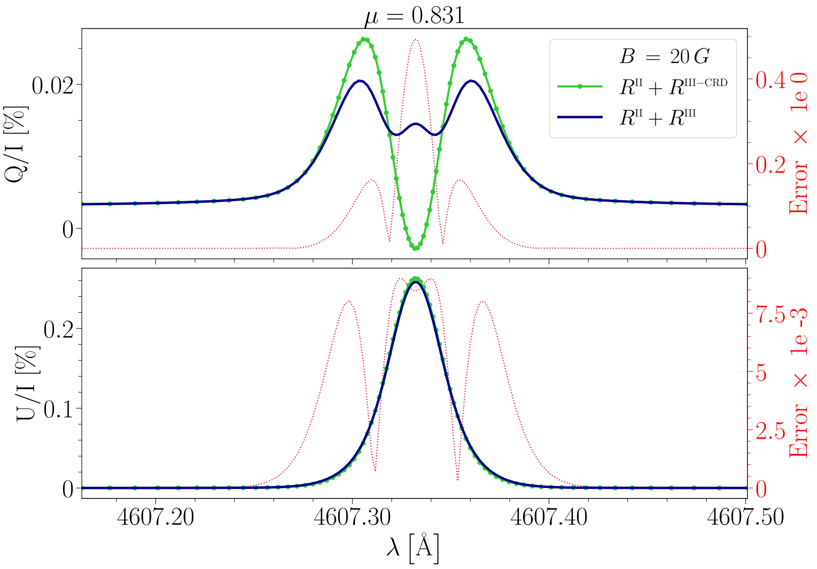

Figure 6.11. *Same as Fig. 6.8, but for $\mu = 0.83$. In this Q/I profile, we observed the maximal error between R^{III} and $R^{\mathrm{III-CRD}}$ calculations.*

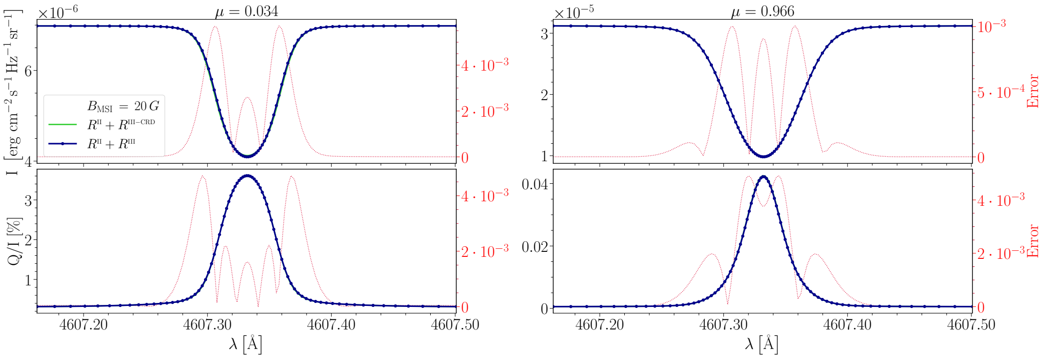

Figure 6.12. *Same as Fig. 6.7, but in the presence of a uniform micro-structured isotropic magnetic field with $B = 20\,G$*

which it is generally difficult to distinguish the contribution of each factor.

In this work, we adopted a 1D atmospheric model extracted from the 3D magnetohydrodynamic (MHD) simulation of Carlsson et al. (2016), performed with the Bifrost code (Gudiksen et al., 2011). In practice, we took a vertical column of such 3D simulation, clipped in the height interval between -100 and 1400 km, which includes the region where the considered spectral lines form, discretized with $N_z = 79$ nodes. The corresponding vertical resolution ranges between 19 km and 100 km. Figure 6.13 shows the variation of the magnetic field, temperature, and vertical component of the bulk velocity as a function of height in the considered model (hereafter, Bifrost model), which shows relatively quiet conditions. Since the 1D module of the RH code (used to calculate the lower-level population) can only handle vertical bulk velocities, in this study we only considered the vertical component of the model's velocity, although our code can in principle take into account velocities of arbitrary direction. As additional information, Fig. 6.14 shows the variation of the coherence fraction $\tilde{\alpha}$ with height in the Bifrost model for the two considered spectral lines, as well as the height at which the optical depth τ, in the frequency intervals of the two spectral lines, is unity (see Sect. 6.3).

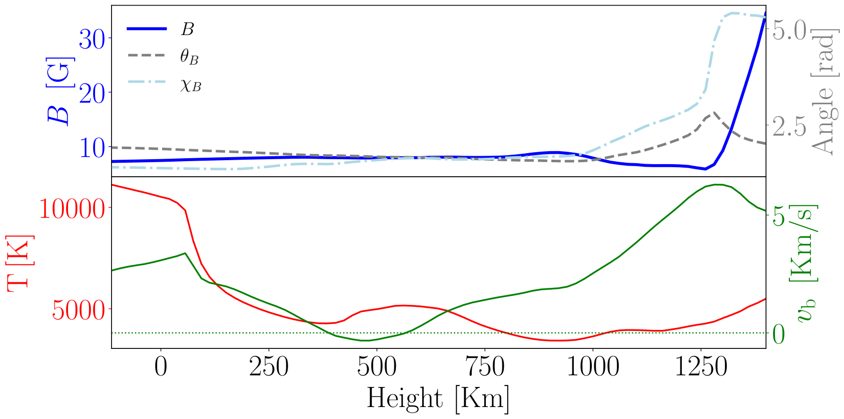

Figure 6.13. *Physical quantities of the considered 1D atmospheric model extracted from the 3D MHD Bifrost simulation en024048_hion (snapshot 385, column 120 × 120). Upper panel: strength (blue solid line), inclination (gray dashed line), and azimuth (light-blue dashed-dotted line) of the magnetic field as a function of height. Lower panel: vertical component of bulk velocity (green solid line) and temperature (red solid line) as a function of height. We adopt the convention that the velocity is positive if pointing outwards in the atmosphere and negative if pointing inwards. For clarity, the horizontal green dotted line indicates zero velocity.*

6.5.1 Ca I 4227 Å line

We first consider the modeling of the Ca I 4227 Å line, which is now discretized with $N_\nu = 141$ unevenly spaced nodes. Figure 6.15 displays all the Stokes profiles, showing an overall good agreement between $R^{\rm III}$ and $R^{\rm III-CRD}$ calculations

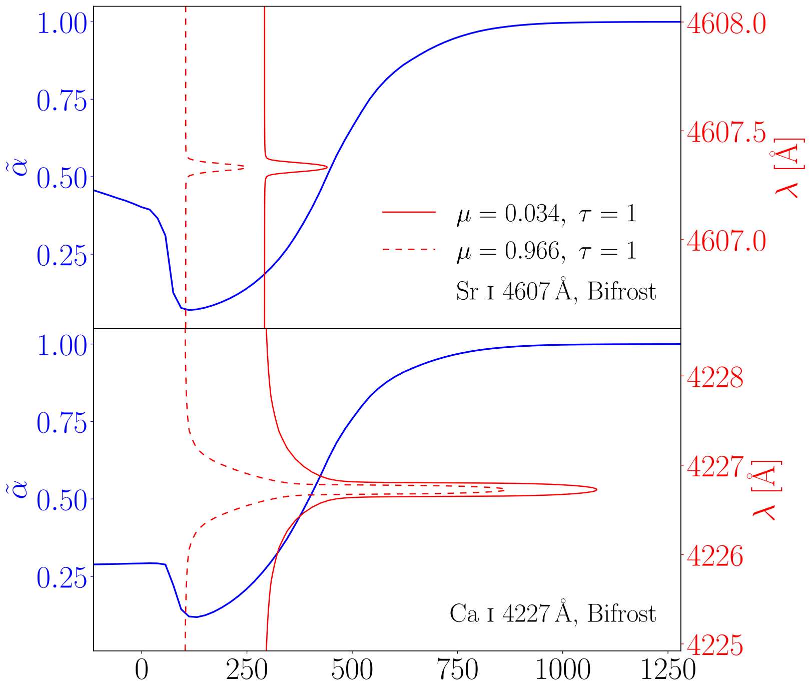

Figure 6.14. *Same as Fig. 6.3 but for the Bifrost atmospheric model.*

for a LOS with $\mu = 0.034$. By contrast, for a LOS with $\mu = 0.996$, we observe some appreciable discrepancies in the Q/I and U/I profiles, with an error up to 3×10^{-1}. We recall that these linear scattering polarization signals, obtained for a LOS close to the disk center, are due to the forward-scattering Hanle effect (e.g. Trujillo Bueno, 2001). Interestingly, such discrepancies disappear in the *absence* of bulk velocities (profiles not reported here). This suggests that the differences between $R^{\rm III}$ and $R^{\rm III-CRD}$ calculations are amplified in the presence of bulk velocities, and they are larger for a LOS close to the vertical, because in this case the Doppler shifts are more pronounced. We finally note that the error affects the amplitudes of the main peaks of the profiles but not their shape.

6.5.2 Sr I 4607 Å line

We now consider the modeling of the Sr I 4607 Å line, which is discretized with $N_\nu = 141$ unevenly spaced nodes. First, we note that in this Bifrost model, this line shows an emission profile in intensity for a LOS of $\mu = 0.034$. We verified that this is due to the thermodynamic structure of this atmospheric model, and it can be explained from the behavior of the source function at the formation height

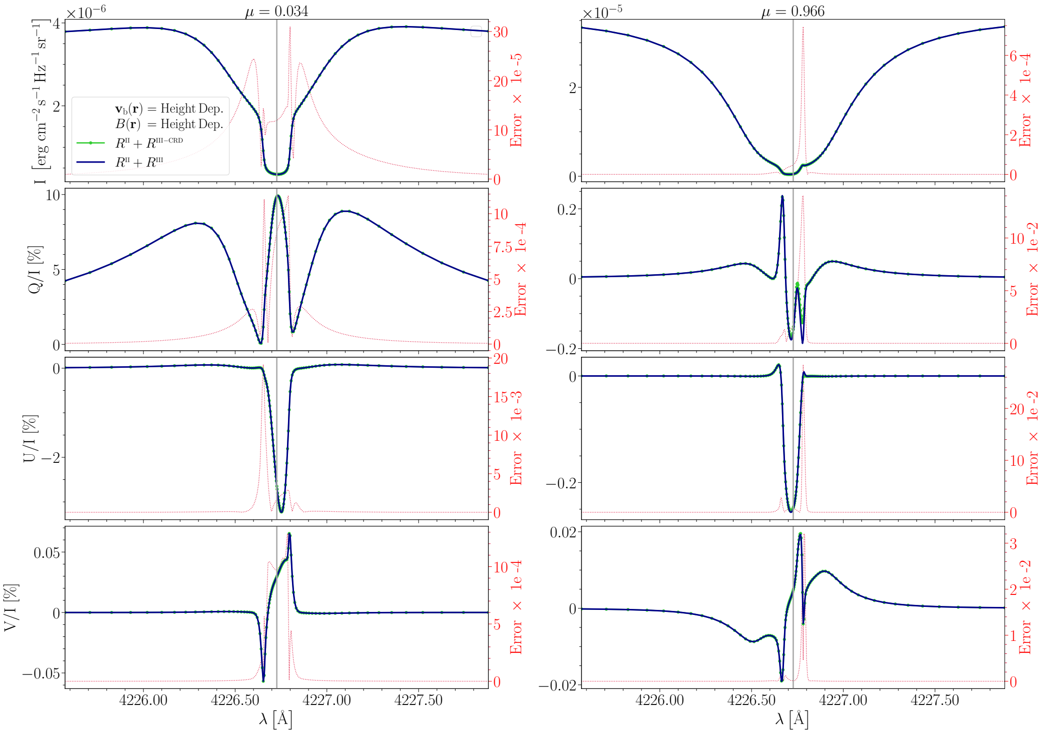

Figure 6.15. *Emergent Stokes profiles for the Ca* I *4227 Å line calculated in the Bifrost model (see Section 6.5) , which includes height-dependent magnetic and (vertical) bulk velocity fields. The vertical gray lines show the central line wavelength.*

of the line for this limb LOS (see Fig. 6.14). Figure 6.16 shows that appreciable discrepancies between $R^{\rm III}$ and $R^{\rm III-CRD}$ calculations are found in all Stokes profiles for $\mu = 0.034$, and in Q/I and U/I for $\mu = 0.966$. The maximal error is found in the U/I profile at $\mu = 0.966$, where however the polarization signal is very weak and thus of limited practical interest. The discrepancies between $R^{\rm III}$ and $R^{\rm III-CRD}$ computations become slightly larger in the Bifrost model, where also a bulk velocity field is included. On the other hand, observing that the most significant errors appear in very weak polarization signals, we can conclude that for practical applications the $R^{\rm III-CRD}$ approximation can be safely used to obtain reliable results.

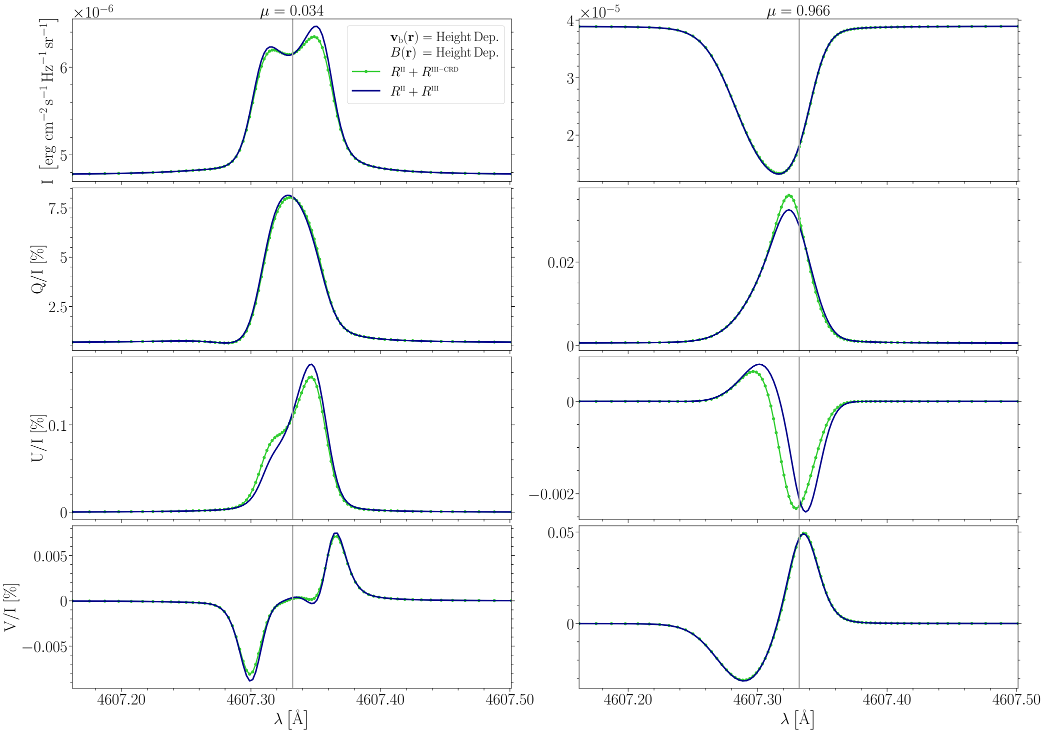

Figure 6.16. *Same as Fig. 6.15 but for the Sr* I *4607 Å line.*

6.6 Conclusions

In this work, we assessed the suitability and range of validity of a widely used approximation for R^{III}, that is its expression under the assumption of CRD in the observer's frame, labelled with $R^{\mathrm{III-CRD}}$. To this aim, we solved the full non-LTE RT problem for polarized radiation in 1D models of the solar atmosphere, considering both the exact expression of R^{III} and its $R^{\mathrm{III-CRD}}$ approximation. With respect to the previous work of Sampoorna et al. (2017), we considered semi-empirical models, as well as 1D models extracted from 3D MHD simulations, which provide a reliable approximation of the solar atmosphere. The analysis was focused on the chromospheric Ca I line at 4227 Å and the photospheric Sr I line at 4607 Å, accounting for the impact of magnetic fields (both deterministic and micro-structured) and bulk velocities.

We first compared the analytical forms of R^{III} and $R^{\mathrm{III-CRD}}$, showing that they correspond when the scattering angle Θ is equal to $\pi/2$, while their difference increases as Θ approaches 0 (forward scattering) or π (backward scattering). The numerical results for the Ca I 4227 Å line showed that the $R^{\mathrm{III-CRD}}$ approximation is suited to model the scattering polarization signals of strong chromospheric lines. This is not surprising, considering that in these lines the contribution of R^{II} dominates with respect to that of R^{III}, both in the core (which forms high in the atmosphere) and in the far wings (where scattering is basically coherent). On the other hand, we verified that the $R^{\mathrm{III-CRD}}$ approximation is also adequate in the near wings, where the contribution of R^{III} cannot be neglected a-priori. The numerical results for the Sr I 4607 Å line showed that the $R^{\mathrm{III-CRD}}$ approximation provides reliable results also in photospheric lines, for which the contribution of R^{III} is significant. Observing that the scattering polarization signal of the Sr I 4607 Å line is extensively used to investigate the small-scale magnetic fields that fill the quiet solar photosphere, we verified that the $R^{\mathrm{III-CRD}}$ approximation is also suitable in the presence of micro-turbulent magnetic fields. On the other hand, some appreciable discrepancies (i.e., relative errors larger than 0.01 according to the error definition (6.8)) are found when deterministic magnetic fields or bulk velocities are included, and the polarization signals are below 0.4 %. However, the qualitative agreement between the two settings remains in general satisfactory; differences in the overall shapes and signs are exceptions and appear when the signals are very weak. These discrepancies may suggest that the limit of CRD (i.e., to fully neglect PRD effects), which is generally adopted to model the intensity and polarization of these weak lines, can possibly introduce some appreciable inaccuracies with respect to a PRD treatment. A quantitative comparison between PRD and CRD calculations for this line is ongoing, and it will be presented in a separate publication.

In conclusion, we can state that in optically-thick media, the use of the lightweight $R^{\mathrm{III-CRD}}$ approximation guarantees reliable results in the modeling of scattering polarization in strong and weak resonance lines, thus confirming and generalizing the results of Sampoorna et al. (2017). This assesses the results of previous studies performed under this approximation and facilitates the development of 3D non-LTE RT simulations modeling the intensity and polarization of solar spectral lines, considering PRD effects (see Benedusi et al., 2023).

Chapter 7

Applications to different spectral lines

In this chapter, we present a series of non-LTE RT calculations carried out with the solution strategy and computational methods previously described. We model the intensity and polarization of three different spectral lines, namely Ca I 4227 Å, Sr II 4078 Å, and Sr I 4607 Å. All these lines show large scattering polarization signals, which can be suitably modeled considering a two-level atom. In the first sections, we show some spectropolarimetric measurements of these lines, performed in regions of the solar disk characterized by different levels of magnetic activity, both close to the edge of the solar disk (limb) and at the solar disk center. This will introduce the reader to the signals that we aim to model and to their magnetic sensitivity. All the observations are carried out at IRSOL with the ZIMPOL polarimeter. Subsequently, we present our theoretical profiles, and we compare the results obtained in the limit of complete frequency redistribution (CRD), and including PRD effects, both under the angle-averaged (AA) approximation and in the general angle-dependent (AD) case. Finally, we highlight the significant differences that are found between AA and AD calculations in the forward-scattering Hanle effect geometry. This is an example of the relevant physical results that the new computational methods developed in this thesis can provide. A paper collecting the results shown in this chapter is presently in preparation.

7.1 Considered spectral lines

We model the intensity and polarization of three spectral lines: the chromospheric Ca I 4227 Å and Sr II 4078 Å, and the photospheric Sr I 4607 Å. All these resonance lines show large scattering polarization signals, which can be suitably modeled considering a two-level atom with unpolarized and infinitely-sharp lower level. The Ca I 4227 Å and Sr I 4607 Å lines originate from a $J_\ell = 0 \longleftrightarrow J_u = 1$ transition, while the Sr II 4078 Å originates from a $J_\ell = 1/2 \longleftrightarrow J_u = 3/2$ transition (see Table A.1). In the intensity spectrum, the Ca I 4227 Å line shows a very broad and deep absorption profile with extended wings (in this sense it is considered a *strong* line; a useful quantity for characterizing the intensity profiles is presented in Sect. 7.4.2). On the contrary, the Sr I 4607 Å is a *weak* line showing a little absorption profile without wings. The Sr II 4078 Å is somehow intermediate, showing some wings, but not as extended as those of Ca I 4227 Å.

The Hanle critical field B_c (i.e., the magnetic field at which the sensitivity of a spectral line to the Hanle effect is highest) is the field strength at which the splitting of the magnetic sublevels of the upper level is of the same order of magnitude as its natural width. Form Eq. (10.29) of Landi Degl'Innocenti and Landolfi (2004), we have

$$B_c = \frac{A_{u\ell}}{8.79 \cdot 10^6 \, g_u}, \tag{7.1}$$

where B_c is expressed in gauss, $A_{u\ell}$ is the Einstein coefficient for spontaneous emission (in s^{-1}), and g_u is the Landé factor of the upper level. The Hanle critical field and other physical parameters of the considered spectral lines are provided in Table 7.1.

Line	Transition		Formation region	$A_{u\ell}$ [s^{-1}]	g_u	B_c [G]
	J_ℓ	J_u				
Sr I 4607 Å	0	1	photosphere	$2.01 \cdot 10^8$	1	22.9
Sr II 4078 Å	$\frac{1}{2}$	$\frac{3}{2}$	chromosphere	$1.41 \cdot 10^8$	4/3	12
Ca I 4227 Å	0	1	chromosphere	$2.18 \cdot 10^8$	1	24.8

Table 7.1. *Physical parameters of the considered lines.*

7.2 Geometrical parameters of observations

Scattering polarization signals are typically observed near the edge of the solar disk (limb), where their amplitude is maximum. A fundamental parameter to interpret these observations is the distance between the observed region and the limb. This can be specified either through the *heliocentric angle* θ (i.e., the angle between the solar radius through the observed point and the direction towards the Earth) or through the angle α (generally expressed in arcsec) that subtends the projected distance h between the observed point and the limb (see Fig. 7.1). The angles α and θ are related by

$$\tan\alpha = \frac{R(1-\sin\theta)}{D} \tag{7.2}$$

where $R \approx 696340\,\mathrm{km}$ is the solar radius and $D \approx 150\times10^6$ km is the Sun-Earth distance. It can be noticed that the heliocentric angle θ coincides with the angle between the local vertical through the observed point and the line of sight (LOS) towards the observer. The latter is generally specified by $\mu = \cos\theta$. Radiation coming from the solar disk center has $\mu = 1$, while radiation coming from regions close to the limb have $\mu \approx 0$. In this study, we report numerical results for three LOS (corresponding to three inclinations from the angular grid used to discretize the problem, see Sect. 1.8.2), namely $\mu = 0.169$ (corresponding to a limb observation), $\mu = 0.619$, and $\mu = 0.966$ (observation close to the disk center).

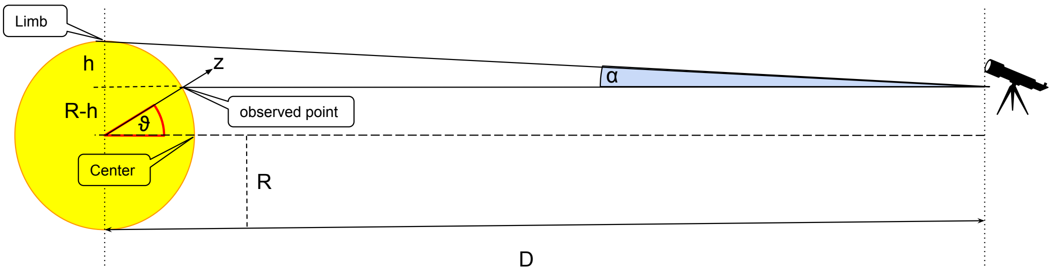

Figure 7.1. *Schematic illustration of the heliocentric angle θ and of the limb distance α used to specify the position of the observed point on the solar disk. D is the Sun-Earth distance, R is the solar radius, and h is the distance between the observed point and the limb, projected on the plane of the sky.*

7.3 Observational results

In this section, we report a series of observations carried out at the Istituto Ricerche Solari Aldo e Cele Daccò (IRSOL) in Locarno (Switzerland), using the Zurich Imaging Polarimeter[1] (ZIMPOL; Ramelli et al., 2010). The main purpose is to present some real spectropolarimetric measurements of the considered spectral lines and to highlight the magnetic sensitivity of the polarization signals through the physical mechanisms discussed in the previous chapters.

[1] IRSOL Instrumentation

7.3.1 Observational targets

We show measurements carried out in solar regions with very low magnetic activity (*quiet Sun*), and with weak or moderate activity. We do not report observations of regions with strong magnetic fields (e.g., sunspots) because these are out of the scope of this work.

Quiet Sun regions close to the limb are generally found near the north or south poles. The polarization measurements performed in these regions are dominated by scattering polarization, while no linear or circular polarization signals produced by the Zeeman effect are appreciable. It must be stressed that small-scale unresolved magnetic fields (a.k.a. micro-turbulent fields), to which the Zeeman effect is insensitive, can be present in these regions and induce a decrease of the scattering polarization signals via the Hanle effect. Taking the reference direction for positive Stokes Q parallel to the limb, the polarization measurements performed in these regions only show Q/I signals, while U/I and V/I are generally zero (i.e., at the noise level)[2].

By weakly magnetized regions, we mean here locations on the solar disk where large-scale deterministic fields, with strengths in the Hanle effect regime (i.e., up to approximately $5 \times B_c$), are present. When these regions are observed at the limb, the linear polarization is still dominated by scattering polarization signals, modified by the joint action of the Hanle and magneto-optical (MO) effects. We recall that the Hanle effect operates in the line core and generally causes a depolarization of Q/I and (in the case of deterministic magnetic fields) the appearance of a U/I signal. MO effects, due to deterministic magnetic fields with a non-zero longitudinal component, operate in the line wings and, similar to the Hanle effect, usually cause a decrease of Q/I and the appearance of U/I signals. The V/I signals produced by the Zeeman effect can be observed in these regions.

Scattering polarization signals are generally quite weak (below 1%). To measure these signals, it is first necessary to apply suitable instruments and observational techniques that allow for reducing systematic errors, e.g., the noise introduced by variations of the Earth's atmosphere (seeing) during the measurements, or by the telescope optics and electronics. When such systematic errors are sufficiently reduced and the measurement error is dominated by the photon

[2] Theoretical calculations (e.g., Jaume Bestard et al. (2021a)) point out that non-zero U/I scattering polarization signals, due to the horizontal inhomogeneities of the solar atmosphere and horizontal gradients of the plasma bulk velocity, could in principle be present in these quiet regions, but very high spatial and temporal resolutions (much higher than those achievable at IRSOL) would be needed to detect them.

(shot) noise, the signal-to-noise ratio is proportional to $\sqrt{N}$, with N the number of collected photons. Under such optimal conditions, also the most sensitive polarimeters like ZIMPOL generally need exposure times of the order of minutes to achieve sufficient signal-to-noise ratios of 10 or better, as in the observations reported below.

7.3.2 Observed Stokes profiles

Figure 7.2 shows two spectropolarimetric measurements of the Ca I 4227 Å line performed close to the limb, one in the quiet Sun (blue line) and one in a weakly magnetized region (green line). The Q/I profile measured in the quiet Sun shows the triplet-peak structure characteristic of strong resonance lines, with a sharp peak in the core and very broad lobes in the wings. We recall that the wing lobes are produced by coherent scattering processes with PRD effects (they cannot be reproduced in a CRD modeling). We note that several other spectral lines fall in the wings of Ca I 4227 Å. These blended lines are responsible for the depolarization features observed in the wings of the Q/I profile. The U/I and V/I profiles of this observation are not reported (practically set to zero) as they do not show any appreciable signals above the noise level. The Q/I profile measured in the weakly magnetized region clearly shows the impact of the Hanle effect, which almost completely depolarizes the line-center peak. The lower Q/I signals observed in the wing lobes are mainly due to the larger limb distance (i.e. larger value of μ) of this observation, although some depolarization produced by MO effects may also be present. The U/I signal shows a peak in the core due to the Hanle effect and some signals in the wings due to MO effects. Finally, some weak V/I signals are also present.

Very similar considerations can be done concerning the spectropolarimetric observations of the Sr II line at 4078 Å carried out at the limb (see Fig. 7.3). In quiet regions (blue line), the Q/I profile shows a triplet-peak structure similar to that of the Ca I 4227 Å line, but with less extended wing lobes. In the weakly magnetized region (green line), the impact of the Hanle effect can be clearly appreciated, both in Q/I and U/I. The Q/I wing peaks of the two observations have very similar amplitudes. This can indicate that very similar limb distances were considered in the two observations and that MO effects do not have an appreciable impact in this case.

Figure 7.4 shows two limb observations for the photospheric Sr I 4607 Å line. This is a weaker line in the intensity spectrum, and its Q/I profile shows a single sharp peak in the line-core. As for the previous lines, the blue and green profiles are obtained in a quiet region, and in a weakly magnetized one, respectively. The different value of Q/I in the continuum is a clear indication of the different limb distances of the two observations. The lower amplitude of the Q/I peak observed in the more magnetized region is partly due to the different limb distance and partly to the Hanle effect, whose operation is testified by the clear U/I peak. Remarkably, this was the first observation ever performed showing a clear U/I Hanle signal in this line (Zeuner et al., 2022).

Forward Scattering Hanle Effect

The line scattering polarization signals are generally stronger at the limb and decrease as the LOS approaches the disk center. For radiation fields having cylindrical symmetry, it can be shown that the amplitude scales by a factor of $(1-\mu^2)$ (e.g. Landi Degl'Innocenti and Landolfi, 2004). However, a deterministic, non-vertical magnetic field breaks this symmetry and can induce significant polarization peaks at the line core frequencies through the Hanle effect. This phenomenon, commonly referred to as the *Forward Scattering Hanle Effect* (hereafter FSHE), was predicted theoretically by Trujillo Bueno (2001). Figures 7.5 and 7.6 report observations of the Ca I 4227 Å and Sr II 4078 Å lines, respectively, carried out in a magnetized region close to the solar disk center (the two measurements were performed one after the other pointing the telescope to the same region). The strong Q/I and U/I signals detected in the Ca I 4227 line can be attributed to the FSHE. Signals of similar amplitudes, also ascribable to the FSHE, are appreciable in the Sr II 4078 Å as well, although they are much noisier due to the reduced transparency of the optics at the lower wavelength of this line. Both observations show clear V/I signals, indicating moderate magnetic activity in the observed region.

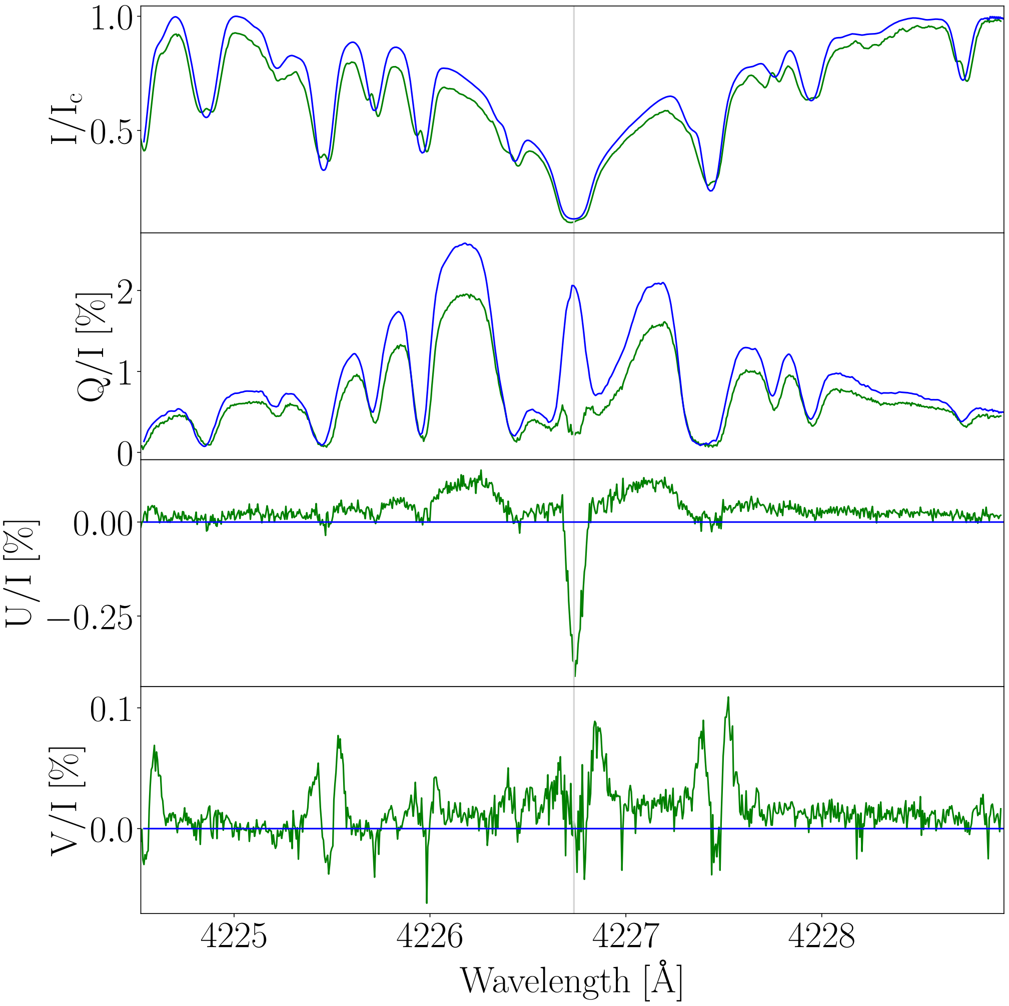

Figure 7.2. *Stokes I, Q/I, U/I, and V/I (top to bottom) profiles measured close to the limb across a wavelength interval containing the Ca* I *4227 Å line. The reference direction for positive Stokes Q is parallel to the closest limb. The blue profiles were measured in a quiet region at* $\mu = 0.1$ *(data from Gandorfer (2002)). The green profiles were measured in 2012 in a weakly magnetized region at* $0.1 \leq \mu \leq 0.2$ *(courtesy of Dr. Michele Bianda).*

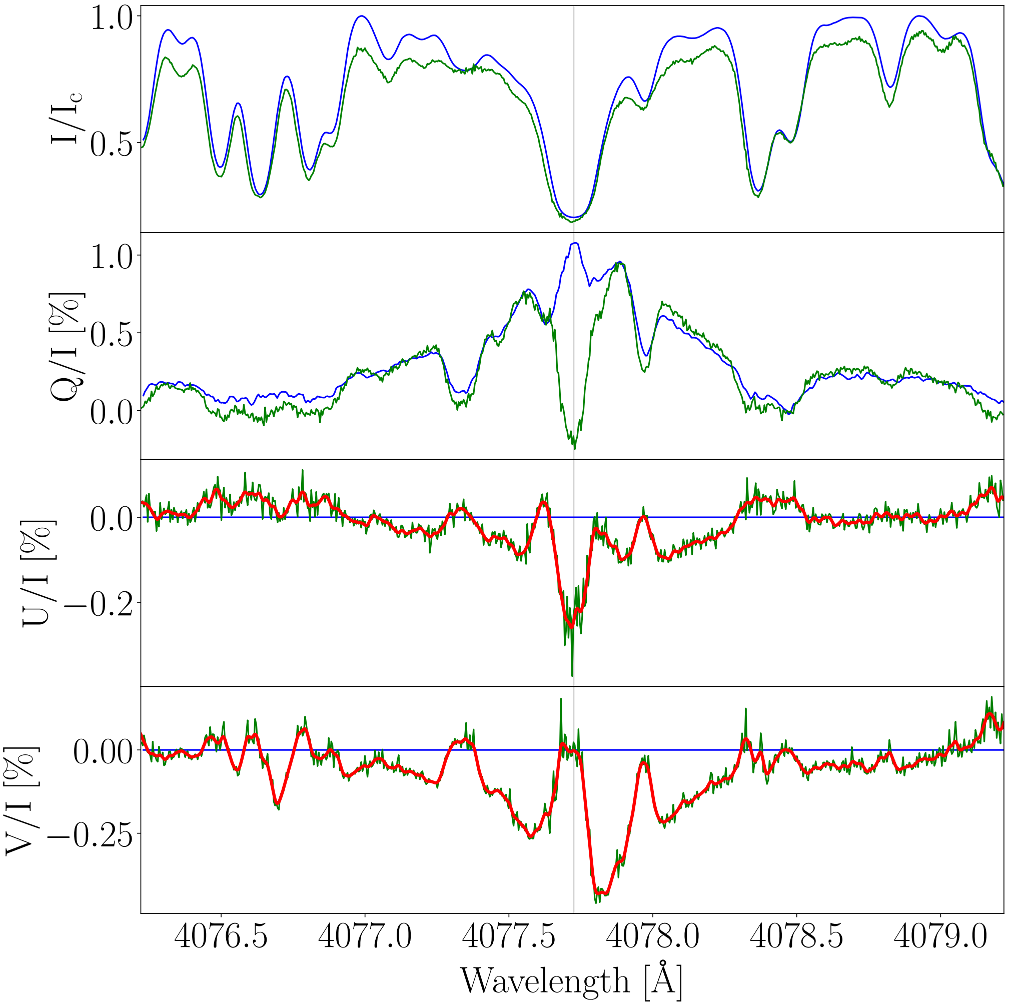

Figure 7.3. *Stokes I, Q/I, U/I, and V/I (top to bottom) profiles measured close to the limb across a wavelength interval containing the Sr* II *4078 Å line. The reference direction for positive Stokes Q is parallel to the closest limb. The blue profiles were measured in a quiet region at $\mu = 0.1$ (data from Gandorfer (2002)). The green profiles were measured in 2012 in a weakly magnetized region at $0.1 \leq \mu \leq 0.2$ (courtesy of Dr. Michele Bianda). The red lines in U/I and V/I are the signals obtained after a noise reduction, performed by applying a Gaussian low-pass filter in the Fourier space.*

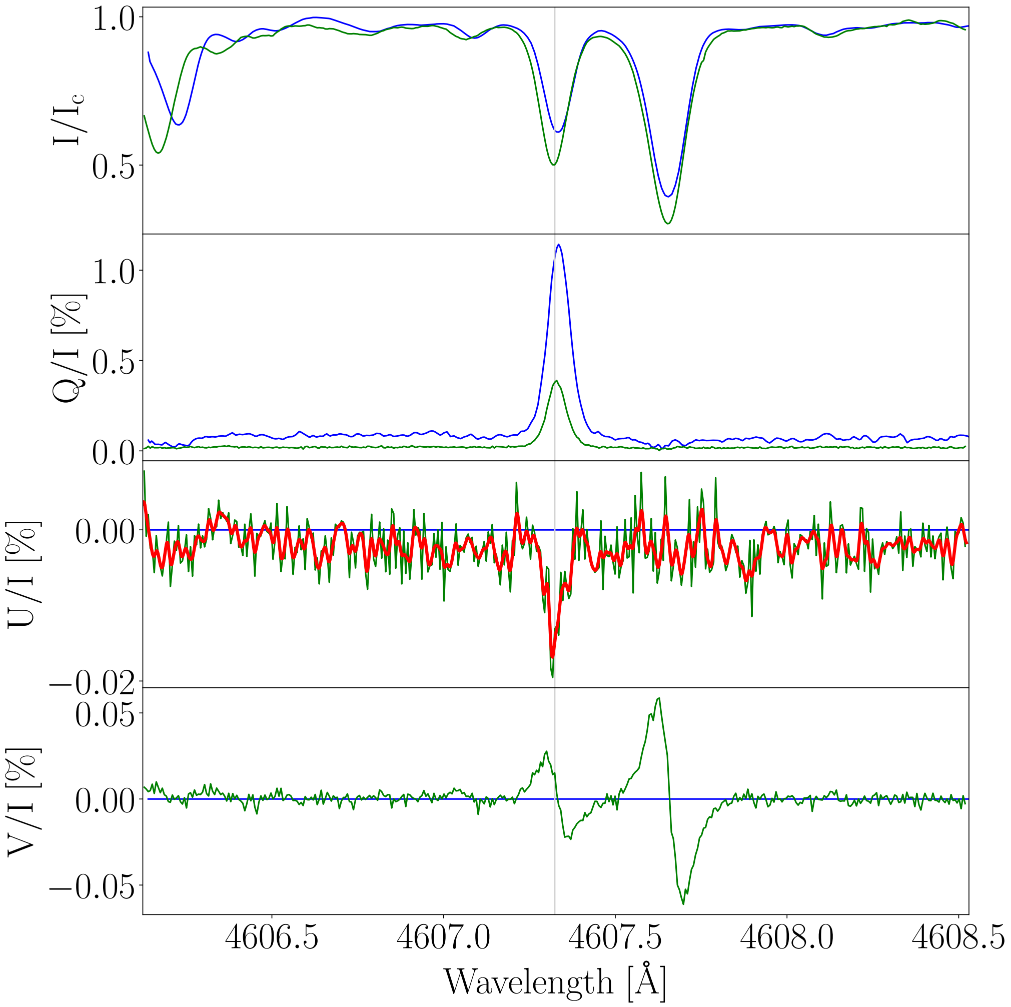

Figure 7.4. *Stokes I, Q/I, U/I, and V/I (top to bottom) profiles measured close to the limb across a wavelength interval containing the Sr* I *4607 Å line. The reference direction for positive Stokes Q is parallel to the closest limb. The blue profiles were measured in a quiet region at $\mu = 0.1$ (data from Gandorfer (2002)). The green profiles were measured on August 11, 2021, in a weakly magnetized region at $0.3 \leq \mu \leq 0.4$ (data from Zeuner et al. (2022)). The red line in Q/I is the signal obtained after a noise reduction, performed by applying a Gaussian low-pass filter in the Fourier space.*

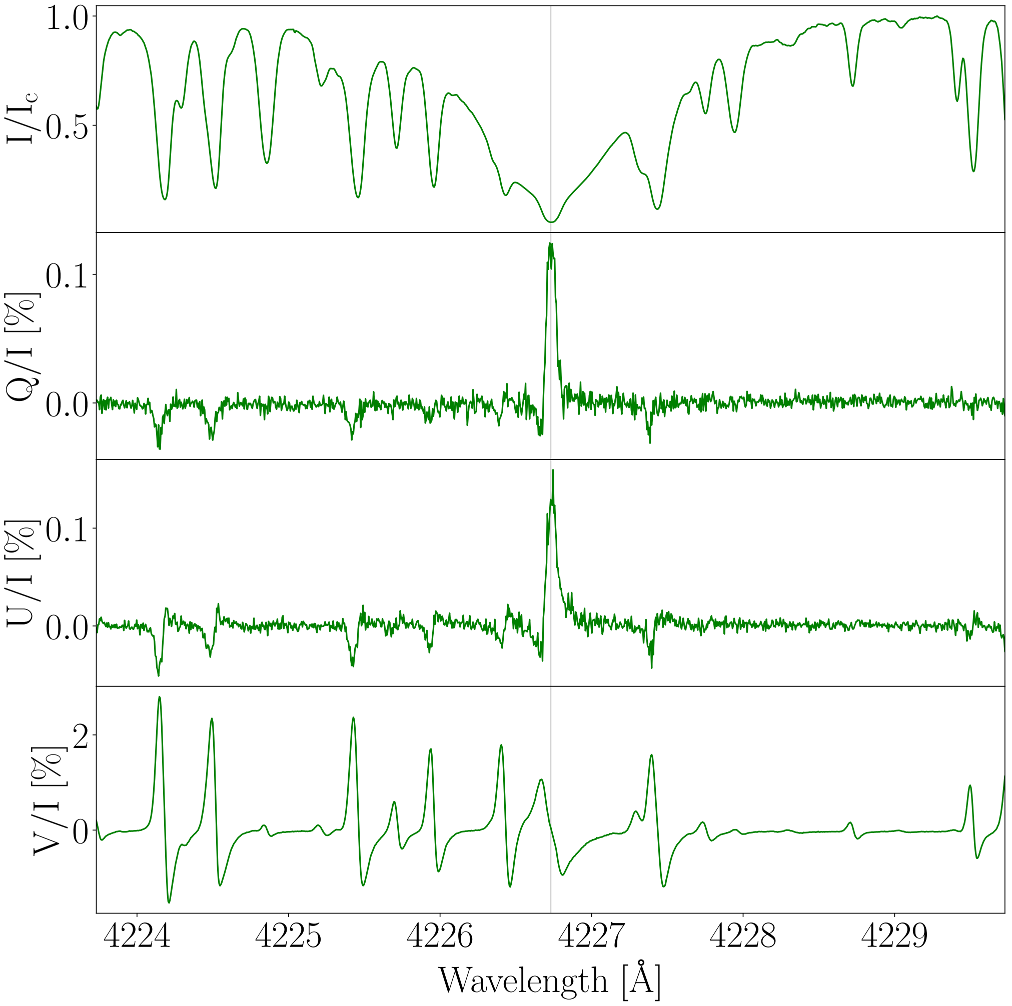

Figure 7.5. *Stokes I, Q/I, U/I, and V/I (top to bottom) profiles across a wavelength interval containing the Ca* I *4227 Å line, measured on Februray 14, 2023, in a moderately magnetized region close to the solar disk center (courtesy of Dr. Franziska Zeuner). The reference direction for positive Stokes Q is parallel to the closest limb.*

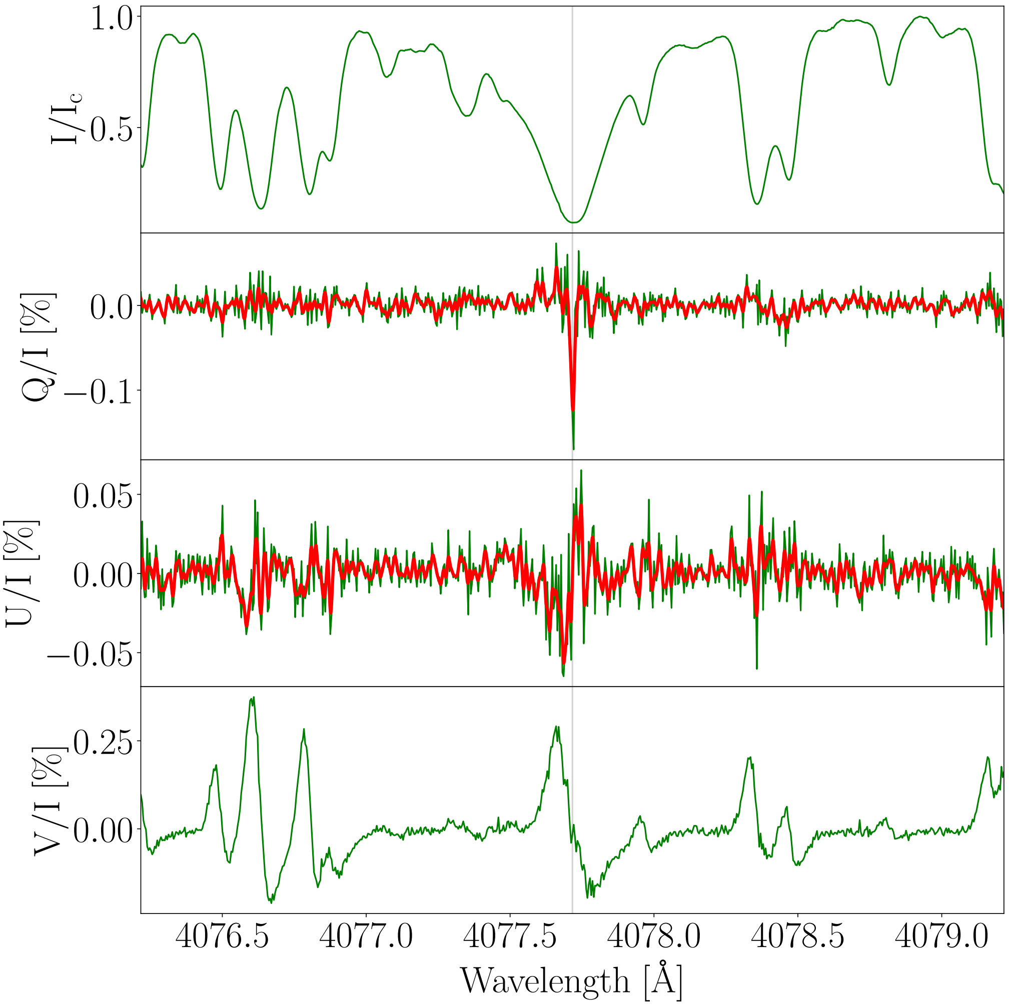

Figure 7.6. *Stokes I, Q/I, U/I, and V/I (top to bottom) profiles across a wavelength interval containing the Sr* II *4078 Å line, measured on Februray 14, 2023, in a moderately magnetized region close to the solar disk center (courtesy of Dr. Franziska Zeuner). The reference direction for positive Stokes Q is parallel to the closest limb. The red lines in Q/I and U/I are the signals obtained after a noise reduction, performed applying a Gaussian low-pass filter in the Fourier space.*

7.4 Theoretical results

In this section, we present the results of non-LTE RT calculations of the intensity and polarization profiles of the aforementioned spectral lines. In particular, we make a quantitative comparison between PRD results obtained considering the exact angle-dependent (AD) expression of R^{II} and its angle-average (AA) approximation (see Eq. A.40), and CRD calculations. In the PRD case, the approximation of CRD in the observer's frame for the R^{III} redistribution matrix ($R^{III-CRD}$) is always considered. The suitability of this approximation, especially for chromospheric lines, was analyzed in the previous chapter 6. The CRD calculations are performed applying the theory of Landi Degl'Innocenti and Landolfi (2004), introducing the corresponding redistribution matrix (R^{CRD}). Hereafter, these cases will be denoted with the following compact notation: AD ($R^{II} + R^{III-CRD}$), AA ($R^{II-AA} + R^{III-CRD}$, and CRD (R^{CRD}). The RT calculations are performed in 1D atmospheric models, both semi-empirical and extracted from a 3D MHD simulation. The problem is solved as in Benedusi et al. (2022) through preconditioned Krylov subspaces iterations (see also Sect. 1.8.2).

7.4.1 Atmospheric models

We consider the following 1D atmospheric models:
a) The semi-empirical model C of Fontenla et al. (1993) (hereafter FAL-C), in the presence of a spatially-constant deterministic magnetic field with direction ($\theta_B = \pi/4$, $\chi_B = \pi/4$) and with a strength corresponding to the *Hanle critical field* of the considered spectral line (see Table 7.1). In this work we also consider the presence of unimodal microstructured isotropic magnetic fields (hereafter MSI) (e.g. Alsina Ballester et al., 2017), that is, a magnetic field that has a specific strength while its orientation varies on scales below the mean free path of photons and is assumed to be uniformly distributed in all directions. The use of MSI magnetic fields is of diagnostic interest for photospheric lines (e.g. the Sr I 4607 Å line), since there is evidence that the solar photosphere is permeated by such small-scale non-deterministic fields (e.g. Trujillo Bueno et al., 2004; Manso Sainz et al., 2004). In these tests, we do not include a bulk velocity field.
b) a 1D atmospheric model extracted from the 3D magnetohydrodynamic (MHD) simulation of Carlsson et al. (2016), performed with the Bifrost code of Gudiksen et al. (2011) (hereafter, Bifrost model). Specifically, we chose a vertical column of such a 3D simulation, clipped in the height interval where the considered spectral lines form, i.e. in the range between -100 and 2200 km, discretized with $N_r = 118$ spatial nodes. The corresponding vertical resolution varies between

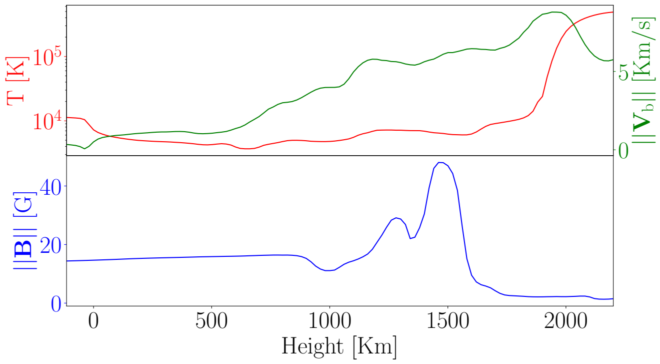

Figure 7.7. *Temperature (upper panel, red line), the vertical component of the bulk velocity (upper panel, green line), and strength of the magnetic field (lower panel, blue line) as a function of height in the Bifrost atmospheric model.*

19 km and 100 km. This atmospheric model includes height-dependent (non-uniform) magnetic and bulk velocity fields, whose values are given in Fig. 7.7.

7.4.2 Optical depth and line formation

Depending on the frequency (e.g., in the continuum or in a spectral line) and LOS, the radiation emergent from the solar atmosphere encodes information on different atmospheric layers. A fairly good approximation of this so-called *formation region* of the radiation is the height where the optical depth τ (see Section 1.7 on page 28), calculated inwards from the top of the solar atmosphere, at the considered frequency and propagation direction is unity. This can be qualitatively interpreted by observing that the height where $\tau \approx 1$ is where the considered beam has its "last" interactions with the atmospheric plasma. Above this height, where $\tau \ll 1$, the atmosphere is essentially transparent to the considered radiation, while below, where $\tau \gg 1$, the radiation still strongly interacts with the plasma and has a high probability of being modified before emerging. Recalling the definition of the optical depth (see Section 1.7 on page 28), it is clear that the formation height is higher in the core of the spectral lines (where the absorption coefficient is larger) and decreases moving to the wings and continuum. Moreover, observing that the distance traveled by a radiation beam between two height points is larger for inclined LOS, it is clear that, for a given frequency, the formation height is higher for a LOS near the limb than at the solar disk center. For instance, in the FAL-C model, the continuum at around 5000 Å at the solar disk center originates at around 0 km, while at the limb it forms between 150 and 200 km.

In Figure 7.9 we compare the formation heights, as a function of wavelength, across the three resonant lines considered in this work for the two models, i.e.

	Ca I 4227 Å		Sr II 4078 Å		Sr I 4607 Å	
μ	FAL-C	Bifrost	FAL-C	Bifrost	FAL-C	Bifrost
0.169	0.952	1.409	0.411	0.574	0.059	0.077
0.699	1.090	1.639	0.390	0.534	0.063	0.081
0.966	1.046	1.580	0.346	0.419	0.060	0.079

Table 7.2. *Equivalent width W_λ (in Å) of the synthetic intensity profiles of the considered spectral lines, calculated for different LOS in the FAL-C and Bifrost models. For I_c, we used the intensity value at the first wavelength point of the grid, which falls in the continuum. The integral was estimated using the trapezoidal rule on the nodes of the frequency grid of the problem U (see Sect. 1.8.2 for more details on the discretization of the problem).*

FAL-C (left column) and Bifrost (right column). It can be noticed that the formation heights of all the considered spectral lines (from the core till the continuum) are slightly higher in the Bifrost model than in FAL-C. This seems to be due to the larger values of the lower-level population in the Bifrost model (see lower panel of Fig. 7.9). Moreover, the curves of the heights where $\tau = 1$ as a function of wavelength are slightly wider in the Bifrost model than in FAL-C, especially for the Ca I 4227 Å and Sr II 4078 Å lines. Interestingly, both the intensity and polarization profiles obtained from the Bifrost model are generally broader than those obtained from FAL-C (see also sections 7.4.4, 7.4.5, and 7.4.6). This can be better quantified by means of the *equivalent width*:

$$W_\lambda = \int_{\mathrm{line}} \frac{I_c - I(\lambda)}{I_c}\,\mathrm{d}\lambda. \tag{7.3}$$

As it can be seen from Table 7.2, the value of W_λ is sensibly larger for the theoretical profiles calculated in the Bifrost model.

These differences in the equivalent width do not seem to be due to the different temperature structures (and thus the Doppler width) of the two models. Indeed, the temperature of FAL-C and Bifrost is very similar between approximately 100 and 400 km, while between 400 and 1100 km (where the cores of the Ca I 4227 Å and Sr II 4078 Å lines form), the Bifrost model is appreciably cooler than FAL-C (see the lower panel of Fig. 7.8). Such differences might instead be due, at least in part, to the larger values of the damping parameter at the formation height of the wings in the Bifrost model (see the lower panel of Fig. 7.8).

In Table 7.2 we report the estimated equivalent width (Eq. (7.3)) derived from the results of the calculations, where we observe that W_λ is usually larger

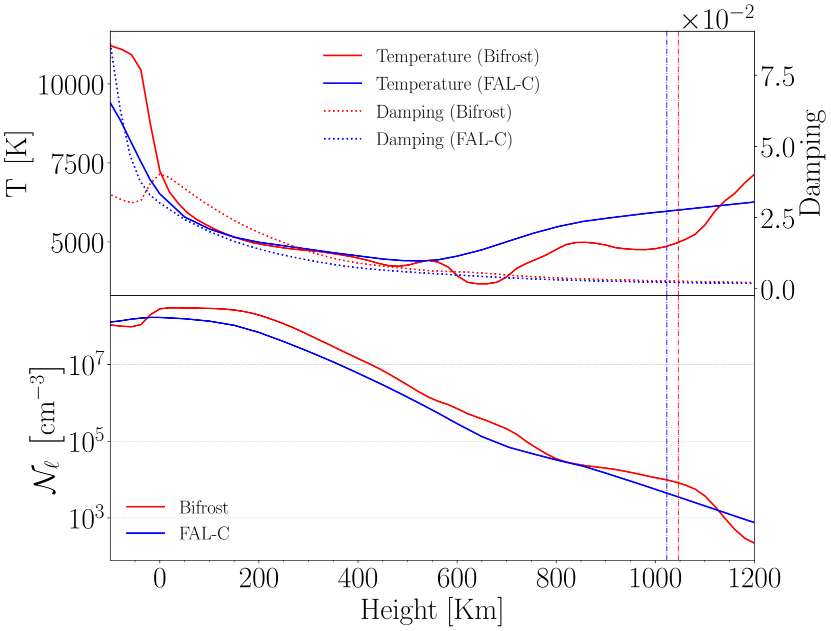

Figure 7.8. *Temperature and Damping parameter (upper panel) and the population of the lower level $\mathcal{N}_\ell$ (lower panel) as a function of the height in the two atmospheric models under consideration: Bifrost (red lines), and FAL-C (blue lines). The vertical lines are where $\tau = 1$ at central line frequency in the two atmospheric models.*

in the Bifrost model. These differences can also be appreciated by carefully observing the plots, where the profiles (both in intensity and polarization) obtained with the Bifrost model are broader when compared with the results obtained with the FAL-C model (see also sections 7.4.4, 7.4.6, and 7.4.5).

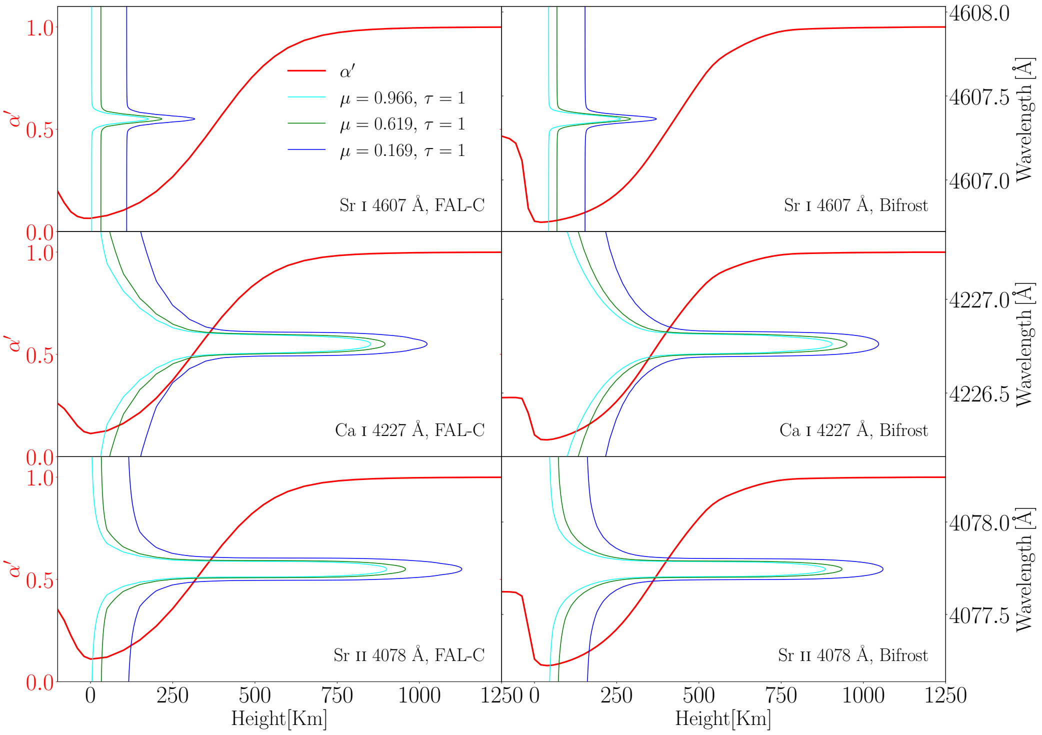

Figure 7.9. *Height variation of $\alpha'(\mathbf{r})$ (red line) for Sr* I *4607 Å (top panels), Ca* I *4227 Å (middle panels), and the Sr* II *4078 Å (bottom panels) in the FAL-C model (left column) and the Bifrost model (right column). The cyan, green, and blue isolines show the wavelength and height below which the optical depth becomes greater than* 1.0 *at directions $\mu = 0.966$, $\mu = 0.619$, and $\mu = 0.199$, respectively (in the absence of a bulk velocity field).*

7.4.3 Objectives and method

This study aims at assessing the reliability of the polarization profiles obtained under the AA and CRD approximations, using the results of AD calculations as reference solutions.

The first aspect that we want to investigate is the suitability of the AA approximation for modeling the scattering polarization wing lobes observed in strong chromospheric lines. These wing signals are receiving increasing attention because it was recently discovered that they are sensitive to the magnetic field via MO effects (Alsina Ballester et al., 2016; Capozzi et al., 2020). We recall that these polarization signals are produced by scattering processes that are coherent in the atomic frame (i.e., scattering processes described by R^{II}), and cannot be reproduced considering the limit of CRD. The analysis will be carried out focusing on the Ca I 4227 Å and Sr II 4078 Å lines.

The second aspect that we want to analyze is the suitability of the CRD approximation for modeling the line-core scattering polarization signals of both strong chromospheric lines and weak photospheric ones. The core of strong chromospheric lines generally forms quite high in the atmosphere, where R^{II} dominates. However, since Doppler redistribution is very efficient in this spectral region, it has been often argued that the CRD approximation may represent a fairly good approximation for modeling the line-core signals of these lines, both at the limb and near the disk center, for an FSHE geometry. On the other hand, the suitability of this approximation has not been supported so far by detailed comparisons between CRD and AD calculations in realistic atmospheric models. This is one of the goals of this work, using the Ca I 4227 Å and Sr II 4078 Å lines as test benches. A recent work (Janett et al., 2019) has already shown that the AA approximation can introduce significant inaccuracies in the modeling of the line-core scattering polarization signals of chromospheric lines. In the case of weak photospheric lines, such as Sr I 4607 Å, the CRD assumption has always been considered a very good approximation, considering that these lines form deep in the photosphere, where the contribution of R^{III} is not negligible with respect to that of R^{II}, and that their intensity and polarization profiles do not show wings outside the Doppler-core region. A quantitative comparison between CRD and AD calculations for this line, which has never been reported in the literature, is presented in Sect. 7.4.6.

The RT problem is solved using the method based on Krylov subspace iterations introduced in Benedusi et al. (2022). The AA approximation is only applied and analyzed in static scenarios because it is not suited to correctly handle the presence of a bulk velocity.

7.4.4 Results for Ca I 4227 Å

The Ca I 4227 Å is a strong chromospheric resonance line. When observed close to the limb, it shows a strong scattering polarization signal with large wing lobes produced by PRD effects. Clear polarization signals in the line core have also been observed in magnetically active regions close to the solar disk center and have been interpreted in terms of the FSHE (Bianda et al., 2011). The relatively large amplitude of these signals, the relevant amount of information that they encode, and the fact that they can be suitably modeled considering a simple two-level atom explain the great interest raised by this line, both for diagnostic purposes and for testing new theoretical and numerical approaches.

In Fig. 7.10 we show the intensity and Q/I profiles (first and second row, respectively) calculated in the FAL-C atmospheric model, in the absence of magnetic fields, for three inclinations of the LOS: $\mu = 0.169$, $\mu = 0.619$, and $\mu = 0.966$ (from left to right column). The first important result of these calculations is that the CRD assumption leads to an appreciable underestimate of the amplitude of the central Q/I peak for all the considered LOS. As expected, the Q/I wing lobes are completely lost under this approximation. The AA approximation provides very accurate results for the Q/I wing lobes but not for the central peak, which differs from the reference AD solution in both amplitude and shape, as already shown in Janett et al. (2021a).

In Fig. 7.11 we show the Q/I, U/I, and V/I profiles (first, second, and third rows, respectively) calculated including a deterministic and spatially constant magnetic field. These calculations confirm that the wing lobes (both in Q/I and U/I) are correctly modeled using the AA assumption even in the presence of a magnetic field. We can also see that both the AA and CRD models fail in modeling the line-core peaks, and that the inaccuracies become particularly relevant for LOS close to the disk center ($\mu = 0.966$), where the FSHE produces relatively strong signals. Finally, both the CRD and AA approximation provide accurate results for the V/I profiles.

Figure 7.12 shows a comparison between the results obtained with and without a magnetic field using the AD model. The left and central columns clearly show the depolarization produced by the Hanle effect in the core of the Q/I profile, while in the right column, it is possible to observe the significant signal produced by the FSHE for a nearly vertical LOS (see also Sect. 7.3). Note also that the results in intensity are identical since they are not affected by the magnetic field.

In Fig. 7.13 we show the I and Q/I profiles obtained including a MSI magnetic field. We recall that, due to cancellation effects, in the presence of such a magnetic field, the Zeeman and MO effects do not produce any signature in the polarization profiles, while the Hanle effect only produces a depolarization in Q/I. Moreover, since an MSI magnetic field preserves the cylindrical symmetry, the FSHE does not appear in this case. This explains why the U/I and V/I signals are zero, the Q/I wing lobes have the same amplitude as in the unmagnetized case while the central peak is depolarized, and the signals disappear as the LOS approaches $\mu = 1$. Also, in this case, the AA and CRD approximations introduce some inaccuracies in the modeling of the central Q/I peak.

Fig. 7.14 shows the results obtained with the Bifrost atmospheric model, in the absence of bulk velocity, but including a height-dependent magnetic field. The impact of the latter, through the joint action of the Hanle, MO, and Zeeman effects is qualitatively similar that that observed in the FAL-C model, in the presence of a height-independent magnetic field. Figure 7.15 is obtained including also the vertical component of the model's bulk velocity. As expected, velocity gradients induce significant enhancements in addition to distortions, and shifts in the scattering polarization Q/I and U/I profiles, especially in the core. The wing lobes are instead very similar in the two cases.

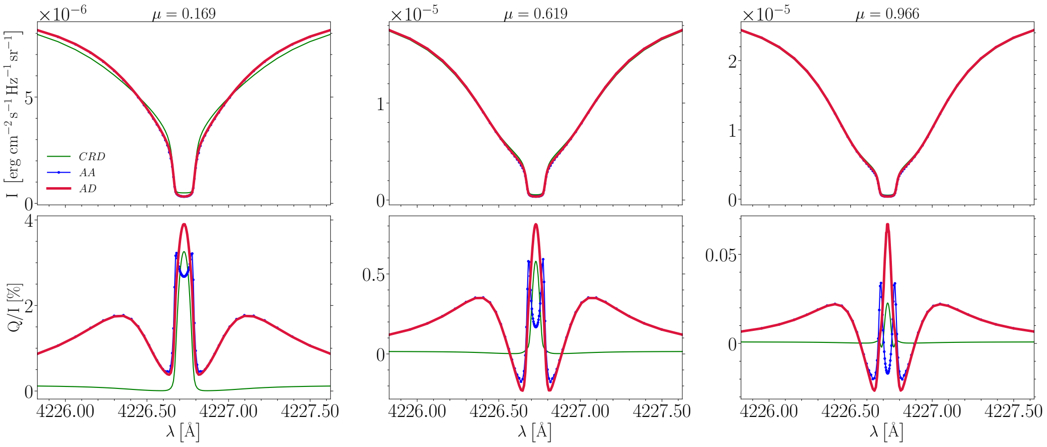

Figure 7.10. *Intensity (upper row) and Q/I (lower row) profiles of the Ca* I *4227Å line calculated in the FAL-C model for three different LOS: $\mu = 0.169$ (left column), $\mu = 0.619$ (central column) and $\mu = 0.966$ (right column). Each plot compares AD, AA, and CRD results (see legend). No magnetic fields and bulk velocities are present. The LOS lies in the $x - z$ plane of the considered reference system; the reference direction for positive Stokes Q is parallel to the y-axis.*

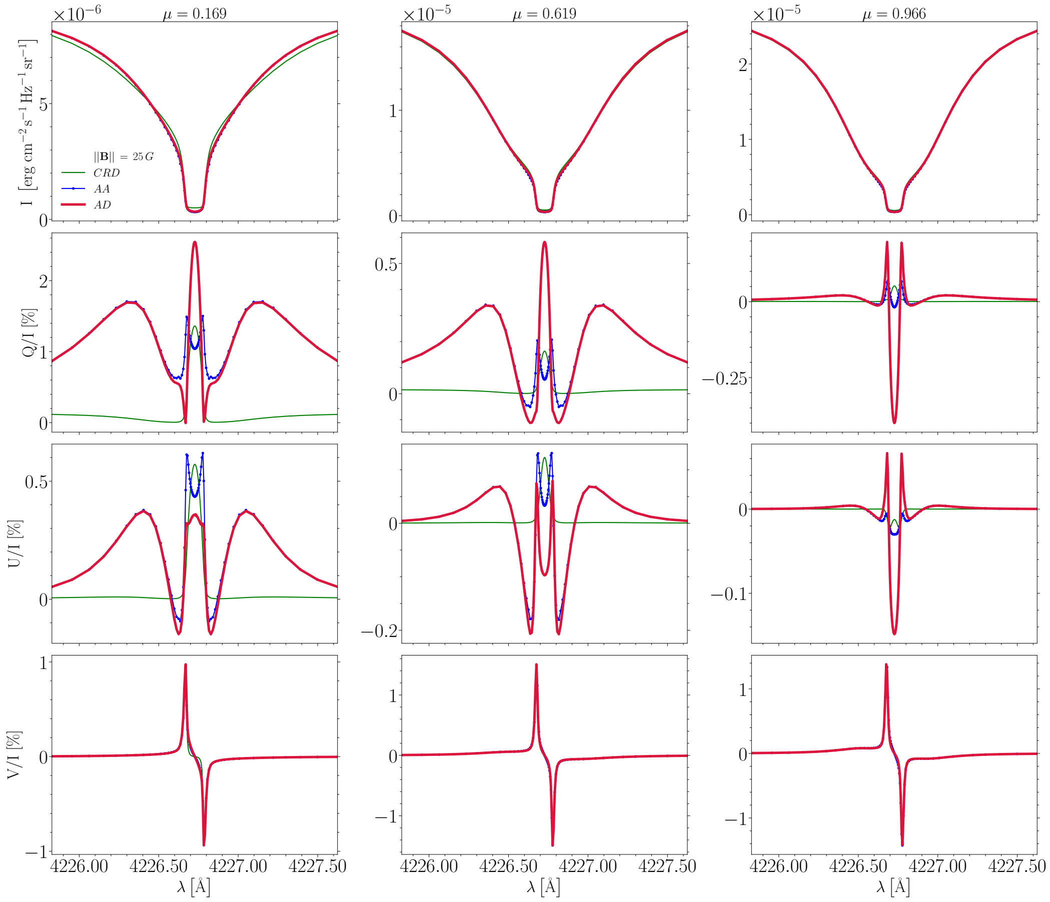

Figure 7.11. *Same as Fig. 7.10, but including a height-independent magnetic field of 25 G, with direction $\theta_B = \pi/4$, and $\chi_B = \pi/4$.*

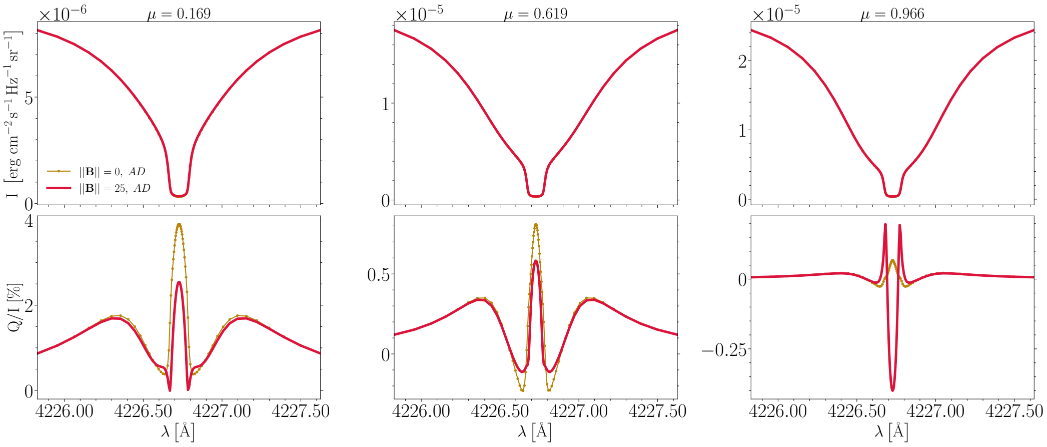

Figure 7.12. *Comparison of I (upper row) and Q/I (lower row) profiles for the Ca I 4227 Å line calculated in the FAL-C model in the most general AD case, in the absence of magnetic fields (dotted orange curves) and considering the same magnetic field as in Fig. 7.11 (red curves). All the other parameters are the same as in the previous figures.*

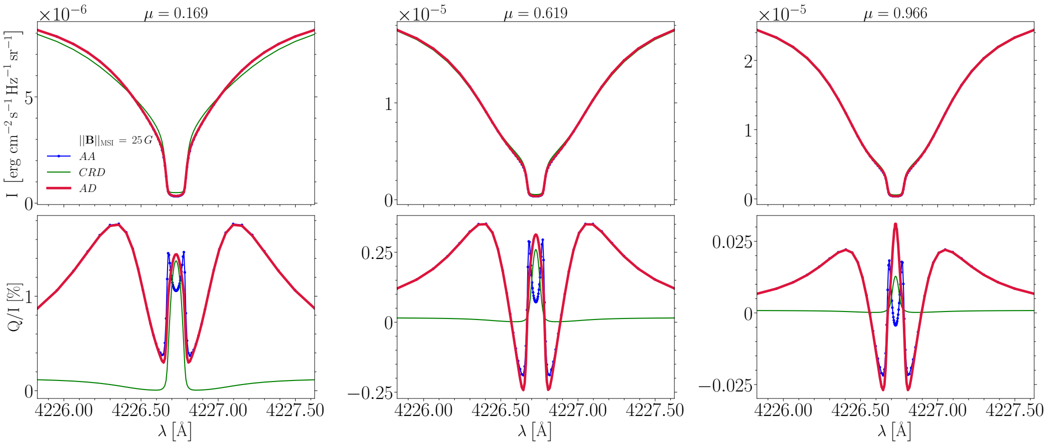

Figure 7.13. *Same as Fig. 7.10 but in the presence of a MSI magnetic field of 25 G.*

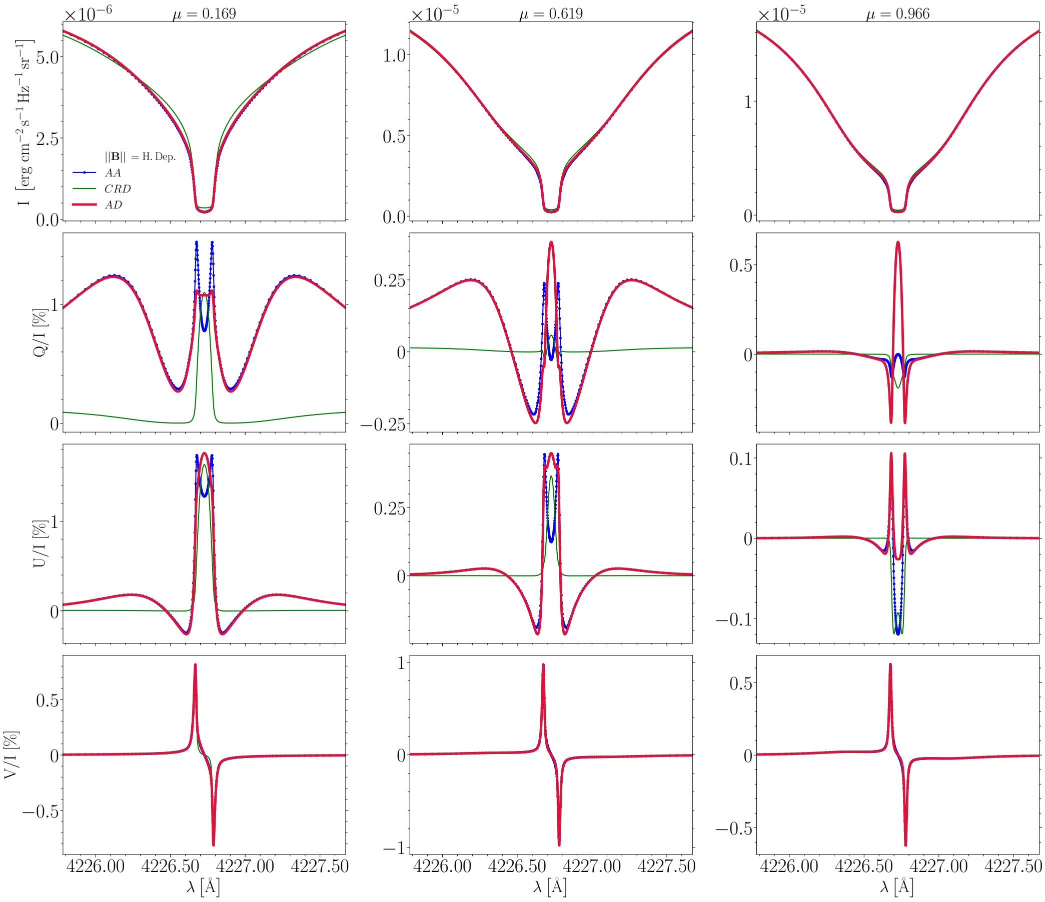

Figure 7.14. *Same as Fig. 7.10 but for the Bifrost model, including the model's magnetic field, but neglecting bulk velocities.*

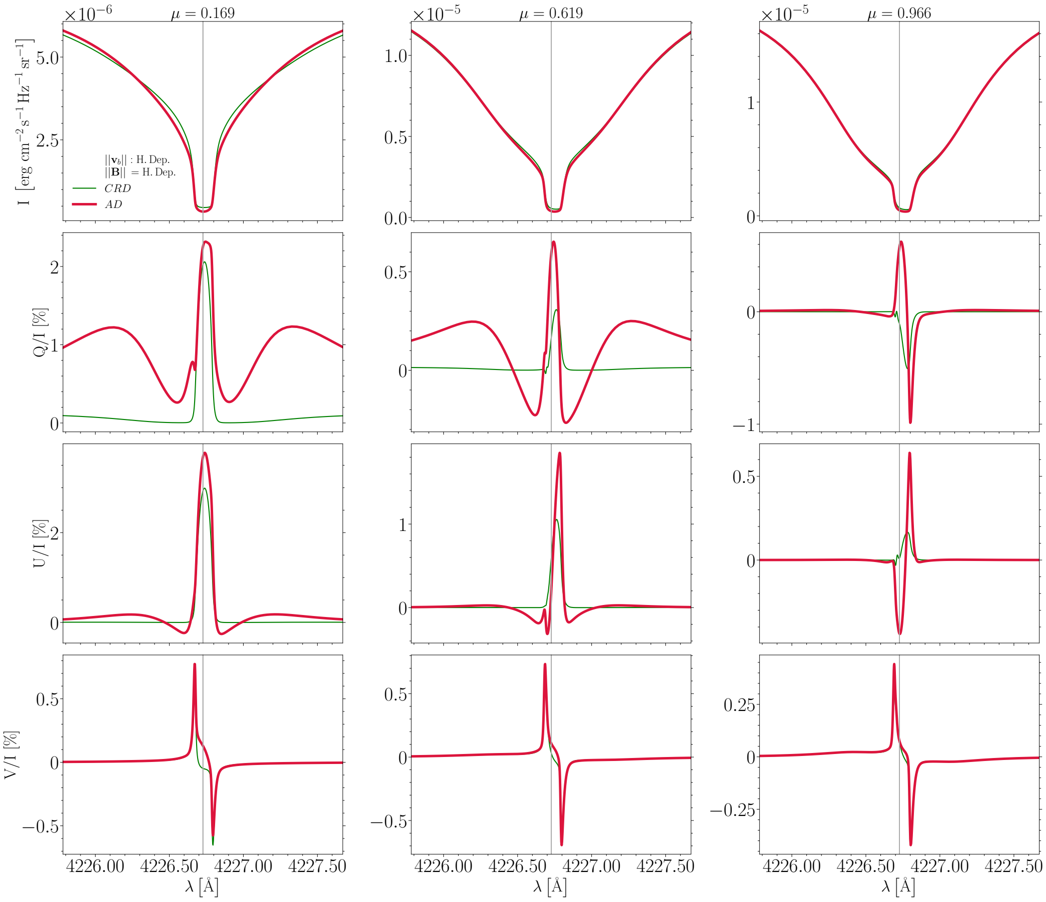

Figure 7.15. *Same as Fig. 7.14 but including bulk velocities.*

7.4.5 Results for Sr II 4078 Å

The Sr II 4078 Å is a chromospheric line originating from a $J_\ell = 1/2 \longleftrightarrow J_u = 3/2$ transition. When observed at the limb, it shows a strong scattering polarization signal with a triplet peak structure similar to that of the Ca I 4227 Å line, but with less extended wing lobes.

Fig. 7.16 shows that our PRD calculations correctly reproduce the triplet peak structure of the Q/I profile, and it highlights that (similarly to the Ca I 4227 Å line) the AA and the CRD models introduce some inaccuracies in the line core peak. These inaccuracies become even more pronounced when a deterministic, inclined magnetic field is included (see Fig. 7.17), especially as the LOS is close to $\mu = 1$ and the FSHE induces a clear polarization signal. As in the Ca I 4227 Å line, the AA approximation provides accurate results for the wing lobes, both in the absence and in the presence of magnetic fields.

Figure 7.18 highlights the impact of the magnetic field, mainly through the Hanle effect, on the scattering polarization profiles. At the limb, it produces a depolarization of the line-core peak, while close to the disk center it produces a clear signal through the FSHE. No clear signatures of MO effects in the wing lobes are appreciable. These results are qualitatively similar to those obtained for the Ca I 4227 Å line.

From the results obtained from the Bifrost model with and without bulk velocities (see figures 7.19, and 7.18) we can infer the same conclusions as in the case of the Ca I 4227 Å line.

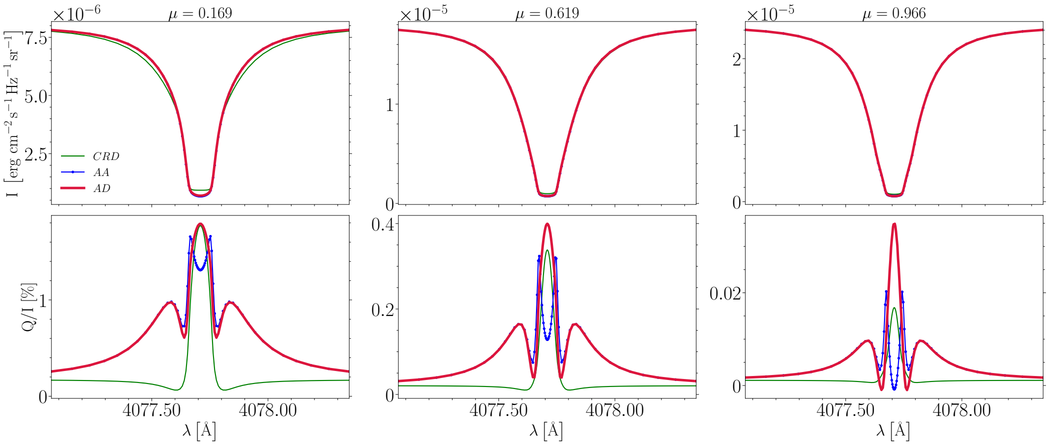

Figure 7.16. *Intensity (upper row) and* Q/I *(lower row) profiles of the Sr* II *4078 Å line calculated in the FAL-C model for three different LOS:* $\mu = 0.169$ *(left column),* $\mu = 0.619$ *(central column) and* $\mu = 0.966$ *(right column). Each plot compares AD, AA, and CRD results (see legend). No magnetic fields and bulk velocities are present. The LOS lies in the* $x-z$ *plane of the considered reference system; the reference direction for positive Stokes* Q *is parallel to the* y*-axis.*

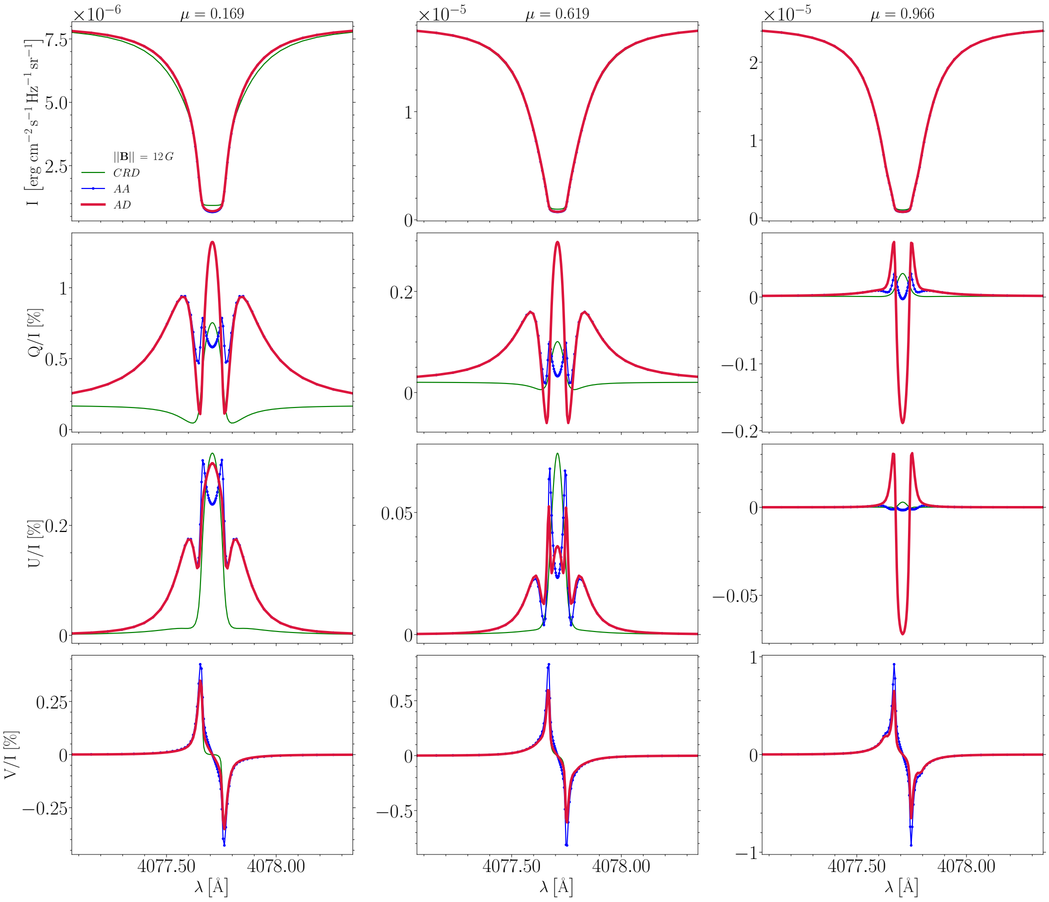

Figure 7.17. *Same as Fig. 7.16 but including a magnetic field of 12 G with direction $\theta_B = \pi/4$, and $\chi_B = \pi/4$.*

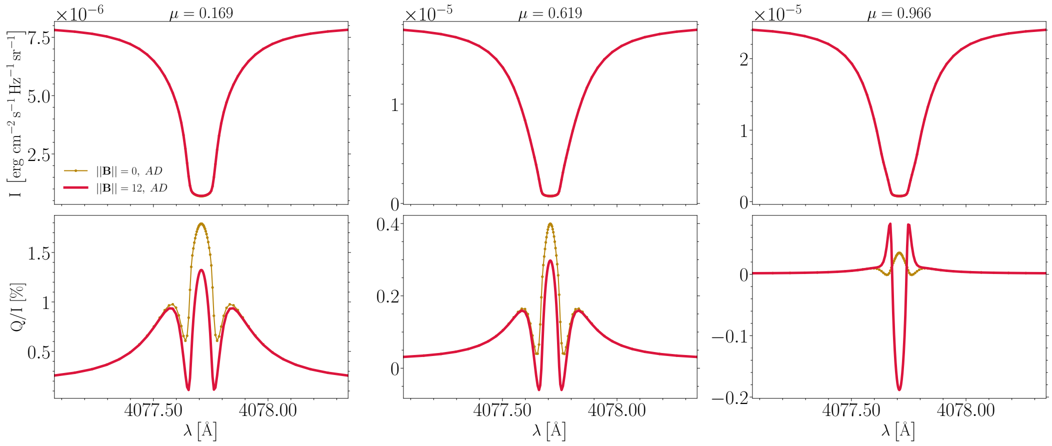

Figure 7.18. *Comparison of* I *(upper row) and* Q/I *(lower row) profiles for the Sr* II *4078 Å line calculated in the FAL-C model in the most general AD case, in the absence of magnetic fields (dotted orange curves) and considering the same magnetic field as in Fig. 7.17 (red curves). All the other parameters are the same as in the previous figures.*

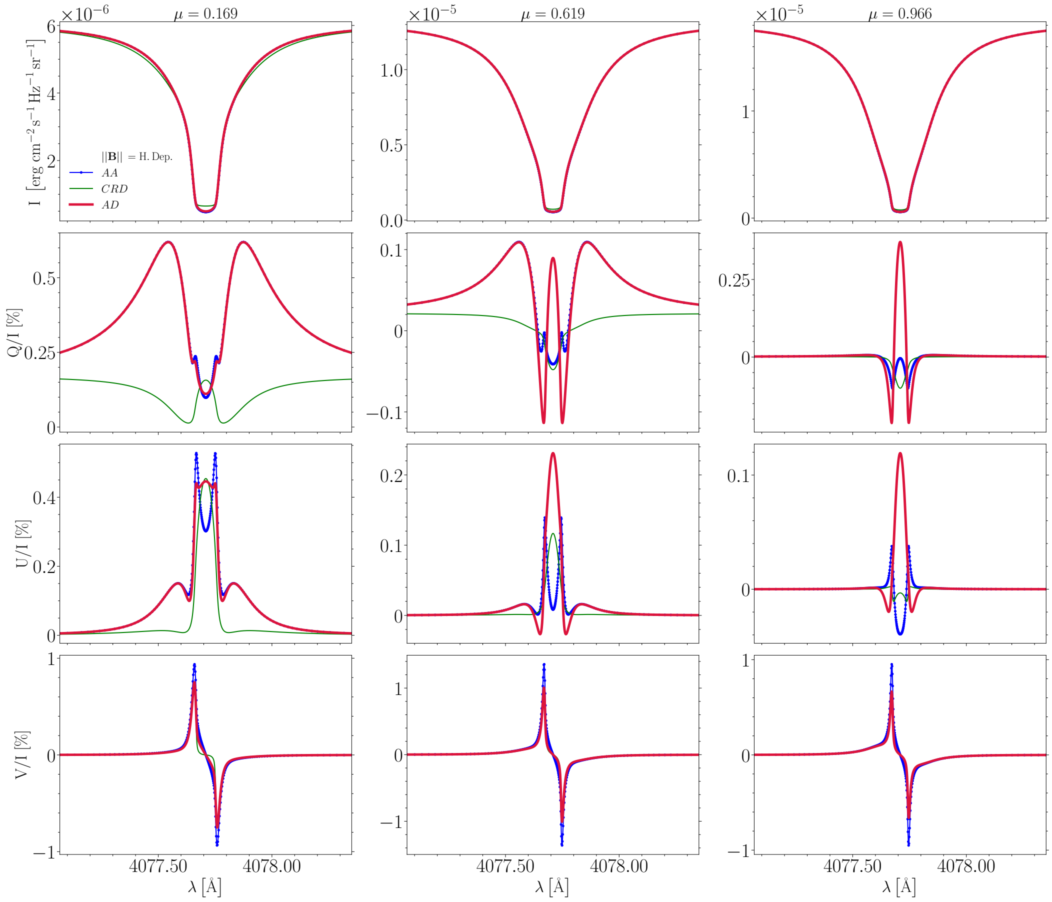

Figure 7.19. *Same as Fig. 7.16 but for the Bifrost model, including the model's magnetic field, but neglecting bulk velocities.*

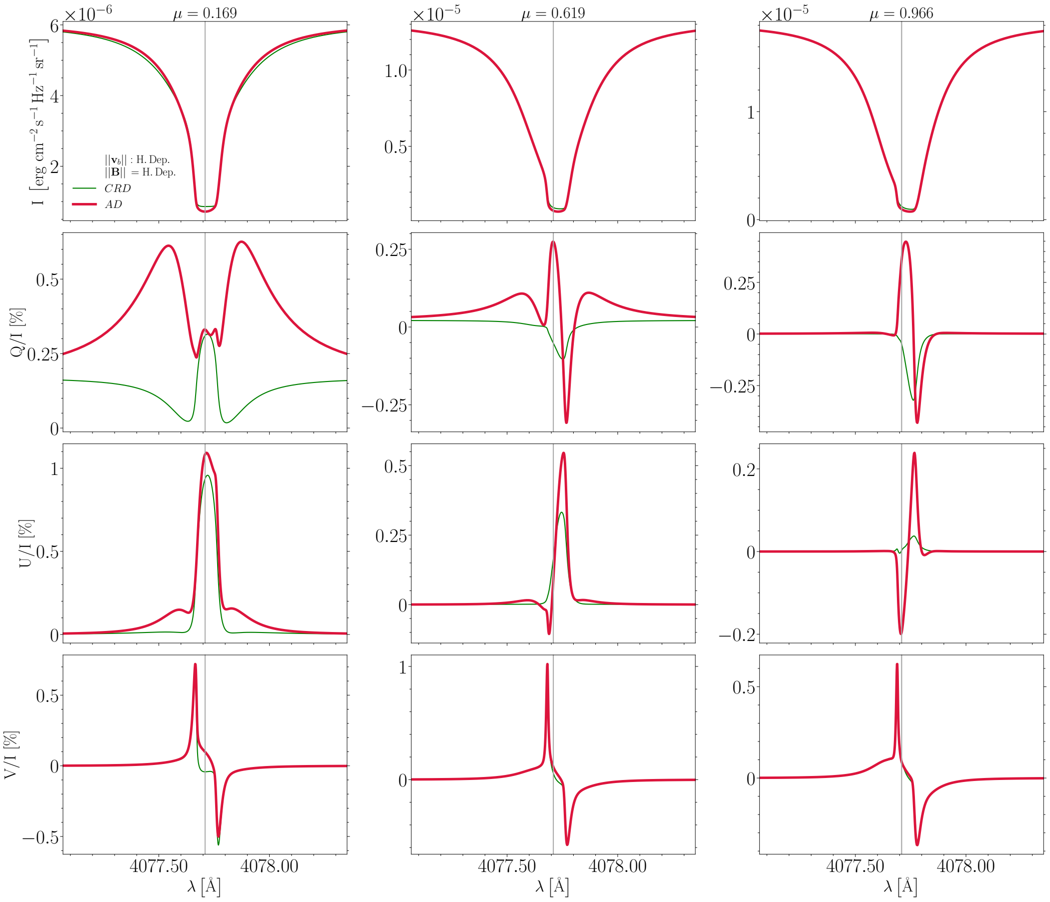

Figure 7.20. *Same as Fig. 7.19 but including bulk velocities.*

7.4.6 Results for Sr I 4607 Å

The Sr I 4607 Å line is a weak line that forms in the photosphere where the collisional rates are relatively high and the contribution of R^{III} is dominant (see Fig. 7.9). In this case, it can be expected that the CRD model is a good approximation. For the sake of completeness, we also report the results obtained with the AA model. As for the previous spectral lines, we present calculations performed in the FAL-C model, both in the unmagnetized case (see Fig. 7.21) and including height-independent deterministic (see Fig. 7.22) and MSI (see Fig.7.23) magnetic fields of 23 G, and in the Bifrost model, both neglecting (see Fig.7.24) and including (see Fig. 7.25) the vertical component of the model's bulk velocity field. From these results, it can be observed that, in general, the CRD and AA models provide reliable results. Observing that in the absence of a magnetic field, the R^{CRD} and $R^{\text{III-CRD}}$ are analytically equivalent, the small differences observed in Fig. 7.21 can be attributed to the contribution of R^{II}, which is small but not totally negligible in the case of Sr I 4607 Å. Significant differences appear only for relatively weak polarization signals.

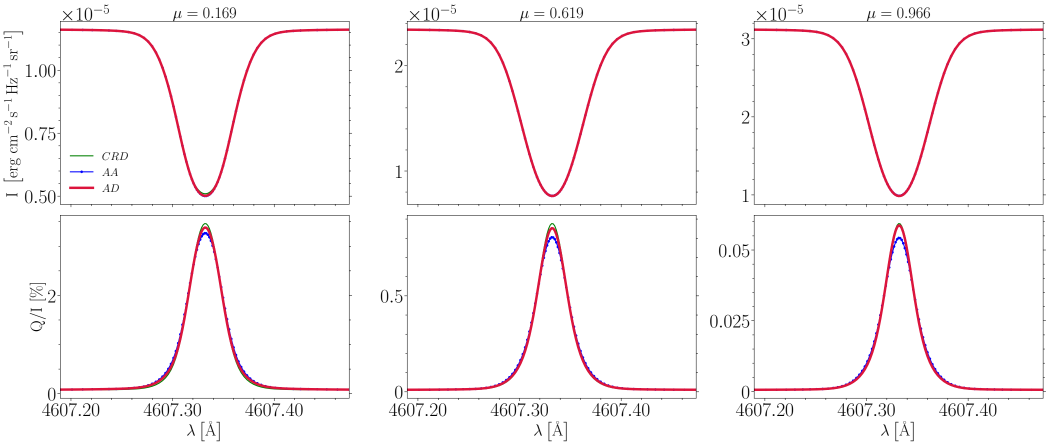

Figure 7.21. *Intensity (upper row) and Q/I (lower row) profiles of the Sr I 4607 Å line calculated in the FAL-C model for three different LOS under consideration in the absence of a magnetic field. The other parameters are the same as in Fig. 7.10.*

Since the Sr I 4607 Å line has been widely applied to investigate the small-scale magnetic fields that permeate the photosphere (Trujillo Bueno et al., 2004), it is an important result to confirm that the CRD model provides accurate results in the presence of an MSI magnetic field (see Fig. 7.23).

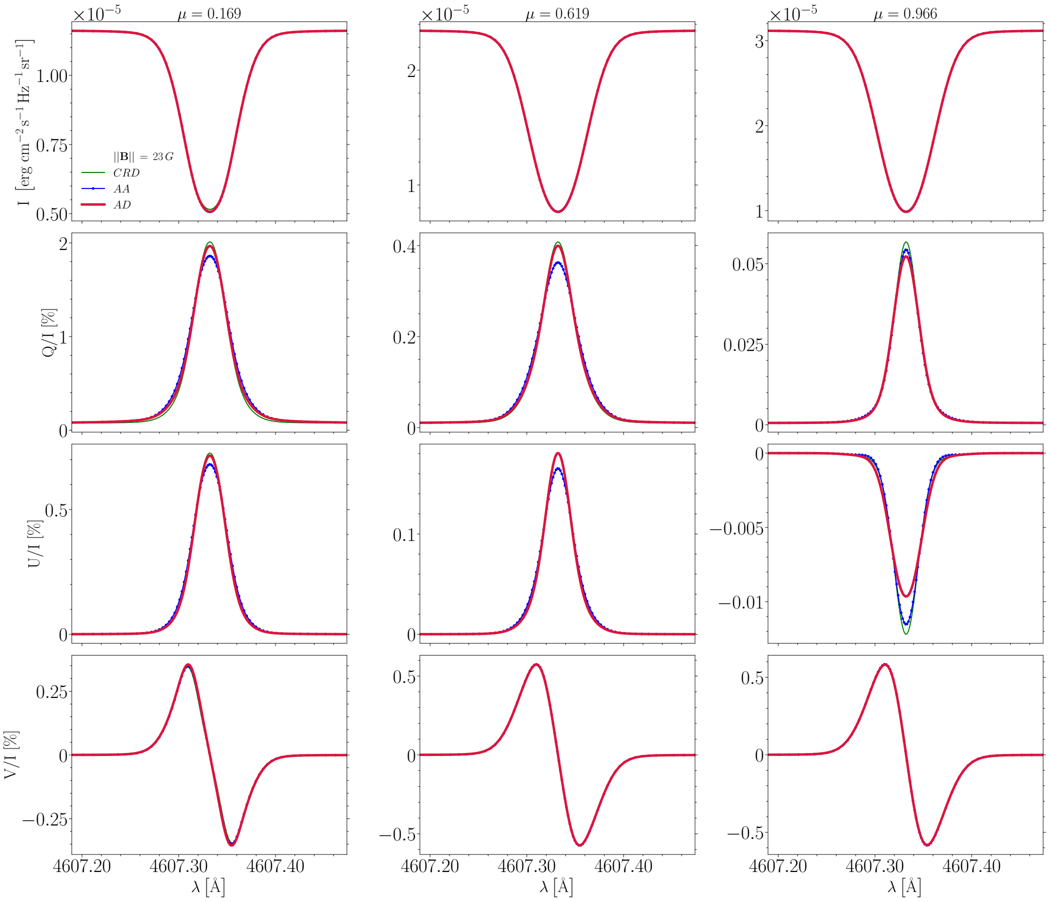

Figure 7.22. *Same as in Fig. 7.21 but with a magnetic filed of 23 G and a direction of $\theta_B = \pi/4$, and $\chi_B = \pi/4$.*

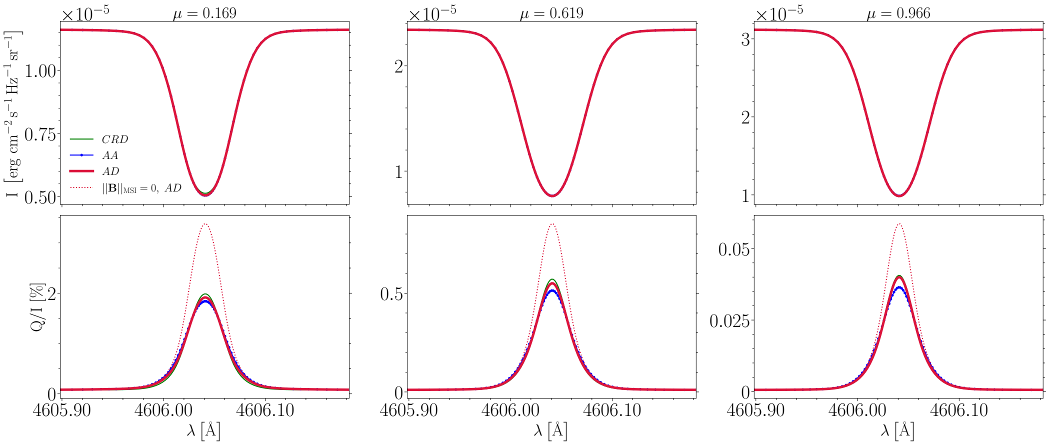

Figure 7.23. *Same as Fig. 7.21 but in the presence of a MSI magnetic filed (solid lines). For comparison, we added the PRD case in the absence of a magnetic field (red dotted line).*

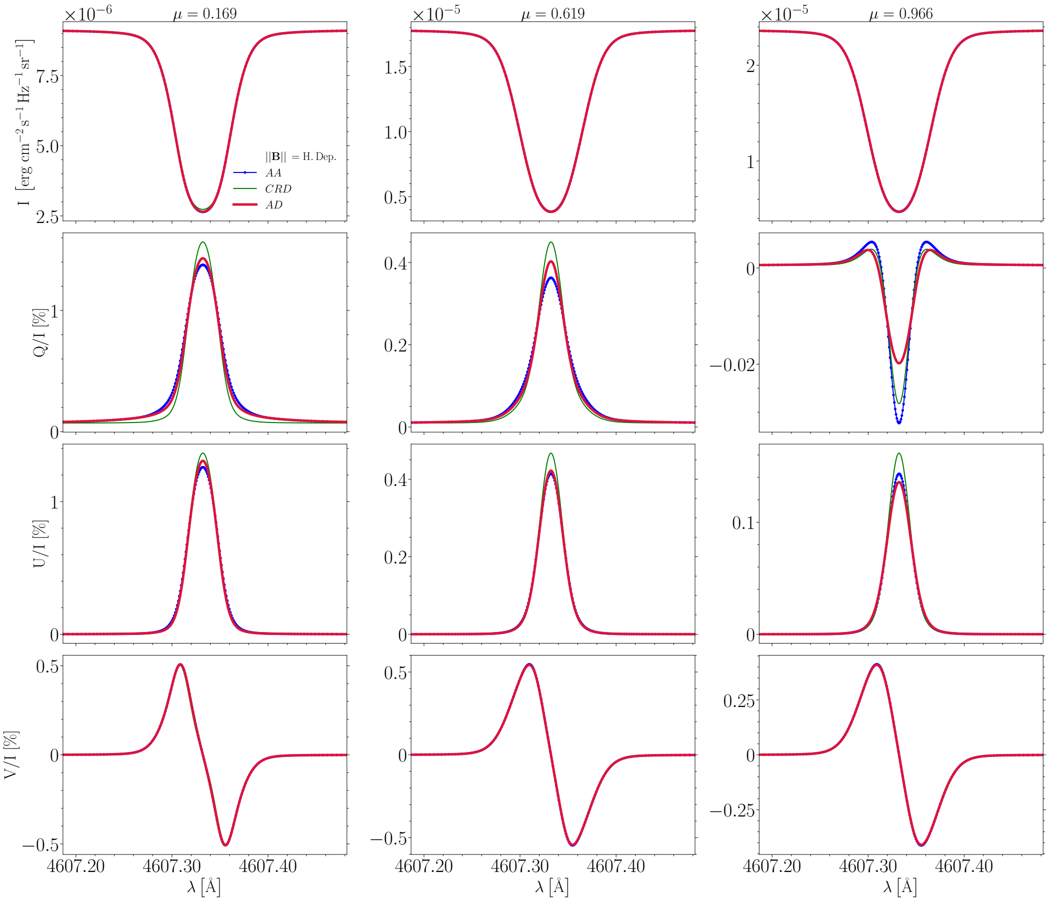

Figure 7.24. *Same as Fig. 7.21 but for the Bifrost model, including the model's magnetic field, but neglecting bulk velocities.*

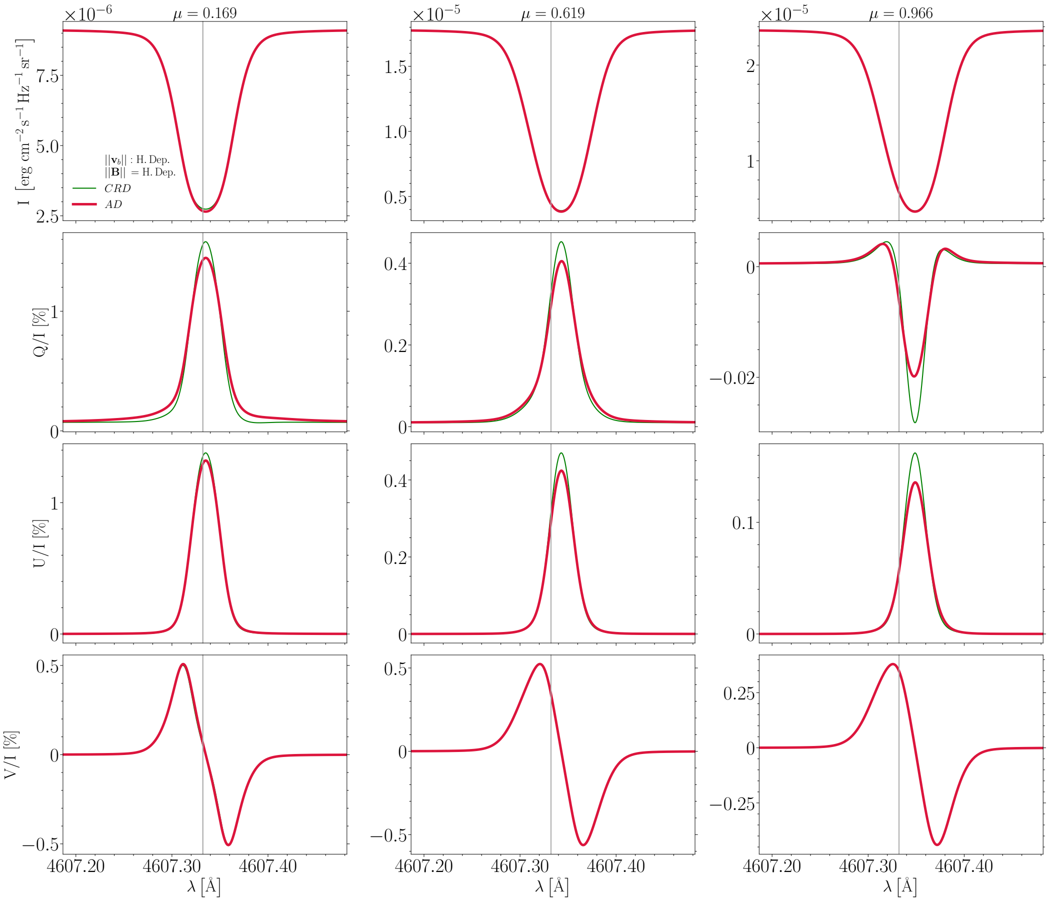

Figure 7.25. *Same as Fig. 7.24 but includes the bulk velocities.*

Remarkable exceptions in the Sr I 4607

Here we report some exceptions to the results shown above, that is a series of synthetic emergent profiles of the Sr I 4607 Å line where we found large relative discrepancies, both in shape and magnitude, between the reference AD case and the AA and CRD models. We point out that such exceptions only appear in case of rather weak Q/I and U/I polarization signals (i.e., $< 0.3\%$), and only when a deterministic magnetic field is included in the calculation.

Since the Sr I 4607 Å line forms in the photosphere, the observed scattering polarization signal can usually be well reproduced under the assumption of CRD and it is mainly used to investigate the small-scale unresolved magnetic fields (e.g., MSI) present in this layer. However, the exceptions shown below are of scientific interest. Indeed, (Zeuner et al., 2022) have recently detected, for the first time in this line, the signatures of a deterministic magnetic field, and the ZIMPOL polarimeter (Ramelli et al., 2010) can measure very weak polarization signals, with amplitudes of the same order of magnitude as those of the synthetic profiles shown in this section.

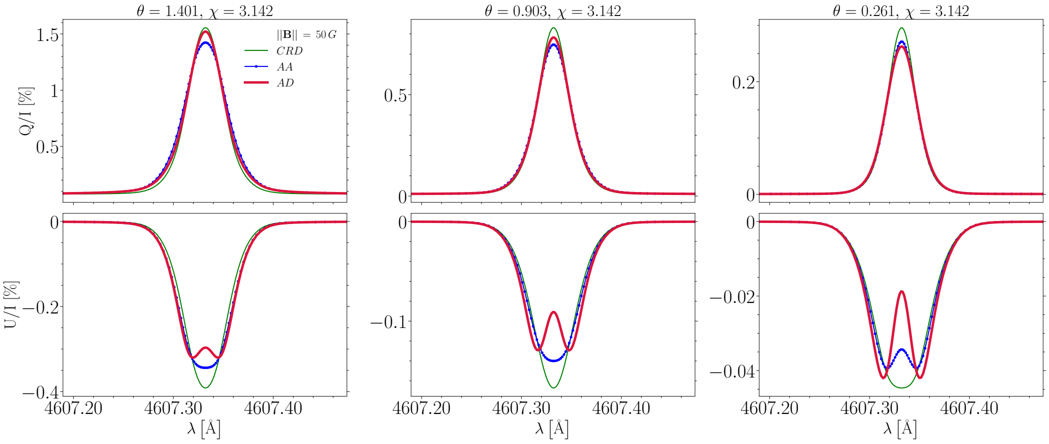

Figure 7.26. *Emergent Q/I (upper panel) and U/I (lower panel) profiles for the Sr I 4607 Å line resulting from the FAL-C atmospheric model with a uniform magnetic field of 50 G with direction $\theta_B = \pi/4$ and $\chi_B = \pi/4$, for three different directions of the LOS. The polar angles of the LOS are indicated in the titles of each column. The inclinations are the same as in the previous figures of this chapter, the azimuth is $\chi = \pi$. Each panel compares the AD, AA, and CRD profiles (see legend).*

Figure 7.26 shows the results obtained considering a magnetic field of 50 G with $\theta_B = \pi/4$ and $\chi_B = \pi/4$, for three different LOS, all having azimuth $\chi = \pi$. In this Figure, it is possible to observe clear discrepancies in U/I that increase as the LOS gets closer to the center of the solar disk (i.e. the inclination θ gets closer to 0). We can also see that the relative discrepancy increases as the absolute amplitude of the polarization signal decreases.

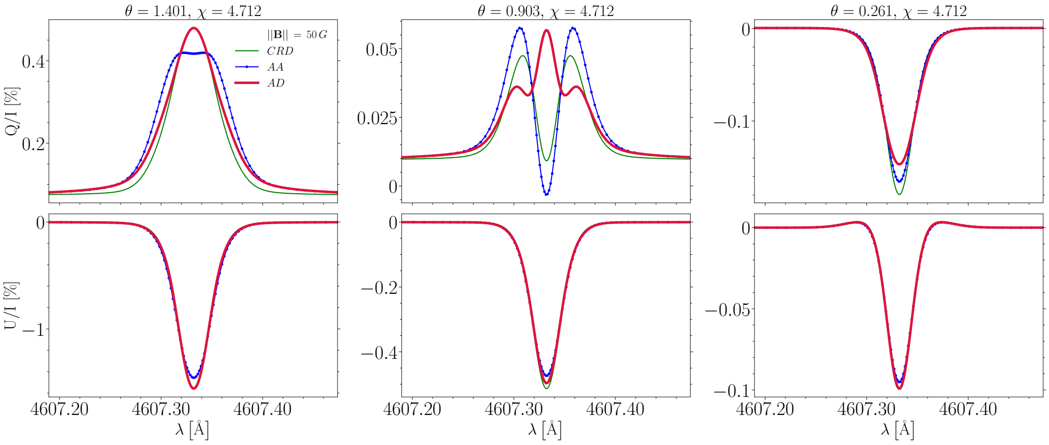

Figure 7.27. *Same as Fig. 7.26 but for LOSs with azimuth $\chi = 3/2\pi$.*

The profiles of Fig. 7.27 have been obtained for the same setting as in Fig. 7.26, but considering LOSs with azimuth $\chi = 3\pi/2$. In this case, we first observe a small discrepancy in Q/I for the near-limb LOS in the results obtained with the AA model (left column, blue line). This can be attributed to the inability of this model to correctly synthesize the line core, as pointed out in Janett et al. (2021a). It can be noticed that moving from the limb to the disk center, the Q/I signal changes its sign. The largest discrepancy is found for the intermediate LOS of $\theta = 0.903$, where the AD profile shows a weak positive peak at the line center, while the CRD and AA profiles show a strong depolarization, reaching negative values in the AA case. The U/I signals are the signature of the Hanle rotation produced by the deterministic magnetic field. The U/I signal calculated at the limb is *qualitatively* similar to the one detected by Zeuner et al. (2022) with ZIMPOL (see also Fig. 7.4).

The profiles of Fig. 7.28 have been calculated with the same setting as for Fig. 7.27, but for a magnetic field of 23 G and considering LOSs with azimuth $\chi = 5\pi/4$. In this case, a remarkable discrepancy is found in Q/I for a LOS close

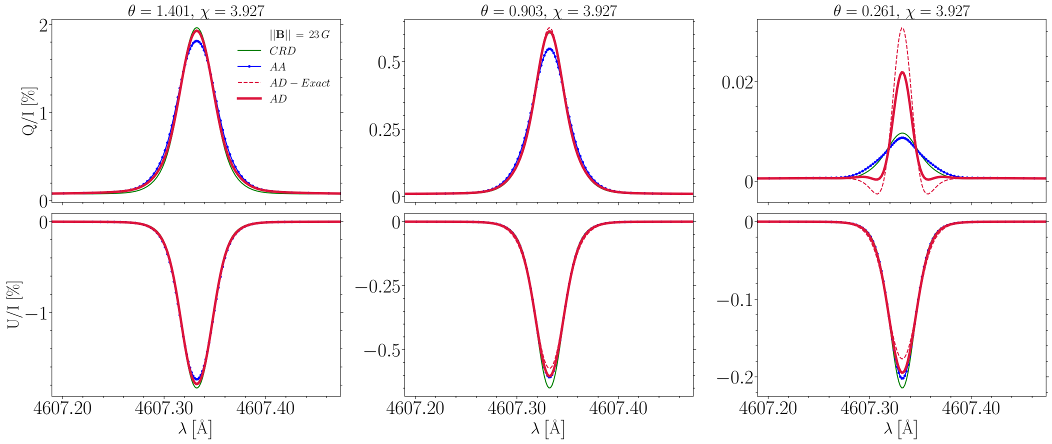

Figure 7.28. *Same as Fig. 7.27 but for a magnetic field strength of 23 G (corresponding to the Hanle critical field for this line, see Tab. 7.1), and a LOS with azimuth $\chi = 5\pi/4$. The dashed red lines are calculated by applying the exact angle-dependent expression of the R^{III} redistribution matrix (see also Chap. 6 and Riva et al. (2023)).*

to the disk center (right column). This weak signal is produced by the FSHE (see also Sect. 7.5). The reference AD calculation (red solid profile) shows a sharp peak at the line center and two negative local minima in the near wings. The AA and CRD calculations (blue and green lines, respectively) show instead a lower central peak and no local minima. Only for this figure, we also report the profiles obtained considering the exact angle-dependent expression of R^{III} (red dashed line), in which the signal produced by the FSHE is even more pronounced (see Chapter 6 and Riva et al. (2023)).

7.5 Focus on the Forward Scattering Hanle Effect

The results reported in the previous sections showed that quite large discrepancies between the reference AD case and the AA and CRD models are often found in the Q/I and U/I signals produced by the FSHE for LOS close to the disk center. This section aims at investigating this scenario in more detail.

As explained in Sect. 7.3.2, the presence of a deterministic non-vertical magnetic field breaks the axial symmetry of the radiation field and can induce appreciable scattering polarization signals also at the solar disk center, via the FSHE mechanism (Trujillo Bueno, 2001). Observational studies have highlighted this phenomenon in different lines, such as He I 10830 and Ca I 4227 Å (Trujillo

Bueno et al., 2002; Bianda et al., 2011), especially in regions where moderate deterministic magnetic fields are present. The observation reported in Fig. 7.5 shows an example of a clear signal produced by the FSHE in the Ca I 4227 Å line. The weak signal observed in the Sr II 4078 Å line reported in Fig. 7.6 can also be attributed to the FSHE.

The calculations reported in the previous sections included signals produced by the FSHE in the presence of a non-vertical deterministic magnetic field. These can be seen, for example, in Fig.7.11 for the Ca I 4227 Å line and in Fig. 7.17 for the Sr II 4078 Å line, for a LOS with $\mu = 0.966$. In the Ca I 4227 Å line, the FSHE produces a clear peak in the line core, in both Q/I and U/I, flanked by two secondary weaker peaks of opposite signs in the near wings. In general, when an inclined deterministic magnetic field is present, the amplitude of the scattering polarization signal at the line center first decreases as the observed point moves away from the solar limb (i.e., in the interval $0 \leq \mu \lessapprox 0.5$) and then it grows again, due to the FSHE, as the LOS approaches the center of the solar disk (i.e., $\mu \rightarrow 1$). The efficiency of the FSHE is maximal for horizontal magnetic fields, while it disappears as the magnetic field direction approaches the vertical (i.e., parallel to the z-axis).

To analyze the suitability of the AA and CRD approximations to model the FSHE, we performed a series of calculations for the Ca I 4227 Å line in the FAL-C model, considering the case of a horizontal magnetic field with a strength of 20 G, an inclination of $\theta_B = \pi/2$, and an azimuth χ_B ranging from 0 to 2π with a step size of $\pi/4$. With these results, we made the *polarization diagram* for a LOS of $\mu = 0.966$ shown in Figure 7.29, where we compare the amplitudes of the Q/I and U/I signals at the central line frequency of the AD results (red curve in the right panel) with the results obtained with the simplified AA model (blue curves in the left and right panels) and with the results obtained with the CRD model (green curve in the left panel). From the polarization diagram, it is evident that both the AA and CRD models significantly underestimate the amplitude of the polarization signals at the central line frequency. This behavior is confirmed in the polarization diagram 7.30, which was derived by applying a non-horizontal magnetic field with a strength of 25 G, an inclination of $\theta_B = \pi/4$, and a varying azimuth as in the previous test.

The second numerical analysis is presented in Figure 7.31, where we show the variation of the amplitudes of Q/I and U/I at the central line frequency as a function of the inclination, μ, of the LOS (i.e., the center-to-limb variation, CLV), **a)** for a horizontal magnetic field of 20 G with $\theta_B = \pi/2$ and $\chi_B = 0$ (left column), and **b)** for a magnetic field of 25 G (corresponding to the Hanle critical field, see Tab. 7.1) with a direction of $\theta_B = \pi/4$ and $\chi_B = \pi/4$ (right column). The left

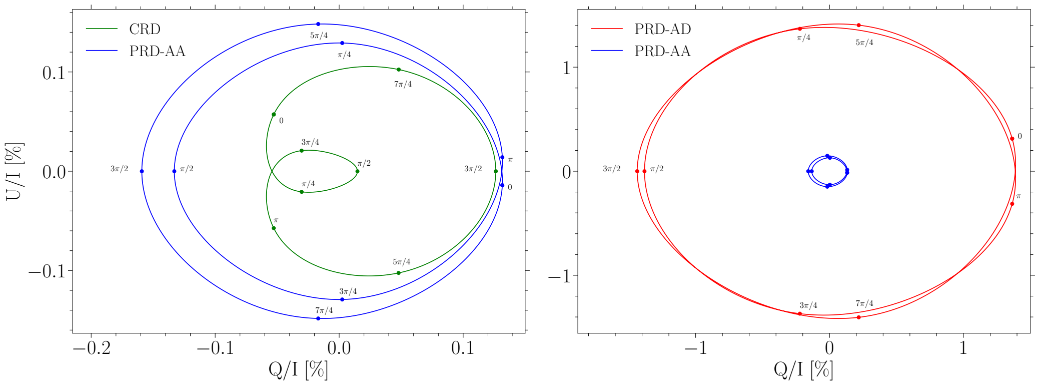

Figure 7.29. *Polarization diagrams for the radiation at the line-center frequency of the Ca* I *4227 Å line, with a LOS of* $\mu = 0.966$*. Calculations are performed in the FAL-C atmospheric model, considering a height-independent horizontal* ($\theta_B = \pi/2$) *magnetic field of 20 G. The points of each curve correspond to different magnetic field azimuths* $\chi_B \in (0, 2\pi)$*. The points corresponding to* $\chi_B = n\,\pi/4$ *(for* $n = 0, \ldots, 7$*) are marked with small circles. Left panel: comparison between CRD and PRD-AA calculations. Right panel: comparison between PRD-AA and PRD-AD calculations. The reference direction for positive Stokes Q is parallel to the* y*-axis of the considered reference system.*

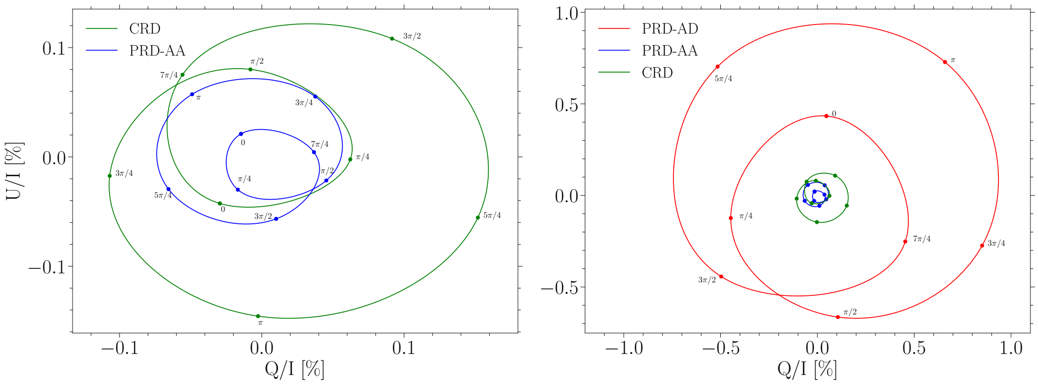

Figure 7.30. *Same as Fig. 7.29 but for a magnetic field of 25 G with an inclination of* $\theta_B = \pi/4$.

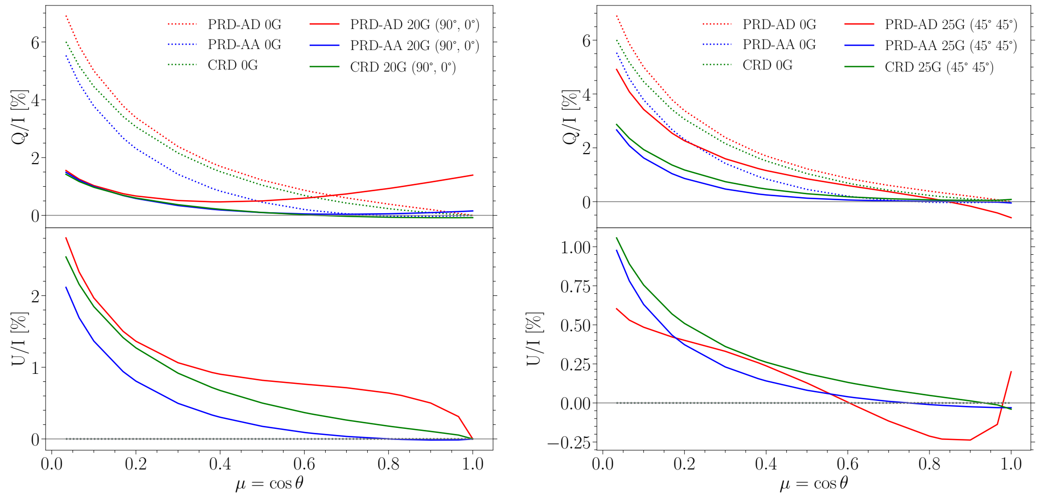

Figure 7.31. *Theoretical center-to-limb variation of the amplitude of the core peaks of Q/I (upper panel) and U/I (lower panel) of the Ca* I *line at 4227 Å, obtained from CRD, PRD-AA, and PRD-AD, in the atmospheric model FAL-C, in the absence (dashed curves) and presence (solid curves) of magnetic fields. The left panel shows the result obtained by applying a horizontal magnetic field of 20 G with direction $[\theta_B = \pi/2, \chi_B = 0]$, while the right panels refer to results obtained with a magnetic field of 25 G and direction $[\theta_B = \pi/4, \chi_B = \pi/4]$. We recall that in the absence of a magnetic field, U/I is equal to zero.*

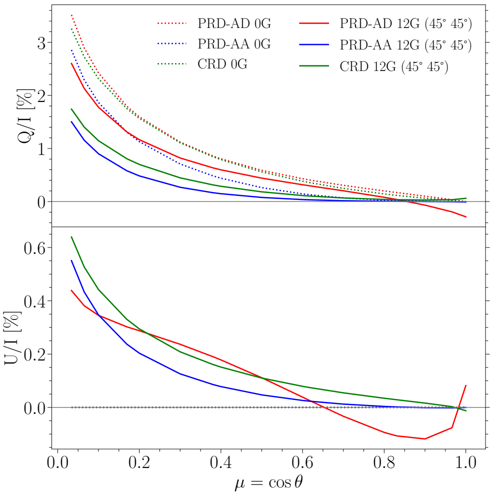

Figure 7.32. *Same as Fig. 7.31, but for the Sr* II *4078 Å line and considering a magnetic field of 12 G (Hanle critical field, see Tab. 7.1) with direction $[\theta_B = \pi/4, \chi_B = \pi/4]$.*

column of the figure 7.31 shows that, for this particular magnetic configuration, the amplitude of Q/I decreases up to an inclination of $\mu \approx 0.4$ and then increases as $\mu \rightarrow 1$ as a consequence of the FSHE. This increase is much more pronounced in the AD calculations. In the right column of the figure 7.31 it can be seen that the FSHE induces a change of sign in both Q/I and U/I when μ gets close to 1. In this case, significant differences between the various models appear both in the signal amplitude and in the value of μ at which the signal changes its sign. Furthermore, Figure 7.32 shows the CLV for Q/I, and U/I at the central line frequency in the Sr II 4078 Å line for a magnetic field of 12 G (Hanle critical field, see Table 7.1) with direction $\theta_B = \pi/4$ and $\chi_B = \pi/4$.

A comparison of the Q/I and U/I profiles of the Ca I 4227 Å line calculated at $\mu = 0.169$ and $\mu = 0.996$ for the magnetic fields considered in the left and right panels of Fig. 7.31 is provided in Fig. 7.33 and 7.34, respectively.

In conclusion, these results show that the AA and CRD approximations can introduce significant inaccuracies in the modeling of the FSHE, especially in strong chromospheric lines. The differences can be in the amplitude, shape, and sign of the polarization profiles, and are highly dependent on the exact orientation of the magnetic field. In some cases, differences in amplitude of more than one order of magnitude are found (see Fig. 7.33). In the case of photospheric lines, appreciable differences are also found between calculations performed with the exact angle-dependent expression of R^{III}, and calculations carried out with $R^{\mathrm{III-CRD}}$ (see Chap. 6).

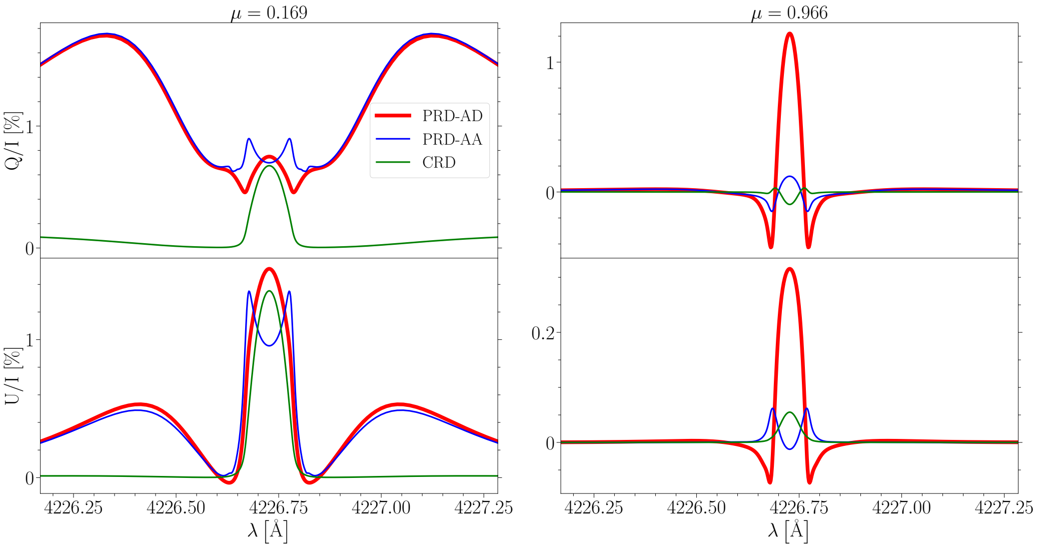

Figure 7.33. *Stokes Q/I (upper panel) and U/I (lower panel) profiles for the Ca* I *line at 4227 Å, at $\mu = 0.169$, and $\mu = 0.966$ (left and right columns respectively), obtained from CRD, PRD-AA, and PRD-AD calculations. Calculations are based on the FAL-C atmospheric model, with a height-independent horizontal ($\theta_B = \pi/2$, $\chi_B = 0$) magnetic field of 20 G.*

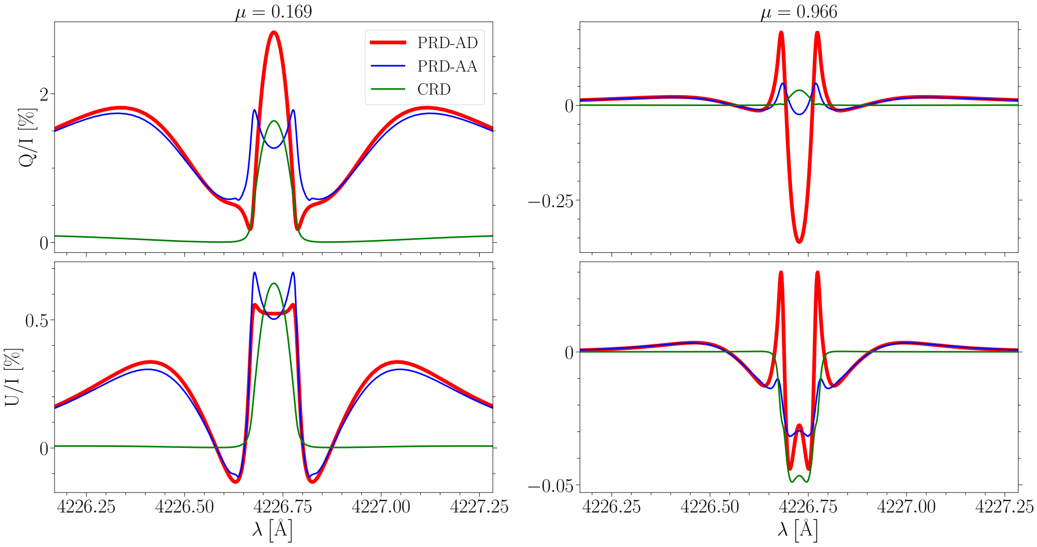

Figure 7.34. *Same as Figure 7.33 but for a magnetic field of 25 G with direction ($\theta_B = \pi/4$, $\chi_B = \pi/4$).*

7.6 Conclusion

In this chapter, we presented computational results for the intensity and polarization signals of the Ca I 4227 Å, Sr II 4078 Å, and Sr I 4607 Å lines, obtained in the most general AD case, and under the AA and CRD approximations. The calculations were carried out in 1D models of the solar atmosphere both semi-empirical and extracted from realistic 3D MHD simulations. The latter also included realistic height-dependent values of the magnetic and bulk velocity fields. Although a quantitative comparison with the observations was beyond the scope of this work, this analysis showed that our reference AD numerical results are physically reliable, since they reproduce many observed features and show a very good qualitative agreement with the measured profiles.

One of the main goals was to assess the suitability of the AA and CRD approximation, having the possibility to contrast them with accurate AD calculations. A very interesting result is that the AA model can correctly reproduce the wing lobes in the Ca I 4227 Å and Sr II 4078 Å lines, although it can introduce inaccuracies in the line core. As expected, the CRD assumption completely fails in reproducing the wing lobes of chromospheric lines. In the line core, it shows minor errors at the limb, but it fails in modeling the FSHE at the center of the solar disk.

The most remarkable finding of this chapter is the very large error that can be introduced by the AA and CRD approximations when modeling the FSHE in strong chromospheric lines, such as Ca I 4227 Å and Sr II 4078 Å lines. For particular geometries, the AA and CRD approximations can introduce significant errors also in the photospheric Sr II 4078 Å line, especially when a deterministic magnetic field is included in the calculations.

In conclusion, our results highlight that the AA and CRD approximations must be used with caution when modeling scattering polarization in chromospheric lines, and that AD calculations are needed in order to accurately model these signals and their magnetic sensitivity, and obtain physically reliable results. The much higher computational cost of AD modeling clearly poses a further challenge for the solution of the inverse problem for scattering polarization in chromospheric lines.

7.A Results for Mg II *k* 2795 Å

In this appendix, we present a series of calculations for the Mg II *k* 2795 Å line, which results from a transition $J_\ell = \frac{1}{2} \longleftrightarrow J_u = \frac{3}{2}$. It must be noted that a correct modeling of the scattering polarization signal of this line requires considering a two-term atomic model (i.e., an atomic model that includes also the nearby Mg II *h* line at 2800 Å and that accounts for quantum interference between the upper levels of the two lines). The results shown in this chapter, obtained with a two-level atomic model, are in any case correct for the line-core region (e.g. Belluzzi and Trujillo Bueno, 2012).

Since the Mg II *k* 2795 Å line is a very broad chromospheric line, we present the whole profile (over a spectral range of about 80 Å) in Figure 7.35, while the details of the core of the line are shown in Figure 7.36. The calculations were performed in the FAL-C atmospheric model with a height-independent magnetic field of 22 G (corresponding to the Hanle critical field for this line, see Eq. (7.1)), with a direction of $\theta_B = \pi/4$ and $\chi_B = \pi/4$, and a frequency grid of 221 nodes. Falling in the near-ultraviolet, this line can only be observed from space. The scattering polarization signal of this line was recently measured by two sounding rocket experiments: CLASP2 (2019) and CLASP2.1 (2021). The Q/I profile observed by CLASP2 is reported in Rachmeler et al. (2022).

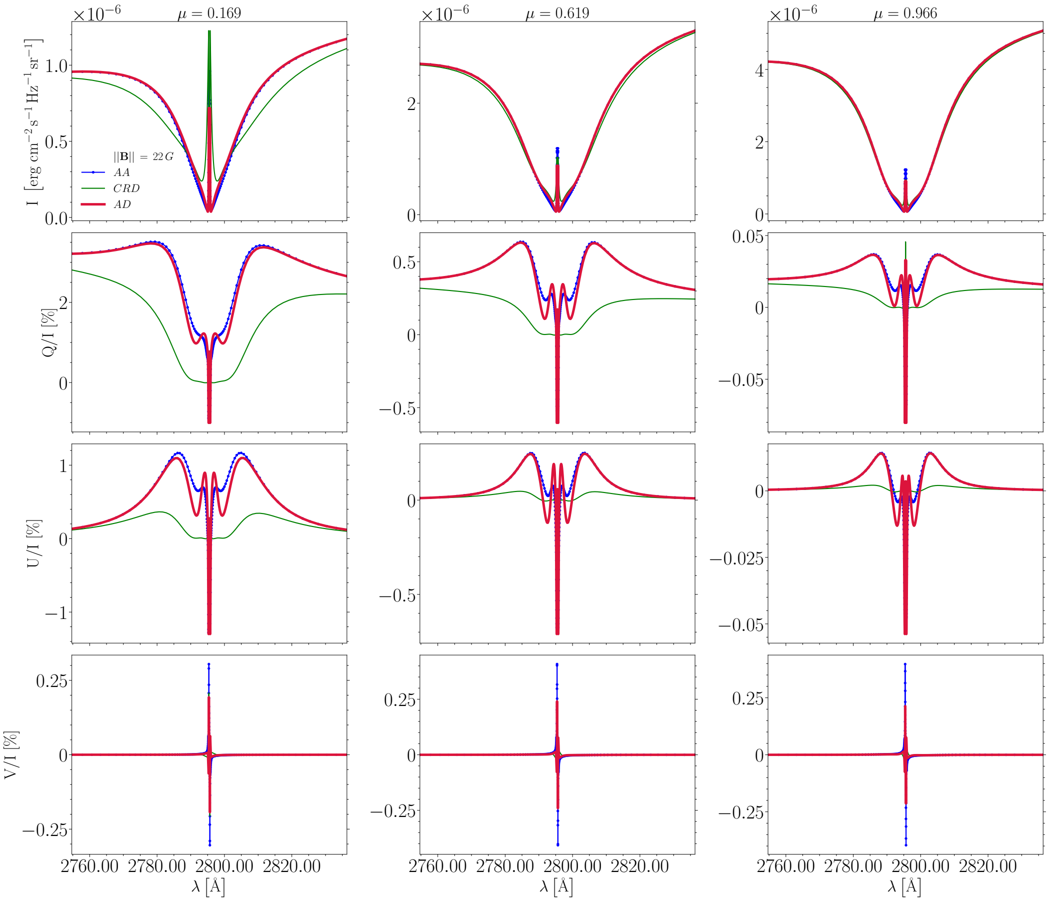

Figure 7.35. *Emergent Stokes I, Q/I, U/I and V/I (top to bottom) profiles of the Mg* II *k 2795 Å line, calculated in the FAL-C atmospheric model for three different LOS. A height-independent magnetic field of 22 G with direction $\theta_B = \pi/4$ and $\chi_B = \pi/4$ is included in the modeling. The figure compares AD, AA and CRD calculations (see legend).*

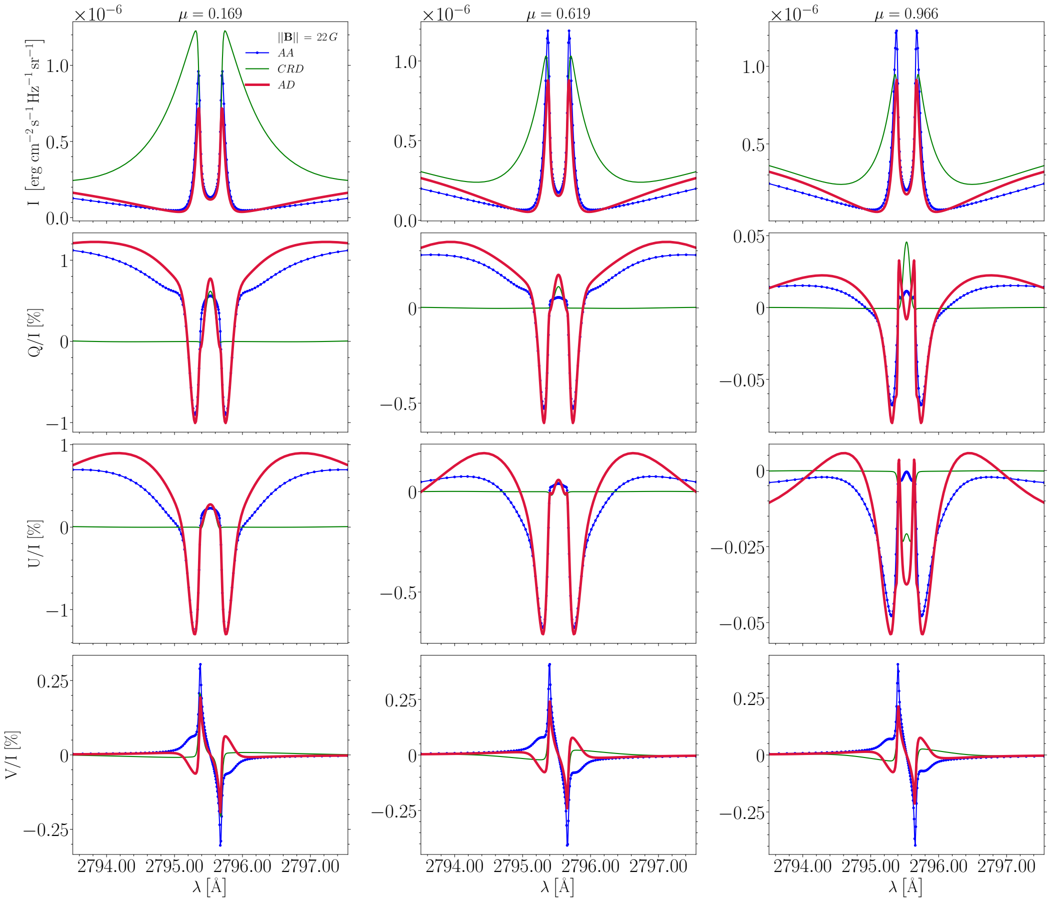

Figure 7.36. *Same as Fig. 7.35, but for a narrower spectral interval, showing the details of the line-core.*

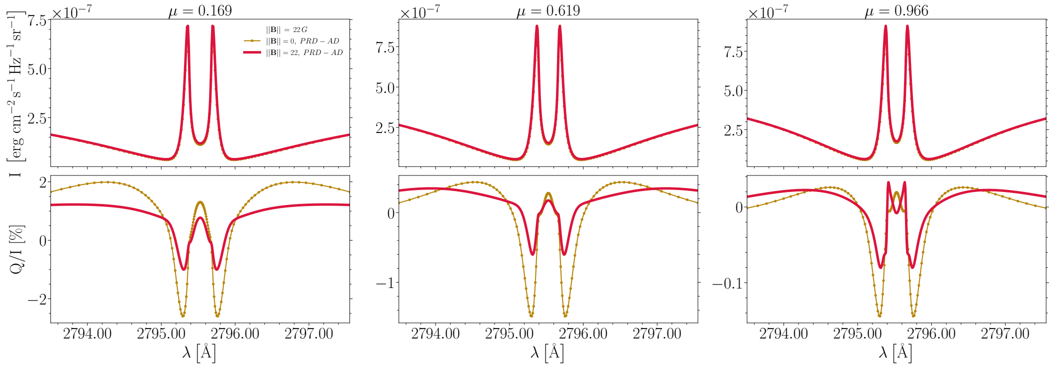

Figure 7.37. *Emergent I and Q/I profiles of the Mg* II *k 2795 Å, calculated in the FAL-C atmospheric model in the AD case, including (red lines) and neglecting (orange lines) the same magnetic field as in the previous figures.*

Conclusions

The main objective of this thesis work was to develop an efficient and accurate algorithm for calculating the emission vector of the RT equation, for a two-level atomic model, taking into account AD-PRD effects. The algorithm was needed as a key component for the development of the *first* code capable of solving the non-LTE RT problem for polarized radiation in realistic 3D models of the solar atmosphere, accounting for PRD effects. Such a code will allow modeling the polarization of strong resonance lines, which encode precious information on the elusive magnetic fields of the solar chromosphere, with unprecedented accuracy. The considered RT problem is extremely challenging from the computational standpoint, and, at the beginning of the project, it could only be solved under simplifying approximations, either on the atmospheric model (e.g., isothermal, optically thin, cylindrically symmetric) or on the description of scattering processes (e.g., the AA approximation, or the weak-field assumption).

The problem was formulated within the framework of the PRD theory of Bommier (1997a,b), using the redistribution matrix formalism. The first goal was to develop a fast computational method for calculating the contribution of the R^{II} redistribution matrix to the emission vector, considering arbitrary magnetic and bulk velocity fields. This is the most time-consuming component of the whole solution strategy. The first part of the work was dedicated to the analysis of the basic functional components of the R^{II} redistribution function and the identification of fast quadrature methods suited to the problem. The developed methods exploit the a priori knowledge of the properties of such functions and the regularities of the angular grid. Second, we studied CPU and GPU architectures and the roofline model to perform a robust algorithm analysis that allowed us to develop a near-optimal application. As a test bench, we used the chromospheric Ca I 4227 Å line.

Based on the experience gained with R^{II}, we also developed a method to compute the contribution of R^{III} in its exact form (which is notoriously challenging). This allowed us to analyze, in realistic atmospheric models, the reliability of simulations carried out considering the assumption of CRD in the observer's frame for R^{III}. The results confirmed the suitability of this approximation, especially in chromospheric lines (Riva et al., 2023).

The above-mentioned methods have been successfully integrated into novel codes that solve the full non-LTE RT problem for polarized radiation in both 1D and 3D atmospheric models. These codes implement new preconditioned Krylov solvers (Benedusi et al., 2022; Benedusi et al., 2023), which converge in a few iterations.

The 1D code allowed us to compare, for the first time, PRD-AD calculations of scattering polarization performed in semi-empirical models of the solar atmosphere, to PRD-AA and CRD calculations. The results highlighted a series of artifacts produced by the AA approximation, which had not been noticed in the past (Janett et al., 2021a). Particularly relevant are the differences found between AA and AD models in polarization signals induced by the *Forward-Scattering Hanle Effect* (FSHE).

These investigations clearly showed the importance of considering PRD effects in their general AD formulation in order to get reliable results that can be compared to the observations. The synthetic signals obtained in the Ca I 4227 Å, Sr II 4078 Å, and Sr I 4607 Å lines showed a very good qualitative agreement with observations carried out with ZIMPOL at IRSOL. In the case of Sr II 4078 Å our calculations for the FSHE predicted observable signals, which have motivated new dedicated observations.

The algorithm revealed to be sufficiently fast, accurate, and lightweight (in terms of memory requirements) to be successfully implemented in the 3D code. This code is presently tested in HPC applications. The first results were positive, showing, for instance, the appearance of U/I signals due to the breaking of the axial symmetry of the problem induced by horizontal fluctuations of the temperature, and the expected signatures of the Hanle and Zeeman effects in the presence of a magnetic field (see Benedusi et al., 2023).

The thesis work can thus be considered fully successful. The developed methods have been fundamental for the progress of the whole project and for the activity of other members of the group, who were focusing on other aspects of the problem. Indeed, in addition to the investigation presented in Riva et al. (2023), the results of this project have been fundamental for the works published in Janett et al. (2021a), Benedusi et al. (2022), Benedusi et al. (2023), and Guerreiro et al. (2023) (submitted).

Due to time reasons, some of the approaches that have been identified could not be investigated in depth (e.g., Lebedev's angular quadrature, and R^{II} calculations in large batches). These strategies would definitely deserve further investigation in the future. The approach developed in this thesis for calculating the emissivity in a two-level atom will also be a perfect starting point for a generalization to the more complex case of two-term atomic models.

Appendix A

Analytical expressions of the RT coefficients

This appendix is taken from the paper:
Assessment of the CRD approximation for the observer's frame R^{III} redistribution matrix (Riva et al., 2023).

A.1 Atomic model and data

In this work, we consider an atomic system composed of two levels (two-level atom). Each level is characterized by the energy E, the quantum number for the total angular momentum J (positive integer or semi-integer values only), and the Landé factor g. Hereafter, the physical quantities referring to the upper and lower levels will be labeled with subscripts u and ℓ, respectively. Each level is composed of $2J+1$ magnetic sublevels characterized by the magnetic quantum number M ($M=-J,-J+1,\ldots,J$). The magnetic sublevels are degenerate in the absence of magnetic fields, while they split in energy when a magnetic field **B** is present (Zeeman effect). Their energies are given by $E(M)=E_J+h\,\nu_L\,g\,M$, where E_J is the energy of the considered J-level, h is the Planck constant, and $\nu_L = e\,||\mathbf{B}||\,(4\pi m_e c)^{-1}$ (with e the elementary charge, m_e the electron mass, and c the speed of light) is the Larmor frequency[1]. The line-center frequency is $\nu_0=(E_u-E_\ell)/h$. The frequency of the Zeeman component corresponding to the transition between the upper magnetic sublevel M_u and the lower magnetic

[1] In the cgs system, the Larmor frequency is numerically approximated by $\nu_L = 1.3996\times10^6\,B$, with B expressed in G and ν_L in s^{-1}.

sublevel M_ℓ is

$$\nu_{M_u M_\ell} = \frac{E(M_u) - E(M_\ell)}{h} = \nu_0 + \nu_L (g_u M_u - g_\ell M_\ell) \,. \tag{A.1}$$

The energies, angular momenta, Landé factors, and Einstein coefficients for spontaneous emission $A_{u\ell}$ for the two-level atomic models considered in this work to synthesize the intensity and polarization of the Ca I 4227 Å, Sr II 4078 Å, and Sr I 4607 Å lines are summarized in Table A.1. The Landé factors of levels with $J_\ell = 0$ are formally taken to be zero.

Line	Air WL [Å]	E_u [cm^{-1}]	J_u	g_u	E_ℓ [cm^{-1}]	J_ℓ	g_ℓ	$A_{u\ell}$ [s^{-1}]
Ca I	4226.73	23652.304	1	1	0	0	0	2.18×10^8
Sr I	4607.33	21698.452	1	1	0	0	0	2.01×10^8
Sr II	4077.71	24516.650	3/2	4/3	0	1/2	2	1.41×10^8
Mg II *k*	2795.53	35760.880	3/2	4/3	0	1/2	2	2.60×10^8

Table A.1. *Spectral lines and corresponding atomic data.*

A.2 The redistribution matrix

All the results presented in this work are obtained considering the redistribution matrix for a two-level atom with unpolarized and infinitely-sharp lower level, in the presence of magnetic fields, as derived by Bommier (1997b). This redistribution matrix is given by the sum of two terms, commonly referred to as R^{II} and R^{III} (e.g., Hummer, 1962). In this appendix, we provide their analytic expressions, written with a slightly different notation than the one used in the original paper or in subsequent works in which the same redistribution matrices were reported and applied (e.g., Alsina Ballester et al., 2017). We first present their expressions in the atomic reference frame, taking the quantization axis along the magnetic field (Sect. A.2.1). We briefly comment on the branching ratios (Sect. A.2.1 and we derive their expressions in a new reference system with the quantization axis along the vertical (Sect. A.2.1). Subsequently, we transform them in the observer's reference frame, including bulk velocities (Sect. A.2.2). Finally, we provide their expressions under a series of simplifying approximations, including the one for R^{III} analyzed in this work (Sect. A.2.3).

A.2.1 Expression in the atomic reference frame

In the atomic reference frame, taking the quantization axis parallel to the magnetic field (magnetic reference system), the $R^{\rm II}$ and $R^{\rm III}$ redistribution matrices are respectively given by Eqs. (51) and (49) in Bommier (1997b). We refer to these redistribution matrices with the symbol $\hat{R}^{\rm X}$ (with X=II,III) to distinguish them from those expressed taking the quantization axis along a different direction. After some variable and index renaming,[2] we rewrite them in the following equivalent form

$$\hat{R}^{\rm X}_{ij}(\mathbf{r},\mathbf{\Omega},\mathbf{\Omega}',\xi,\xi') = \sum_{K,K'=0}^{2} \sum_{Q=-K_{\rm min}}^{K_{\rm min}} \mathcal{R}^{{\rm X},KK'}_{Q}(\mathbf{r},\xi,\xi')(-1)^Q\, \hat{\mathcal{T}}^{K'}_{Q,i}(\mathbf{r},\mathbf{\Omega})\, \hat{\mathcal{T}}^{K}_{-Q,j}(\mathbf{r},\mathbf{\Omega}')\,, \tag{A.2}$$

where $\mathbf{r}$ is the spatial point, $\mathbf{\Omega}$ the propagation direction, and ξ the radiation frequency in the atomic reference frame. The convention that primed and unprimed quantities refer to the incident and scattered radiation, respectively, is used. The indices i and j can take values 1, 2, 3, and 4, while $K_{\rm min} = \min(K,K')$. The quantity $\hat{\mathcal{T}}^{K}_{Q,i}$ (with $K = 0,...,2$ and $Q = -K,...,K$) is the geometrical tensor defined in Sect. 5.11 of LL04, evaluated in the magnetic reference system. Given that the direction of the magnetic field may vary with the spatial point, this tensor depends in general on $\mathbf{r}$.

The function $\mathcal{R}^{{\rm III},KK'}_{Q}$ is given by (see Eq. (49) of Bommier, 1997b)

$$\mathcal{R}^{{\rm III},KK'}_{Q}(\mathbf{r},\xi,\xi') = \sum_{K''=|Q|}^{2J_u} \left(\beta^{K''}_{Q}(\mathbf{r}) - \alpha_Q(\mathbf{r})\right) \Phi^{K''K'}_{Q}(\mathbf{r},\xi)\, \Phi^{K''K}_{Q}\left(\mathbf{r},\xi'\right)\,, \tag{A.3}$$

where

$$\beta^{K}_{Q}(\mathbf{r}) = \frac{\Gamma_R}{\Gamma_R + \Gamma_I(\mathbf{r}) + D^{(K)}(\mathbf{r}) + 2\pi {\rm i}\,\nu_L(\mathbf{r}) g_u Q}\,, \tag{A.4}$$

and

$$\alpha_Q(\mathbf{r}) = \frac{\Gamma_R}{\Gamma_R + \Gamma_I(\mathbf{r}) + \Gamma_E(\mathbf{r}) + 2\pi {\rm i}\,\nu_L(\mathbf{r}) g_u Q}\,. \tag{A.5}$$

The quantities Γ_R, Γ_I, and Γ_E are the line broadening constants due to radiative decays, inelastic collisions, and elastic collisions, respectively:

$$\Gamma_R = A_{u\ell}\,, \quad \Gamma_I = C_{u\ell}\,, \quad \Gamma_E = Q_{\rm el}\,, \tag{A.6}$$

[2] Variable and index renaming for Eqs. (49) and (51) in Bommier (1997b):

$$\mathbf{\Omega}_1 \to \mathbf{\Omega}'\,, \quad \nu_1 \to \xi'\,, \quad \nu \to \xi\,, \quad K \to K''\,, \quad K' \to K\,, \quad K'' \to K'\,,$$
$$J' \to J_u\,, \quad J \to J_\ell\,, \quad M \to M_u\,, \quad N \to M_\ell\,, \quad M' \to M'_u\,, \quad N'' \to M'_\ell\,.$$

where $C_{u\ell}$ is the rate of inelastic collisions inducing transitions from the upper to the lower level and $Q_{\rm el}$ is the rate of elastic collisions with neutral perturbers (mainly hydrogen and helium atoms). The quantity $D^{(K)}$ is the depolarizing rate due to elastic collisions. In the absence of experimental data or detailed theoretical calculations for the rates $D^{(K)}$, the approximate relation $D^{(2)} = 0.5\,Q_{\rm el}$ is generally used. Under the assumption that the interaction between the atom and the perturber is described by a single tensor operator of rank K' , the rates with $K \neq 2$ can be obtained from $D^{(2)}$ as discussed in Sect. 7.13 of LL04 (see equation after (7.108) and Eq. (7.109), which relates $D^{(1)}$ and $D^{(2)}$ for the case of $K' = 2$) . The generalized profile $\Phi_Q^{KK'}$ is given by Eq. (10.40) of LL04 and can be equivalently written as

$$\Phi_Q^{KK'}(\mathbf{r},\xi) = \sum_{M_u,M_u'=-J_u}^{J_u} \sum_{M_\ell=-J_\ell}^{J_\ell} \sum_{q,q'=-1}^{1} \mathcal{B}_{KK'QM_uM_u'M_\ell qq'} \frac{1}{2}\left[\Phi_{M_uM_\ell}(\mathbf{r},\xi) + \overline{\Phi}_{M_u'M_\ell}(\mathbf{r},\xi)\right], \tag{A.7}$$

where $K = 0, ..., 2J_u$, $K' = 0, 1, 2$, and $Q = -K_{\rm min}, ..., K_{\rm min}$ with $K_{\rm min} = \min(K, K')$. The notation $\overline{f}(\cdot)$ refers to the complex conjugate function. The quantity $\mathcal{B}_{KK'QM_uM_u'M_\ell qq'}$ is given by

$$\begin{aligned}\mathcal{B}_{KK'QM_uM_u'M_\ell qq'} &= (-1)^{1+J_u-M_u+q'}\sqrt{3(2J_u+1)(2K+1)(2K'+1)}\\ &\times \begin{pmatrix} J_u & J_\ell & 1 \\ -M_u & M_\ell & -q \end{pmatrix}\begin{pmatrix} J_u & J_\ell & 1 \\ -M_u' & M_\ell & -q' \end{pmatrix}\\ &\times \begin{pmatrix} J_u & J_u & K \\ M_u' & -M_u & -Q \end{pmatrix}\begin{pmatrix} 1 & 1 & K' \\ q & -q' & -Q \end{pmatrix},\end{aligned} \tag{A.8}$$

where the quantities in parentheses are the so-called $3j$ symbols (e.g., Sect. 2.2 of LL04). It must be observed that the $3j$ symbols are non-zero only if the sum of the arguments of the lower raw is zero. Because of this, the values of q and q' are uniquely determined once M_u, M_u', and M_ℓ are assigned. Thus, the sums over these indices in Eq. (A.7) are in practice redundant. The complex profile $\Phi_{M_uM_\ell}$ is defined as

$$\Phi_{M_uM_\ell}(\mathbf{r},\xi) = \frac{1}{\pi}\frac{1}{\Gamma(\mathbf{r}) - \mathrm{i}(\nu_{M_uM_\ell}(\mathbf{r}) - \xi)}, \tag{A.9}$$

with $\Gamma = (\Gamma_R + \Gamma_I + \Gamma_E)/4\pi$.

The function $\mathcal{R}_Q^{\mathrm{II},KK'}$ is given by (see Eq. (51) of Bommier 1997b)

$$\mathcal{R}_Q^{\mathrm{II},KK'}(\mathbf{r},\xi,\xi') = \alpha_Q(\mathbf{r}) \sum_{M_u,M_u'=-J_u}^{J_u} \sum_{M_\ell,M_\ell'=-J_\ell}^{J_\ell} \sum_{p,p',p'',p'''=-1}^{1} \mathcal{C}_{KK'QM_uM_u'M_\ell M_\ell' pp'p''p'''} \times \delta(\xi-\xi'-\nu_{M_\ell M_\ell'}(\mathbf{r}))\,\frac{1}{2}\left[\Phi_{M_u'M_\ell}(\mathbf{r},\xi') + \overline{\Phi}_{M_uM_\ell}(\mathbf{r},\xi')\right], \tag{A.10}$$

where α_Q is given by Eq. (A.5), $\delta(\cdot)$ is the Dirac delta, and $\Phi_{M_uM_\ell}$ is the complex profile defined in Eq. (A.9). The quantity $\nu_{M_\ell M_\ell'}$ is given by

$$\nu_{M_\ell M_\ell'}(\mathbf{r}) = \nu_L(\mathbf{r})(g_\ell M_\ell - g_\ell M_\ell'), \tag{A.11}$$

and $\mathcal{C}_{KK'QM_uM_u'M_\ell M_\ell' pp'p''p'''}$ is defined as (see Eq. (12) of Bommier, 1997b)

$$\begin{aligned}\mathcal{C}_{KK'QM_uM_u'M_\ell M_\ell'\; pp'p''p'''} = {} & 3\,(2J_u+1)\,\sqrt{2K+1}\,\sqrt{2K'+1}\,(-1)^{2J_u-M_\ell-M_\ell'} \\ & \times \begin{pmatrix} J_u & J_\ell & 1 \\ M_u & -M_\ell & -p \end{pmatrix} \begin{pmatrix} J_u & J_\ell & 1 \\ M_u' & -M_\ell & -p' \end{pmatrix} \\ & \times \begin{pmatrix} J_u & J_\ell & 1 \\ M_u & -M_\ell' & -p'' \end{pmatrix} \begin{pmatrix} J_u & J_\ell & 1 \\ M_u' & -M_\ell' & -p''' \end{pmatrix} \\ & \times \begin{pmatrix} 1 & 1 & K \\ -p & p' & Q \end{pmatrix} \begin{pmatrix} 1 & 1 & K' \\ -p'' & p''' & Q \end{pmatrix}. \end{aligned} \tag{A.12}$$

As for q and q' in Eq. (A.7), the sums over p, p', p'', and p''' in Eq. (A.10) are in practice redundant.

Branching ratios

The quantities $(\beta_Q^K - \alpha_Q)$ and α_Q appearing in Eqs. (A.3) and (A.10), respectively, are the branching ratios between R^{II} and R^{III}. These terms also contain the branching ratio for the scattering contribution to the total emissivity, $1-\epsilon$, with

$$\epsilon(\mathbf{r}) = \frac{\Gamma_I(\mathbf{r})}{\Gamma_R + \Gamma_I(\mathbf{r})} \tag{A.13}$$

the photon destruction probability. Factorizing this term (i.e., writing $\alpha_Q = (1-\epsilon)\,\tilde{\alpha}_Q$ and $\beta_Q^K = (1-\epsilon)\,\tilde{\beta}_Q^K$), the net branching ratios for R^{II} and R^{III} are therefore

$$\tilde{\alpha}_Q(\mathbf{r}) = \frac{\Gamma_R + \Gamma_I(\mathbf{r})}{\Gamma_R + \Gamma_I(\mathbf{r}) + \Gamma_E(\mathbf{r}) + 2\pi\mathrm{i}\,\nu_L(\mathbf{r})g_uQ}, \tag{A.14}$$

$$\begin{aligned}\tilde{\beta}_Q^K(\mathbf{r}) - \tilde{\alpha}_Q(\mathbf{r}) = {} & \frac{\Gamma_R + \Gamma_I(\mathbf{r})}{\Gamma_R + \Gamma_I(\mathbf{r}) + D^{(K)}(\mathbf{r}) + 2\pi\mathrm{i}\,\nu_L(\mathbf{r})g_uQ} - \\ & \frac{\Gamma_R + \Gamma_I(\mathbf{r})}{\Gamma_R + \Gamma_I(\mathbf{r}) + \Gamma_E(\mathbf{r}) + 2\pi\mathrm{i}\,\nu_L(\mathbf{r})g_uQ}. \end{aligned} \tag{A.15}$$

It can be observed that the branching ratios for the multipolar component $K = Q = 0$ coincides with those for the unpolarized case. If there are no elastic collisions ($\Gamma_E = D^{(K)} = 0$), the branching ratio for $R^{\rm III}$ vanishes and that for $R^{\rm II}$ goes to unity (limit of coherent scattering in the atomic frame). If elastic collisions are very efficient ($\Gamma_E \gg \Gamma_R, \Gamma_I$), the branching ratio for $R^{\rm II}$ becomes negligible with respect to that for $R^{\rm III}$. However, in this case also $D^{(K)}$ takes very large values and atomic polarization also becomes negligible.

Rotation of the quantization axis

The expressions of $R^{\rm II}$ and $R^{\rm III}$ provided above can be transformed into a new reference system with the quantization axis directed along any arbitrary direction by rotating the tensors $\hat{\mathcal{T}}^K_{Q,i}$ in Eq. (A.2). In this work, the problem is formulated considering a Cartesian reference system with the z-axis (quantization axis) directed along the vertical (vertical reference system) and the x-axis directed so that the LOS towards the observer lies in the $x-z$ plane **see Fig. 1.4)**. The relation between the geometrical tensors in the magnetic reference system ($\hat{\mathcal{T}}^K_{Q,i}$) and in the vertical reference system ($\mathcal{T}^K_{Q,i}$) is

$$\hat{\mathcal{T}}^K_{Q,i}(\mathbf{r},\mathbf{\Omega}) = \sum_{Q'=-K}^{K} \mathcal{T}^K_{Q',i}(\mathbf{\Omega})\, \overline{\mathcal{D}}^K_{QQ'}(R_B(\mathbf{r}))\,, \tag{A.16}$$

where $\mathcal{D}^K_{QQ'}$ is the *rotation matrix* (e.g., Sect. 2.6 of LL04), and R_B is the rotation that brings the magnetic reference system onto the vertical one. A bar over a quantity indicates the complex conjugate. It must be noticed that the tensor $\mathcal{T}^K_{Q,i}$ defined in the vertical reference system only depends on the propagation direction of the incident (or scattered) radiation, and does not depend on the spatial point $\mathbf{r}$. The rotation R_B is specified by the Euler angles $R_B(\mathbf{r}) = (0, -\theta_B(\mathbf{r}), -\chi_B(\mathbf{r}))$, where θ_B and χ_B are the inclination and azimuth, respectively, of the magnetic field in the vertical reference system. In the vertical reference system, the redistribution matrices are thus given by

$$R^{\rm X}_{ij}(\mathbf{r},\mathbf{\Omega},\mathbf{\Omega}',\xi,\xi') = \sum_{K,K'=0}^{2} \sum_{Q=-K_{\rm min}}^{K_{\rm min}} \mathcal{R}^{{\rm X},KK'}_{Q}(\mathbf{r},\xi,\xi')\, \mathcal{P}^{KK'}_{Q,ij}(\mathbf{r},\mathbf{\Omega},\mathbf{\Omega}')\,, \tag{A.17}$$

with

$$\mathcal{P}^{KK'}_{Q,ij}(\mathbf{r},\mathbf{\Omega},\mathbf{\Omega}') = \sum_{Q'=-K}^{K} \sum_{Q''=-K'}^{K'} (-1)^{Q'}\, \mathcal{T}^{K'}_{Q'',i}(\mathbf{\Omega})\, \mathcal{T}^{K}_{-Q',j}(\mathbf{\Omega}')\, \overline{\mathcal{D}}^{K'}_{QQ''}(\mathbf{r})\, \mathcal{D}^{K}_{QQ'}(\mathbf{r})\,. \tag{A.18}$$

For notational simplicity, we have only included the spatial point dependency of the rotation matrices, leaving implicit that the rotation R_B is always considered. Furthermore, we have used the relation

$$\overline{\mathcal{D}}^{K}_{QQ'}(R) = (-1)^{Q-Q'}\mathcal{D}^{K}_{-Q-Q'}(R)\,.$$

A.2.2 Expression in the observer's reference frame

We now present the expressions of R^{II} and R^{III} in the observer's frame, where it is assumed that the atom is moving with velocity $\mathbf{v}$. Considering the Doppler effect, the frequencies measured in the atomic frame, ξ' and ξ, and those measured in the observer's frame, ν' and ν, are related by:

$$\xi' = \nu' - \frac{\nu_0}{c}\mathbf{v}\cdot\mathbf{\Omega}'\,, \tag{A.19}$$

$$\xi = \nu - \frac{\nu_0}{c}\mathbf{v}\cdot\mathbf{\Omega}\,, \tag{A.20}$$

where c is the speed of light. The velocity $\mathbf{v}$ is generally given by the sum of two terms, namely,

$$\mathbf{v}(\mathbf{r}) = \mathbf{v}_{\text{th}}(\mathbf{r}) + \mathbf{v}_{\text{b}}(\mathbf{r})\,, \tag{A.21}$$

where $\mathbf{v}_{\text{th}}$ is the thermal component and $\mathbf{v}_{\text{b}}$ is the bulk component. The thermal component is generally well described by a Maxwellian distribution

$$\mathcal{P}(\mathbf{v}_{\text{th}}(\mathbf{r})) = \left(\frac{m}{2\pi k_B T(\mathbf{r})}\right)^{3/2} \exp\left(-\frac{m v_{\text{th}}(\mathbf{r})^2}{2k_B T(\mathbf{r})}\right)\,, \tag{A.22}$$

where k_B is the Boltzmann constant, T the temperature, and m the mass of the considered atom or ion. Let $\check{\mathcal{R}}^{\text{x},KK'}_{Q}$ be the frequency-dependent part of the redistribution matrix (see Eq. (A.10) for R^{II} and Eq. (A.3) for R^{III}), expressed in terms of the frequencies ν and ν' through Eqs. (A.20) and (A.19), for an atom moving with velocity $\mathbf{v}$. The expression of $\mathcal{R}^{\text{x},KK'}_{Q}$ in the observer's frame is obtained by locally averaging $\check{\mathcal{R}}^{\text{x},KK'}_{Q}$ over the distribution of thermal velocities:

$$\mathcal{R}^{\text{x},KK'}_{Q}(\mathbf{r},\mathbf{\Omega},\mathbf{\Omega}',\nu,\nu') = \int \mathrm{d}^3\mathbf{v}_{\text{th}}(\mathbf{r})\,\mathcal{P}(\mathbf{v}_{\text{th}}(\mathbf{r}))\,\check{\mathcal{R}}^{\text{x},KK'}_{Q}(\mathbf{r},\mathbf{\Omega},\mathbf{\Omega}',\nu,\nu')\,. \tag{A.23}$$

This average is performed following the same approach as in the unpolarized case (e.g., Hummer, 1962; Mihalas, 1978). We provide the final expressions, which are better formulated by defining the Doppler width

$$\Delta\nu_D(\mathbf{r}) = \frac{\nu_0}{c}\sqrt{\frac{2k_B T(\mathbf{r})}{m}}\,, \tag{A.24}$$

the damping constant $a(\mathbf{r}) = \Gamma(\mathbf{r})/\Delta\nu_D(\mathbf{r})$, and the reduced frequency

$$u(\mathbf{r},\nu) = \frac{(\nu_0 - \nu)}{\Delta\nu_D(\mathbf{r})}\,. \tag{A.25}$$

Moreover, we introduce the reduced magnetic and bulk velocity shifts, respectively defined as

$$u_{M_u M_\ell}(\mathbf{r}) = \frac{\nu_L(\mathbf{r})(g_u M_u - g_\ell M_\ell)}{\Delta\nu_D(\mathbf{r})}\,, \text{ and } \; u_b(\mathbf{r},\boldsymbol{\Omega}) = \frac{\nu_0}{c}\,\frac{\mathbf{v}_b(\mathbf{r})\cdot\boldsymbol{\Omega}}{\Delta\nu_D(\mathbf{r})}\,. \tag{A.26}$$

In order to make the notation lighter, in the following equations, we will also use the variable

$$\tilde{u}_{M_u M_\ell}(\mathbf{r},\boldsymbol{\Omega},\nu) = u(\mathbf{r},\nu) + u_{M_u M_\ell}(\mathbf{r}) + u_b(\mathbf{r},\boldsymbol{\Omega})\,. \tag{A.27}$$

R^{II} redistribution matrix

The R^{II} redistribution matrix in the observer's frame, taking the quantization axis along the vertical, is given by

$$R^{\text{II}}_{ij}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu') = \sum_{K,K'=0}^{2}\,\sum_{Q=-K_{\min}}^{K_{\min}} \mathcal{R}^{\text{II},KK'}_{Q}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu')\,\mathcal{P}^{KK'}_{Q,ij}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}')\,, \tag{A.28}$$

where $\mathcal{P}^{KK'}_{Q,ij}$ is given by Eq. (A.18) and $\mathcal{R}^{\text{II},KK'}_{Q}$ takes different analytic expressions depending on relative orientation of $\boldsymbol{\Omega}$ and $\boldsymbol{\Omega}'$.[3]

- If $\boldsymbol{\Omega}' \neq \boldsymbol{\Omega}, -\boldsymbol{\Omega}$:

$$\begin{aligned}\mathcal{R}^{\text{II},KK'}_{Q}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu') = {} & \frac{1}{\Delta\nu_D(\mathbf{r})^2}\,\alpha_Q(\mathbf{r}) \sum_{M_u,M_u'}\,\sum_{M_\ell,M_\ell'}\,\sum_{p,p',p'',p'''} \mathcal{C}_{KK'QM_uM_u'M_\ell M_\ell' pp'p''p'''} \\ & \times \frac{1}{2\pi\sin\Theta}\exp\left[-\left(\frac{\tilde{u}_{M_u M_\ell'}(\mathbf{r},\boldsymbol{\Omega},\nu) - \tilde{u}_{M_u M_\ell}(\mathbf{r},\boldsymbol{\Omega}',\nu')}{2\sin(\Theta/2)}\right)^2\right] \\ & \times \left[W\left(\frac{a(\mathbf{r})}{\cos(\Theta/2)},\frac{\tilde{u}_{M_u' M_\ell'}(\mathbf{r},\boldsymbol{\Omega},\nu) + \tilde{u}_{M_u' M_\ell}(\mathbf{r},\boldsymbol{\Omega}',\nu')}{2\cos(\Theta/2)}\right)\right. \\ & \left. + \overline{W}\left(\frac{a(\mathbf{r})}{\cos(\Theta/2)},\frac{\tilde{u}_{M_u M_\ell'}(\mathbf{r},\boldsymbol{\Omega},\nu) + \tilde{u}_{M_u M_\ell}(\mathbf{r},\boldsymbol{\Omega}',\nu')}{2\cos(\Theta/2)}\right)\right],\end{aligned} \tag{A.29}$$

[3] For notational simplicity, the ranges of the various sums are not explicitly indicated.

where Θ is the angle between the directions $\mathbf{\Omega}$ and $\mathbf{\Omega}'$ (scattering angle)

$$\cos\Theta = \mathbf{\Omega}\cdot\mathbf{\Omega}' = \cos\theta\,\cos\theta' + \sin\theta\,\sin\theta'\,\cos(\chi - \chi')\,. \tag{A.30}$$

In the previous equation, θ and χ are respectively the inclination and azimuth of the direction $\mathbf{\Omega}$, and θ' and χ' are the same angles for the direction $\mathbf{\Omega}'$. The Faddeeva function $W(y,x)$ is defined as (e.g., Sect. 5.4 of LL04):

$$W(y,x) = H(y,x) + \mathrm{i}L(y,x) = \mathrm{e}^{-z^2}\,\mathrm{erfc}(-\mathrm{i}z)\,, \tag{A.31}$$

where H and L are the Voigt and associated dispersion profiles, respectively, erfc is the complementary error function, and $z = x + \mathrm{i}y$.

- If $\mathbf{\Omega}' = \mathbf{\Omega}$ (forward scattering):

$$\begin{aligned}\mathcal{R}_Q^{\mathrm{II},KK'}(\mathbf{r},\mathbf{\Omega},\mathbf{\Omega},\nu,\nu') &= \frac{1}{\Delta\nu_D(\mathbf{r})^2}\,\alpha_Q(\mathbf{r}) \\ &\times \sum_{M_u,M_u'}\sum_{M_\ell,M_\ell'}\sum_{p,p',p'',p'''} \mathcal{C}_{KK'QM_uM_u'M_\ell M_\ell' pp'p''p'''} \\ &\times \frac{1}{2\pi^{1/2}}\left[W(a(\mathbf{r}),\tilde{u}_{M_u'M_\ell}(\mathbf{r},\mathbf{\Omega},\nu')) + \overline{W}(a(\mathbf{r}),\tilde{u}_{M_uM_\ell}(\mathbf{r},\mathbf{\Omega},\nu'))\right] \\ &\times \delta\left(\tilde{u}_{M_uM_\ell}(\mathbf{r},\mathbf{\Omega},\nu') - \tilde{u}_{M_uM_\ell'}(\mathbf{r},\mathbf{\Omega},\nu)\right)\,. \end{aligned} \tag{A.32}$$

- If $\mathbf{\Omega}' = -\mathbf{\Omega}$ (backward scattering):

$$\begin{aligned}\mathcal{R}_Q^{\mathrm{II},KK'}(\mathbf{r},\mathbf{\Omega},-\mathbf{\Omega},\nu,\nu') &= \frac{1}{\Delta\nu_D(\mathbf{r})^2}\,\alpha_Q(\mathbf{r}) \\ &\times \sum_{M_u,M_u'}\sum_{M_\ell,M_\ell'}\sum_{p,p',p'',p'''} \mathcal{C}_{KK'QM_uM_u'M_\ell M_\ell' pp'p''p'''} \\ &\times \frac{1}{4\pi^{3/2}}\,\exp\left[-\left(\frac{\tilde{u}_{M_uM_\ell'}(\mathbf{r},\mathbf{\Omega},\nu) - \tilde{u}_{M_uM_\ell}(\mathbf{r},-\mathbf{\Omega},\nu')}{2}\right)^2\right] \\ &\times \left[\varphi\left(a(\mathbf{r}),\frac{\tilde{u}_{M_u'M_\ell'}(\mathbf{r},\mathbf{\Omega},\nu) + \tilde{u}_{M_u'M_\ell}(\mathbf{r},-\mathbf{\Omega},\nu')}{2}\right)\right. \\ &\quad\left. + \overline{\varphi}\left(a(\mathbf{r}),\frac{\tilde{u}_{M_uM_\ell'}(\mathbf{r},\mathbf{\Omega},\nu) + \tilde{u}_{M_uM_\ell}(\mathbf{r},-\mathbf{\Omega},\nu')}{2}\right)\right], \end{aligned} \tag{A.33}$$

where $\varphi(y,x)$ is defined as

$$\varphi(y,x) = \frac{1}{y - \mathrm{i}x}\,. \tag{A.34}$$

R^{III} redistribution matrix

The R^{III} redistribution matrix in the observer's frame, taking the quantization axis along the vertical, is given by

$$R^{\mathrm{III}}_{ij}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu') = \sum_{K,K'=0}^{2} \sum_{Q=-K_{\min}}^{K_{\min}} \mathcal{R}^{\mathrm{III},KK'}_{Q}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu')\,\mathcal{P}^{KK'}_{Q,ij}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}')\,, \tag{A.35}$$

where $\mathcal{P}^{KK'}_{Q,ij}$ is given by Eq. (A.18) and $\mathcal{R}^{\mathrm{III},KK'}_{Q}$ by the following expression

$$\begin{aligned}\mathcal{R}^{\mathrm{III},KK'}_{Q}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu') = {} & \frac{1}{\Delta\nu_D^2(\mathbf{r})} \sum_{K''=|Q|}^{2J_u} \left(\beta^{K''}_{Q}(\mathbf{r}) - \alpha_Q(\mathbf{r})\right) \\ & \times \sum_{M_u,M_u'} \sum_{M_\ell} \sum_{q,q'} \mathcal{B}_{K''K'QM_uM_u'M_\ell qq'} \sum_{M_u'',M_u'''} \sum_{M_\ell'} \sum_{q'',q'''} \mathcal{B}_{K''KQM_u''M_u'''M_\ell'q''q'''} \\ & \times \frac{1}{4}\Big[\mathcal{J}_{(M_uM_\ell),(M_u''M_\ell')}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu') + \mathcal{J}_{(M_uM_\ell),\overline{(M_u'''M_\ell')}}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu') \\ & \quad + \mathcal{J}_{\overline{(M_u'M_\ell)},(M_u''M_\ell')}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu') + \mathcal{J}_{\overline{(M_u'M_\ell)},\overline{(M_u'''M_\ell')}}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu')\Big]\,.\end{aligned} \tag{A.36}$$

The quantity $\mathcal{J}_{(M_uM_\ell),(M_u'M_\ell')}$ is given by the integral of the product of three functions: an exponential function, which only depends on the integration variable y; a function depending on y, the frequency and direction of the scattered radiation, and the first pair of magnetic quantum numbers in the subscript; and a function depending on y, the frequency and direction of the incident radiation, and the second pair of magnetic quantum numbers in the subscript. A bar over a pair of magnetic quantum numbers means that the complex conjugate of the corresponding function has to be considered. The explicit expression of $\mathcal{J}_{(M_uM_\ell),(M_u'M_\ell')}$ is provided below.

- If $\boldsymbol{\Omega}' \neq \boldsymbol{\Omega}, -\boldsymbol{\Omega}$:

$$\begin{aligned}\mathcal{J}_{(M_uM_\ell),(M_u'M_\ell')}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu') = {} & \frac{1}{\pi^2\sin\Theta} \\ & \times \int \mathrm{d}y\, \exp\left(-y^2\right) \\ & \quad W\left(\frac{a(\mathbf{r})}{\sin\Theta}, \frac{\tilde{u}_{M_uM_\ell}(\mathbf{r},\boldsymbol{\Omega},\nu) + y\cos\Theta}{\sin\Theta}\right) \\ & \quad \varphi\left(a(\mathbf{r}), \tilde{u}_{M_u'M_\ell'}(\mathbf{r},\boldsymbol{\Omega}'\nu') + y\right)\,.\end{aligned} \tag{A.37}$$

- If $\mathbf{\Omega}' = \mathbf{\Omega}$ (forward scattering):

$$\begin{aligned}\mathcal{J}_{(M_u M_\ell),(M_u' M_\ell')}(\mathbf{r}, \mathbf{\Omega}, \mathbf{\Omega}, \nu, \nu') = &\frac{1}{\pi^{5/2}} \\ \times \int &\mathrm{d}y \exp\left(-y^2\right) \\ &\varphi\left(a(\mathbf{r}), \tilde{u}_{M_u M_\ell}(\mathbf{r}, \mathbf{\Omega}, \nu) + y\right) \\ &\varphi\left(a(\mathbf{r}), \tilde{u}_{M_u' M_\ell'}(\mathbf{r}, \mathbf{\Omega}, \nu') + y\right).\end{aligned} \tag{A.38}$$

- If $\mathbf{\Omega}' = -\mathbf{\Omega}$ (backward scattering):

$$\begin{aligned}\mathcal{J}_{(M_u M_\ell),(M_u' M_\ell')}(\mathbf{r}, \mathbf{\Omega}, -\mathbf{\Omega}, \nu, \nu') = &\frac{1}{\pi^{5/2}} \\ \times \int &\mathrm{d}y \exp\left(-y^2\right) \\ &\varphi\left(a(\mathbf{r}), \tilde{u}_{M_u M_\ell}(\mathbf{r}, \mathbf{\Omega}, \nu) - y\right) \\ &\varphi\left(a(\mathbf{r}), \tilde{u}_{M_u' M_\ell'}(\mathbf{r}, -\mathbf{\Omega}, \nu') + y\right).\end{aligned} \tag{A.39}$$

The function $\varphi(y, x)$ is that defined in Eq. (A.34).

A.2.3 Approximate expressions

The expressions of the redistribution matrices in the observer's frame derived in the previous section show the complex coupling between frequencies and angles introduced by the Doppler effect. This coupling makes the evaluation of the scattering integral ((1.4)) extremely demanding from a computational point of view. To reduce the computational cost of the problem, approximate expressions in which the frequency and angular dependencies are partially or totally decoupled are often used.

In the absence of bulk velocities, or when working in a reference frame in which the bulk velocity is zero (comoving frame), the arguments of the exponential and of the Faddeeva functions do not depend on $\mathbf{\Omega}$ and $\mathbf{\Omega}'$, and the angular dependence of the $\mathcal{R}_Q^{\text{II},KK'}$ function (A.29) is reduced to the scattering angle Θ. In this case, a commonly used approximation is the so-called *angle-averaged* (AA) one, which consists in averaging this function with respect to Θ (e.g., Rees and Saliba, 1982):

$$\mathcal{R}_Q^{\text{II}-\text{AA},KK'}(\mathbf{r}, \nu, \nu') = \frac{1}{2} \int_0^{\pi} \mathrm{d}\Theta \, \sin\Theta \, \mathcal{R}_Q^{\text{II},KK'}(\mathbf{r}, \Theta, \nu, \nu'). \tag{A.40}$$

The function $\mathcal{R}_Q^{\mathrm{II},KK'}$ in the r.h.s. of Eq. (A.40) is given by Eq. (A.29), assuming no bulk velocities. The ensuing $R_{ij}^{\mathrm{II-AA}}$ redistribution matrix in the observer's frame is characterized by a complete decoupling of the frequencies and angles. Thus, the computational cost of the problem is significantly lowered.

The AA approximation can in principle be applied also to $\mathcal{R}_Q^{\mathrm{III},KK'}$ (e.g., Bommier, 1997b). However, an even stronger assumption is often considered for this function, namely that there is no correlation between the frequencies of the incident and scattered radiation in the observer's frame (e.g., Mihalas, 1978). Under this approximation, often referred to as the limit of *complete frequency redistribution* (CRD) in the observer's frame, we have:

$$\mathcal{R}_Q^{\mathrm{III-CRD},KK'}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu') = \frac{1}{\Delta\nu_D^2(\mathbf{r})}\sum_{K''=|Q|}^{2J_u}\Big(\beta_Q^{K''}(\mathbf{r})-\alpha_Q(\mathbf{r})\Big) \times \Phi_Q^{K''K'}(\mathbf{r},\boldsymbol{\Omega},\nu)\,\Phi_Q^{K''K}(\mathbf{r},\boldsymbol{\Omega}',\nu')\,, \tag{A.41}$$

where $\Phi_Q^{KK'}$ is the generalized profile (see Eq. (A.7)), defined in the observer's frame. This is obtained by convolving the profiles $\Phi_{M_uM_\ell}$ of Eq. (A.9) with a Gaussian function in order to account for the thermal and microturbulent velocity distribution, thus obtaining a Faddeeva function:

$$\Phi_Q^{KK'}(\mathbf{r},\boldsymbol{\Omega},\nu) = \sum_{M_u,M_u'}\sum_{M_\ell}\sum_{q,q'}\mathcal{B}_{KK'QM_uM_u'M_\ell qq'} \times \frac{1}{2\sqrt{\pi}}\Big[W(a(\mathbf{r}),\tilde{u}_{M_uM_\ell}(\mathbf{r},\boldsymbol{\Omega},\nu)) + \overline{W}(a(\mathbf{r}),\tilde{u}_{M_u'M_\ell}(\mathbf{r},\boldsymbol{\Omega},\nu))\Big]\,. \tag{A.42}$$

In the absence of bulk velocities or when working in the comoving frame, the $R^{\mathrm{III-CRD}}$ redistribution matrix is characterized by a complete decoupling between angles and frequencies, and between the frequencies of the incident and scattered radiation.

A.3 Line contribution to the propagation matrix and thermal emissivity

Neglecting stimulated emission (which is generally an excellent assumption in the solar atmosphere), the elements of the propagation matrix for a two-level atom with an unpolarized lower level, in the observer's reference frame, are given by (see App. 13 of LL04)

$$\eta_i^\ell(\mathbf{r},\boldsymbol{\Omega},\nu) = k_L(\mathbf{r}) \sum_{K=0}^{2} \Phi_0^{0K}(\mathbf{r},\boldsymbol{\Omega},\nu) \sum_{Q=-K}^{K} \mathcal{T}_{Q,i}^{K}(\boldsymbol{\Omega})\, \overline{\mathcal{D}}_{0Q}^{K}(\mathbf{r}), \qquad (i=1,2,3,4), \tag{A.43}$$

$$\rho_i^\ell(\mathbf{r},\boldsymbol{\Omega},\nu) = k_L(\mathbf{r}) \sum_{K=0}^{2} \Psi_0^{0K}(\mathbf{r},\boldsymbol{\Omega},\nu) \sum_{Q=-K}^{K} \mathcal{T}_{Q,i}^{K}(\boldsymbol{\Omega})\, \overline{\mathcal{D}}_{0Q}^{K}(\mathbf{r}), \qquad (i=2,3,4), \tag{A.44}$$

where $\mathcal{T}_{Q,i}^K$ is the geometrical tensor evaluated in the vertical reference system and $\mathcal{D}_{QQ'}^K$ are the rotation matrices (see App. A.2.1). The quantities Φ_0^{0K} are particular components of the generalized profile of Eq. (A.42), while Ψ_0^{0K} are particular components of the generalized *dispersion* profile, defined as (see App. 13 of LL04)

$$\begin{aligned} \mathrm{i}\,\Psi_Q^{KK'}(\mathbf{r},\boldsymbol{\Omega},\nu) = & \sum_{M_u,M_u'} \sum_{M_\ell} \sum_{q,q'} \mathcal{B}_{KK'QM_uM_u'M_\ell qq'} \\ & \times \frac{1}{2\sqrt{\pi}} \left[W(a(\mathbf{r}),\tilde{u}_{M_uM_\ell}(\mathbf{r},\boldsymbol{\Omega},\nu)) - \overline{W}(a(\mathbf{r}),\tilde{u}_{M_u'M_\ell}(\mathbf{r},\boldsymbol{\Omega},\nu)) \right]. \end{aligned} \tag{A.45}$$

The explicit expression of the frequency-integrated line absorption coefficient k_L is

$$k_L(\mathbf{r}) = \frac{h\nu_0}{4\pi} B_{\ell u}\, \mathcal{N}_\ell(\mathbf{r}) = \frac{c^2}{8\pi\nu_0^2} \frac{2J_u+1}{2J_\ell+1} A_{u\ell}\, \mathcal{N}_\ell(\mathbf{r}), \tag{A.46}$$

where $A_{u\ell}$ and $B_{\ell u}$ are the Einstein coefficients for spontaneous emission and absorption, respectively, and $\mathcal{N}_\ell$ is the population of the lower level (see also Table A.1).

Under the assumption of isotropic inelastic collisions, the line thermal contribution to the emissivity is given by (e.g., Alsina Ballester et al., 2017):

$$\varepsilon_i^{\ell,\mathrm{th}}(\mathbf{r},\boldsymbol{\Omega},\nu) = \epsilon\,(\mathbf{r})\; W\,(\nu, T(\mathbf{r}))\; \eta_i^\ell(\mathbf{r},\boldsymbol{\Omega},\nu), \qquad (i=1,2,3,4), \tag{A.47}$$

where W is the Planck function in the Wien limit (consistently with the assumption of neglecting stimulated emission), and ϵ is the photon destruction probability defined in Eq. (A.13).

A.4 Continuum contributions

In the visible range of the solar spectrum, continuum processes only contribute to the emission coefficient (with a thermal and a scattering term) and to the absorption coefficient for intensity. Labeling continuum contributions with the apex 'c', we have:

$$\varepsilon_i^c(\mathbf{r},\mathbf{\Omega},\nu) = \varepsilon_I^{c,\mathrm{th}}(\mathbf{r},\nu)\,\delta_{i0} + \varepsilon_i^{c,\mathrm{sc}}(\mathbf{r},\mathbf{\Omega},\nu) \qquad (i=1,2,3,4) \tag{A.48}$$

$$\eta_i^c(\mathbf{r},\mathbf{\Omega},\nu) = k_c(\mathbf{r},\nu)\,\delta_{i0} \qquad (i=1,2,3,4) \tag{A.49}$$

$$\rho_i^c(\mathbf{r},\mathbf{\Omega},\nu) = 0 \qquad (i=2,3,4) \tag{A.50}$$

where $\varepsilon_I^{c,\mathrm{th}}$ is the continuum thermal emissivity and k_c the continuum total opacity. Under the assumption that continuum scattering processes are coherent in the observer's frame, the scattering contribution to the continuum emission coefficient is given by

$$\varepsilon_i^{c,\mathrm{sc}}(\mathbf{r},\mathbf{\Omega},\nu) = \sigma(\mathbf{r},\nu)\sum_{K=0}^{2}\sum_{Q=-K}^{K}(-1)^Q\,\mathcal{T}_{Q,i}^K(\mathbf{\Omega})\,J_{-Q}^K(\mathbf{r},\nu)\,, \tag{A.51}$$

where σ is the continuum absorption coefficient for scattering and J_Q^K is the radiation field tensor (see Sect. 5.11 of LL04), given by

$$J_Q^K(\mathbf{r},\nu) = \oint \frac{\mathrm{d}\mathbf{\Omega}}{4\pi}\sum_{j=1}^{4}\mathcal{T}_{Q,j}^K(\mathbf{\Omega})\,I_j(\mathbf{r},\mathbf{\Omega},\nu)\,. \tag{A.52}$$

Line and continuum contributions to the emissivity and propagation matrix simply add together.

A.5 Micro-structured isotropic magnetic field

In this section, we provide the expressions of the various RT coefficients in the presence of an unimodal micro-structured magnetic field, namely a magnetic field with a given intensity and an orientation that changes over scales below the photons' mean-free path. In particular, we consider a magnetic field whose orientation is isotropically distributed over all possible directions. To describe this scenario, the RT coefficients must be averaged over this distribution of magnetic field orientations:

$$\tilde{\eta}_i^\ell(\mathbf{r},\mathbf{\Omega},\nu) = \langle \eta_i^\ell(\mathbf{r},\mathbf{\Omega},\nu)\rangle = \frac{1}{4\pi}\int_0^{2\pi}\mathrm{d}\chi_B\int_0^{\pi}\mathrm{d}\theta_B\sin\theta_B\,\eta_i^\ell(\mathbf{r},\mathbf{\Omega},\nu)\,, \tag{A.53}$$

$$\tilde{\rho}_i^\ell(\mathbf{r},\mathbf{\Omega},\nu) = \langle \rho_i^\ell(\mathbf{r},\mathbf{\Omega},\nu)\rangle = \frac{1}{4\pi}\int_0^{2\pi}\mathrm{d}\chi_B\int_0^{\pi}\mathrm{d}\theta_B\sin\theta_B\,\rho_i^\ell(\mathbf{r},\mathbf{\Omega},\nu)\,, \tag{A.54}$$

$$\tilde{\varepsilon}_i^\ell(\mathbf{r},\mathbf{\Omega},\nu) = \langle \varepsilon_i^\ell(\mathbf{r},\mathbf{\Omega},\nu)\rangle = \frac{1}{4\pi}\int_0^{2\pi}\mathrm{d}\chi_B\int_0^{\pi}\mathrm{d}\theta_B\sin\theta_B\,\varepsilon_i^\ell(\mathbf{r},\mathbf{\Omega},\nu)\,, \tag{A.55}$$

Recalling the expressions of the RT coefficients in the presence of a deterministic magnetic field, and observing that all the dependence on the orientation of the magnetic field is contained in the rotation matrices, it can be easily verified that the calculation of the averages above reduces to the evaluation of the following integrals (see Eqs. (47a) and (47b) of Alsina Ballester et al., 2017)

$$\frac{1}{4\pi}\int_0^{2\pi} \mathrm{d}\chi_B \int_0^{\pi} \mathrm{d}\theta_B \sin\theta_B \,\overline{\mathcal{D}}^K_{0Q}(\mathbf{r}) = \delta_{K0}\,\delta_{Q0}\,, \tag{A.56}$$

$$\frac{1}{4\pi}\int_0^{2\pi} \mathrm{d}\chi_B \int_0^{\pi} \mathrm{d}\theta_B \sin\theta_B \,\overline{\mathcal{D}}^{K'}_{QQ''}(\mathbf{r})\,\mathcal{D}^K_{QQ'}(\mathbf{r}) = \delta_{KK'}\,\delta_{Q'Q''}\frac{1}{2K+1}\,. \tag{A.57}$$

Considering the integrals above, it can be immediately seen that the only non-zero element of the propagation matrix is:

$$\tilde{\eta}^{\ell}_1\,(\mathbf{r},\boldsymbol{\Omega},\nu) = k_L\,(\mathbf{r})\;\Phi^{00}_0\,(\mathbf{r},\boldsymbol{\Omega},\nu)\,. \tag{A.58}$$

Similarly, the only non-zero element of the thermal term of the emissivity is:

$$\tilde{\varepsilon}^{\ell,\mathrm{th}}_1\,(\mathbf{r},\boldsymbol{\Omega},\nu) = \epsilon\,(\mathbf{r})\;W_T\,(\mathbf{r},\nu)\;\tilde{\eta}^{\ell}_1\,(\mathbf{r},\boldsymbol{\Omega},\nu)\,, \tag{A.59}$$

Finally, the scattering term of the emissivity is still given by Eq. (1.4), considering the redistribution matrices

$$\tilde{R}^{\mathrm{x}}_{ij}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu') = \sum_{K=0}^{2}\sum_{Q=-K}^{K}\mathcal{R}^{\mathrm{x},KK}_{Q}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}',\nu,\nu')\,\tilde{\mathcal{P}}^K_{ij}(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}')\,, \tag{A.60}$$

where $\mathcal{R}^{\mathrm{II},KK}_{Q}$ and $\mathcal{R}^{\mathrm{III},KK}_{Q}$ have the expressions provided in Sects. A.2.2 and A.2.2, respectively, and

$$\tilde{\mathcal{P}}^K_{ij}\left(\mathbf{r},\boldsymbol{\Omega},\boldsymbol{\Omega}'\right) = \frac{1}{2K+1}\sum_{Q'=-K}^{K}(-1)^{Q'}\mathcal{T}^K_{Q',i}\,(\boldsymbol{\Omega})\;\mathcal{T}^K_{-Q',j}\left(\boldsymbol{\Omega}'\right)\,. \tag{A.61}$$

A.6 Table of constants

Symbol	Name	Value - gcs	Value - SI
c	Speed of light	29979245800 cm/s	299792458 m/s
K_B	Boltzmann constant	$1.3807 \times 10^{-16} cm^2 g\, s^{-2} K^{-1}$	$1.380649 \times 10^{-23}\, JK^{-1}$
h	Planck's constant	$6.6261 \times 10^{-27}\, cm^2 g\, s^{-1}$	$6.6261 \times 10^{-34}\, m^2 kg/s$
e	Elementary charge	$4.8032 \times 10^{-10}\, cm^{3/2} g^{1/2} s^{-1}$	$1.6021765 \times 10^{-19}\, C$
m_e	Electron mass	$9.10938291 \times 10^{-31}\, g$	$9.10938291 \times 10^{-31}\, kg$
m	Unit atomic mass	$1.6605402 \times 10^{-24}\, g$	$1.6605402 \times 10^{-27}\, kg$
$M_\odot$	Solar mass	$1.989 \times 10^{33}\, g$	$1.989 \times 10^{30}\, kg$
$d_{\oplus\odot}$	Earth-Sun distance	$1.49597870 \times 10^{13}\, cm$	$1.49597870 \times 10^{11}\, m$
$R_\odot$	Solar radius	$6.955 \times 10^{10}\, cm$	$6.955 \times 10^{8}\, m$

Table A.2. *Table of the constants.*

Appendix B

Gaussian Quadratures

Gaussian quadratures are a family of numerical integration methods that are, in practice, the most efficient and accurate quadrature rules known. They guarantee an accuracy of the order of $2N - 1$ (i.e., the quadrature rule is exact for all polynomials $P \in \mathbb{P}_{2N-1}$), which is the maximum known accuracy with respect to the number of nodes. The original formulation of Gaussian quadratures based on orthogonal polynomials was developed by Carl Gustav Jacobi (1804-1851).

The other relevant property of Gaussian quadratures is that all quadrature weights are positive and bounded under small maxima (for a sufficiently large number of nodes, the weights are less than one). Then, in general, Gaussian quadratures are numerically stable, and in practice they have been shown to work well with any kind of integrand (including non-continuous and non-smooth functions).

Gaussian quadratures were extended by Kronrod (1965) and Patterson (1968), who derived methods for computing nested rules. These were later used to develop highly accurate and efficient adaptive quadrature algorithms.

For the motivation described above, we believe it is important to understand how Gaussian quadratures are built and why they ensure an order of accuracy of $2N - 1$, so in this section we present a detailed description of all the necessary steps and proofs.

In the Newton-Cotes rule (e.g. Deuflhard and Hohmann, 2003), given a set of N known and distinct nodes $\{x_i\}_{i=1}^{N}$, the problem of building a quadrature rule is to find N unknown weights $\{w_i\}_{i=1}^{N}$ while achieving an order of accuracy of $N-1$. The main idea behind Gaussian quadrature is to define a method where both nodes and weights are not given; then we can define a system with $2N$ unknowns (the nodes and weights) by obtaining a quadrature rule with an order of accuracy of $2N - 1$. Defining a nonlinear system of $2N$ equations to find the nodes and weights is generally possible, but this kind of method is not convenient to use. Therefore, the most appropriate method for constructing Gaussian quadratures is based on orthogonal polynomials.

B.1 Orthogonal polynomials

Two real polynomials $P_D(x)$ and $P_K(x)$ with degrees D and K, where all roots are distinct and lie in an interval $[a, b]$, are called orthogonal if for a given inner product $\langle \cdot, \cdot \rangle$ we have:

$$\langle P_D, P_K \rangle = a\,\delta_{N,M} = \begin{cases} 0 & \text{if } D \neq K \\ \{a \in \mathbb{R} \mid a \neq 0\} & \text{if } D = K, \end{cases} \tag{B.1}$$

where $\delta_{D,K}$ is the Kronecker delta and a is a non-null constant.

The inner product between two polynomials is defined as a weighted integral on the interval $[a, b]$:

$$\langle P_D, P_K \rangle = \int_a^b w(x) P_D(x) P_K(x)\, dx, \tag{B.2}$$

where $w(\cdot)$ is a given nonnegative weight function in the interval $[a, b]$. The function $w(\cdot)$ must have the following three properties:

1. *It must be a non negative function $w(x) \geq 0$ on the interval $[a, b]$.*

2. *All the moments $\mu_j = \int_a^b x^j w(x)\, dx \geq 0$ for $j = 0, 1, 2, \ldots$ are bounded and non negative.*

3. *For any non negative polynomial $P(x) \geq 0$ on $x \in [a, b]$:* $\int_a^b P(x) w(x)\, dx = 0 \iff P(x) \equiv 0$.

The above properties are met if $w(\cdot)$ is positive and continuous on the interval $[a, b]$ (e.g. Stoer and Bulirsch, 2013; Deuflhard and Hohmann, 2003).

A set of orthogonal polynomials $\mathcal{B} = \{P_i\}_{i=0}^{M}$ forms a linear basis on the interval $[a, b]$, if all polynomials $Q_K(\cdot)$ with a degree $K \leq M$ can be described as a linear combination of polynomials $P_i(\cdot) \in \mathcal{B}$, i.e.,

$$Q_K(x) = \sum_{i=0}^{M} c_i P_i(x), \tag{B.3}$$

where $c_i \in \mathbb{R}$ are the constants that uniquely define $Q_K(x)$ with respect to the given linear basis of polynomials.

Given the L_2 norm $\left\|P_K(x)\right\| = \sqrt{\langle P_K(x), P_K(x)\rangle}$ induced by the inner product, we can derive an orthonormal basis $\left\{P_i^e\right\}_{i=0}^{M}$ where each polynomial is defined by normalizing the original orthogonal basis, i.e.,

$$P_i^e = \frac{P_i}{\left\|P_i\right\|} \quad \forall i = 0, 1, \dots, D, \tag{B.4}$$

so that $\left\langle P_K^e, P_K^e \right\rangle = 1$.

Without describing the detailed procedure (which can be found in numerous text books (e.g. Stoer and Bulirsch, 2013; Deuflhard and Hohmann, 2003)), if we start with the polynomials $L_0(x) = 1$ and $L_1(x) = x$ by using a recursive procedure based on the Gram-Schmidt process, we can derive the set of Legendre polynomials that are orthogonal on the interval $[-1, 1]$ under the weight function $w(x) = 1$. Then, the inner product induced by the Legendre polynomials is:

$$\langle L_D(x), L_K(x)\rangle_L = \int_{-1}^{1} L_D(x) L_K(x)\, dx. \tag{B.5}$$

The Legendre polynomials of degree K are defined with the following equation:

$$L_K(x) = \frac{1}{2^K K!} \frac{d}{dx} \left(x^2 - 1\right)^K, \tag{B.6}$$

where it is straightforward to observe that all the roots of $L_K(\cdot)$ lie in the interval $[-1, 1]$.

Other common orthogonal polynomials are the Chebyshev polynomials of the first and second kind in the interval $[-1, 1]$, the Jacobi polynomials in $[-1, 1]$, the Laguerre polynomials in $[0, +\infty)$, the Hermite polynomials in $(-\infty, +\infty)$, and so on. The table B.1 lists some linear basis of polynomials with the corresponding weighting functions.

Orthogonal polynomials	**Weight function**	**Interval**
Legendre	$w(x) = 1$	$[-1, 1]$
Jacobi	$w(x) = (1-x)^{\alpha}(1+x)^{\beta}$	$[-1, 1]$
Chebyshev 1^{st} kind	$w(x) = \left(1-x^2\right)^{-1/2}$	$[-1, 1]$
Laguerre	$w(x) = e^{-x}$	$[0, +\infty)$
Hermite	$w(x) = e^{-x^2}$	$(-\infty, +\infty)$

Table B.1. *Orthogonal polynomials*

B.2 Building a Gaussian quadrature

Let us assume that we want to build a Gaussian quadrature rule with N distinct nodes and weights $\{t_i, w_i\}_{i=1}^{N}$ such that it is exact for all polynomials up to degree $2N-1$. For our purposes, we need a basis of $N+1$ orthogonal polynomials up to degree N, i.e.: $\mathcal{B} = \{P_i\}_{i=0}^{N}$ on the interval $[a, b]$.

We know that the polynomial with the higher degree $P_N(\cdot)$ form the basis $\mathcal{B}$ has N distinct roots t_i lying in the interval $[a, b]$:

$$a < t_1 < t_2 < \cdots < t_N < b. \tag{B.7}$$

Theorem 1. *The polynomial with the highest degree $P_N(\cdot)$ from the linear basis $\mathcal{B}$ is orthogonal with respect to all arbitrary polynomials $Q_D(\cdot)$ with degree $D < N$:*

$$\langle Q_D, P_N \rangle = 0 \quad \forall \ \{Q_D \in \mathbb{P}_D | D < N\} \tag{B.8}$$

Proof. From Equation (B.3) on page 238, we know that any polynomial $Q_D(\cdot)$ can be described as a linear combination of orthogonal polynomials from a linear basis. If we assume that the degree of $Q_D(\cdot)$ is $D = N-1$, the inner product between $Q_{N-1}(\cdot)$ and $P_N(\cdot)$ is:

$$\begin{aligned} \langle Q_{N-1}, P_N \rangle &= \left\langle \sum_{i=0}^{N-1} c_i P_i, P_N \right\rangle \\ &= \sum_{i=0}^{N-1} c_i \langle P_i, P_N \rangle \\ &= 0, \end{aligned} \tag{B.9}$$

where the result of the proof is simply achieved by applying the linearity of the inner product and the main definition of orthogonal polynomials. □

To prove that it is possible to build a quadrature rule that is exact for all polynomials up to a degree $2N-1$ using orthogonal polynomials, we need to define an arbitrary polynomial $Z_{2N-1}(\cdot)$ with a degree $2N-1$.

If we apply the division by polynomials between $Z_{2N-1}(\cdot)$ and $P_N(\cdot)$ we retrieve a quotient $Q_{N-1}(\cdot)$ and a reminder $R_{<N}(\cdot)$, i.e.:

$$\frac{Z_{2N-1}(x)}{P_N(x)} = Q_{N-1}(x) \quad \text{Reminder: } R_{<N}(x), \tag{B.10}$$

where the reminder $R_{<N}(\cdot)$ is a polynomial with a degree less than N.

Then $Z_{2N-1}(\cdot)$ can be written with respect to the quotient and the reminder, i.e.:

$$Z_{2N-1}(x) = Q_{N-1}(x)P_N(x) + R_{<N}(x). \tag{B.11}$$

The relation in Equation (B.11) must be considered bidirectional, since, given a polynomial with degree $2N-1$ we can uniquely derive a $Q_{N-1}(\cdot)$ and $R_{<N}(\cdot)$ or, vice versa, given $Q_{N-1}(\cdot)$ and $R_{<N}(\cdot)$ we can uniquely derive a polynomial with degree $2N-1$.

If we multiply both sides of Equation (B.11) by the weight function $w(\cdot)$ and perform the integration on $[a,b]$, we have:

$$\begin{aligned}\int_a^b w(x)Z_{2N-1}(x)\,dx &= \int_a^b w(x)Q_{N-1}(x)P_N(x)\,dx + \int_a^b w(x)R_{<N}(x)\,dx\\ &= \langle Q_{N-1}, P_N\rangle + \int_a^b w(x)R_{<N}(x)\,dx,\end{aligned} \tag{B.12}$$

where we can observe that the first integral on the right-hand side corresponds to the definition of the inner product between two polynomials (equation B.2).

From Theorem 1 we know that $P_N(\cdot)$ is orthogonal with all polynomials with a degree less than N, then, the inner product in Equation (B.12) is equivalent to 0, that is,

$$\langle Q_{N-1}, P_N\rangle = 0. \tag{B.13}$$

If we combine the results of Equations (B.12) and (B.13) we can conclude that:

$$\int_a^b w(x)Z_{2N-1}(x)\,dx = \int_a^b w(x)R_{<N}(x)\,dx, \tag{B.14}$$

so that the weighted integral of $Z_{2N-1}(\cdot)$ is equivalent to the weighted integral of the reminder $R_{<N}(\cdot)$.

If we consider the Newton-Cotes rule (Deuflhard and Hohmann, 2003, sec. 9.2), given a set of distinct and arbitrary nodes $\{x_i\}_{i=1}^{N}$ we can write the weighted integral of $R_{<N}(x)$ in Equation (B.14) as a summation of the Lagrange basis $\ell_i(\cdot)$:

$$\int_a^b w(x) R_{<N}(x)\, dx = \sum_{i=1}^{N} R_{<N}(x_i) \underbrace{\int_a^b w(x) \ell_i(x)\, dx}_{w_i} \tag{B.15}$$

The relation in Equation (B.15) is true for all polynomials up to degree $N-1$.

The next step is to select a set of constrained nodes $\{t_i\}_{i=1}^{N}$ such that the relation in Equation B.14 is true if expressed in the form of a summation. If this set of nodes exists, we can define a quadrature rule that is exact for all polynomials up to degree $2N-1$, that is,:

$$\sum_{i=1}^{N} Z_{2N-1}(t_i) \underbrace{\int_a^b w(x) \ell_i(x)\, dx}_{w_i} = \sum_{i=1}^{N} R_{<N}(t_i) \underbrace{\int_a^b w(x) \ell_i(x)\, dx\,.}_{w_i} \tag{B.16}$$

The relation in the above Equation (B.16) is true, **if and only if**, the inner product between $Q_{N-1}(\cdot)$ and $P_N(\cdot)$ expressed in the form of a summation is equal to zero, then the relation in (B.13) holds, if:

$$\langle Q_{N-1}, P_N \rangle = \sum_{i=1}^{N} w(x_i) Q_{N-1}(x_i) P_N(x_i) = 0. \tag{B.17}$$

The only set of nodes that ensures that the relation in Equation (B.17) is true are the roots of $P_N(\cdot)$ (the polynomial with the highest degree in the linear basis $\mathcal{B}$).

Therefore, a Gaussian quadrature rule consists of a set of nodes and weights: $\mathcal{G} = \{t_i, w_i\}_{i=1}^{N}$ where t_i are the roots of $P_N(\cdot)$.

Once the nodes are determined, we need to calculate the weights of the quadrature rule. As we have already anticipated in Equations ((B.15)) and ((B.16)) this can be done by using a procedure similar to the one used in the Newton-Cotes rule (see Deuflhard and Hohmann (2003)) where the weights are calculated by integrating the Lagrange basis $\ell_i(\cdot)$ and the weight function $w(\cdot)$ in the integration interval $[a, b]$, that is:

$$w_i = \int_a^b w(x) \ell_i(x)\, dx = \int_a^b w(x) \prod_{\substack{j=1\\ j\neq i}}^{N} \frac{x - t_j}{t_i - t_j} dx. \tag{B.18}$$

The weights w_i can also be calculated with the generic form (e.g. Deuflhard and Hohmann, 2003):

$$w_i = \frac{a_i}{a_{i-1}} \frac{\langle P_{i-1}, P_{i-1}\rangle}{\left[\frac{d}{dt} P_i(t_i)\right] P_{i-1}(t_i)}, \tag{B.19}$$

where a_i and a_{i-1} are, respectively, the leading coefficients of the basis polynomials $P_i(\cdot)$ and $P_{i-1}(\cdot)$.

Theorem 2. *All the weighs w_i of a Gaussian quadrature are non-null positive reals: $\{w_i \in \mathbb{R}_{>0}\}$.*

Proof. We need to consider the set of all non-negative polynomials of degree $2N-2$ that have the following form:

$$T^m_{2N-2}(x) = \prod_{\substack{j=1 \\ j\neq m}}^{N} \left(x - t_j\right)^2, \tag{B.20}$$

where t_i are the roots of $P_N \in \mathcal{B}$, so that t_m is the only node where the value of $T^m_{2N-2}(\cdot)$ does not vanish, therefore: $T^m_{2N-2}(t_m) > 0$, and $T^m_{2N-2}(t_i) = 0$ for $i \neq m$. The value of the integral of $T^m_{2N-2}(\cdot)$ is clearly larger than zero, and by using the fact that a Gaussian quadrature is exact for all polynomials with a degree up to $2N-1$, hence, for all polynomials of this family we have:

$$0 < \int_a^b T^m_{2N-2}(x)\,dx = \sum_{i=1}^{N} w_i\, T^m_{2N-2}(t_i) = w_m\, T^m_{2N-2}(t_m), \tag{B.21}$$

then, the value of the weight w_m is larger than zero, i.e.:

$$0 < w_m = \frac{\int_a^b T^m_{2N-2}(x)\,dx}{T^m_{2N-2}(t_m)}. \tag{B.22}$$

Since any Gaussian quadrature is exact on all polynomials having the form $T^m_{2N-2}(\cdot)$, and because their degree is less than $2N-1$, all the weights w_i of any Gaussian quadrature must be necessary greater than zero. □

B.3 Gauss-Kronrod Rule

The Gauss-Kronrod rule (e.g. Kronrod, 1965; Patterson, 1968; Laurie, 1997) is widely used in many mathematical applications due to its very good performance when used in adaptive algorithms. For example, Matlab's `integral` function and Mathematica's `NIntegrate` use the Gauss-Kronrod rule as the default numerical integration rule, and in the open source project `QUADPACK` (Piessens et al., 2012) the generic adaptive quadrature is based on a 21-node Gauss-Kronrod rule. In the review Gonnet (2012) adaptive methods based on the Gauss-Kronrod rule have proven to perform on average better than other quadrature methods.

The Gauss-Kronrod rule belongs to the class of nested rules.

Introduction to nested quadrature rules

By nested rules we mean the class of all quadrature rules where an original set of N nodes $T = \{t_i\}_{i=1}^{N} \in [a,b]$ is extended with a set of new nodes $R = \left\{r_j\right\}_{j=1}^{M} \in [a,b]$ such that:

$$r_j \neq t_i \quad \forall \quad j = 1 \ldots M, \quad i = 1 \ldots N. \tag{B.23}$$

We only consider cases where for each adjacent pair of nodes $t_i.t_{i+1} \in T$ there is at most one nested node $r_j \in R$, i.e.

$$\forall i, \quad \exists j \quad | \quad t_i < r_j < t_{i+1} < r_{j+1}. \tag{B.24}$$

In such a way that a new set of integration nodes E is given by the union of the original set of nodes T and the set of new nodes R:

$$E = T \cup R, \tag{B.25}$$

then, we can say that the set of nodes T is nested with respect to E.

Nested rules are often associated with adaptive methods based on the trapezoidal rule. The most efficient method based on a nested rule applied to the trapezoidal rule is the Romberg adaptive algorithm (e.g. Deuflhard and Hohmann, 2003).

More in general, nested rules allow to refine the trapezoidal rule by calculating the values of the function only at new nodes.

Given a set of N nodes from a Gaussian quadrature rule (which we denote with the superscript G)

$$\left\{x_i^G\right\}_{i=1}^{N} \in [-1,1], \tag{B.26}$$

a Gauss-Kronrod rule extends the original set of nodes to a set of $2N+1$ nodes (denoted by the superscript K) so that the Gauss nodes are nested with respect to the Kronrod nodes:

$$\begin{aligned} &\left\{x_j^K\right\}_{j=1}^{2N+1} \in [-1,1] \\ &x_{2i}^K = x_i^G \quad \forall \quad i = 1 \ldots N. \end{aligned} \tag{B.27}$$

Thus, a Gauss-Kronrod consists of a set of $2N+1$ nodes and weights $\mathcal{K} = \left\{x_j^K, w_j^K\right\}_{j=1}^{2N+1}$ such that the equation:

$$\int_{-1}^{1} P_{d_n}(x)\,dx = \sum_{j=1}^{2N+1} w_j^K P_{d_n}\left(x_j^K\right), \tag{B.28}$$

where: $d_n = \begin{cases} 3N+1 & \text{if} \quad N \text{ is even} \\ 3N+2 & \text{if} \quad N \text{ is odd,} \end{cases}$ so that, Equation (B.28) is exact for all polynomials $P \in \mathbb{P}_{d_n}$ (Notaris, 1993).

If we consider the main integration interval $[-1,1]$ the upper bound of the error for a Gauss-Kronrod rule is:

$$\left|E_N^K(f)\right| < \frac{(N!)^2}{2^{N-3}(2N)!(2N+2)!} \max_{-1\le t\le 1} \left|f^{(d_n+1)}(t)\right| \tag{B.29}$$

where $f(\cdot)$ is a smooth function with d_n+1 continuous derivatives (Notaris, 1993).

The Gauss-Kronrod rule is typically used in the development of highly efficient and accurate adaptive integration algorithms.

B.4 Kronrod-Patterson Extensions

With the Gauss-Kronrod rule, we can only extend a Gaussian quadrature only with one set of $N+1$ nested nodes. This limitation was broken by Patterson who proposed a numerically stable procedure to iterate the Kronrod extension, in a way to obtain a tower of nested nodes (e.g. Patterson, 1968; Bourquin, 5 04) by progressively increasing the order of the quadrature.

The Kronrod-Patterson extensions consist of a tower of nested quadrature rules, such that each rule extends the previous one by adding $N+1$ nodes, i.e. $\{x_i\}_{i=1}^{N} \subset \{x_i\}_{i=1}^{2N+1}$.

Then we have a tower of quadrature rules $\left\{\mathcal{P}_j\right\}_{j=0}^{M}$, so that $\mathcal{P}_0$ is a midpoint rule on an interval $[-1,1]$ with degree 1, its extension $\mathcal{P}_1$ is a Gaussian quadrature rule on 3 nodes with degree 5, and finally for $j \geq 2$ we have the tower of Kronrod-Patterson quadrature rules with $N = 2^{j+1} - 1$ nodes and degree $d = 2^{j+1} + N - 1$, considering that we always have an odd number of nodes, the effective degree is $d = 2^{j+1} + N$.

Then a Kronrod-Patterson rule $\mathcal{P}_j = \{x_i, w_i\}_{i=1}^{N=2^{j+1}-1}$ is a quadrature that is exact for all polynomials $P \in \mathbb{P}_{2^{j+1}+N}$ for all $j \geq 2$.

The use of a Gauss-Legendre rule as a starting point is justified by the fact that the extensions based on Legendre polynomials ensure positive weights.

Recently, Bourquin (5 04) proposed an algorithm to compute the tower of nested nodes up to an arbitrary quadrature degree. Unfortunately, the computation of the Patterson extensions is computationally hard, since the time complexity grows exponentially with the number of levels, and it requires the use of *variable precision* floating point numbers. These two points limit the development of effective algorithms based on the Pattersons extensions.

Appendix C

Adaptive quadrature

C.1 Introduction to the adaptive numerical integration

The problem of numerical integration consists in approximating the value of an integral $I = \int_a^b f(x)\,dx$ in a way that the error of the approximated result $\tilde{I}$ is smaller than a given tolerance δ, i.e. compute the approximation $\tilde{I}$ of I such that

$$\left|I - \tilde{I}\right| < \delta\,|I|\,. \tag{C.1}$$

With an approach based on a fixed-node quadrature rule (e.g., the Gauss-Legendre rule), the prior knowledge of the integrand $f(\cdot)$ is usually used to construct a sequence of Gaussian quadrature rules that exploit the behavior of $f(\cdot)$, usually by exploiting the additivity of the integral by generating a sequence of sub-interval where their shape and order are adjusted according to the predicted behavior of the integrand. This approach only works if the prior knowledge is robust enough to predict the behavior of the integrand with good reliability. On the other hand, we must consider that there are numerous circumstances in which we do not have sufficient prior knowledge of the integrand, or, despite the availability of prior knowledge, it may be difficult to construct an appropriate quadrature rule.

The purpose of an adaptive quadrature algorithm is to define a procedure that is capable of automatically adjusting the sequence of subintervals and eventually the order of the quadrature rules with an iterative algorithm that is capable of detecting the behavior of the integrand $f(x)$. In such a way that the desired accuracy is achieved by minimizing the number of evaluations of the integrand.

In an adaptive quadrature, we usually give an interval of integration $[a, b]^i$

and make an approximation $\tilde{I}_i^1$ of the integral on that interval:

$$\tilde{I}_1^i \approx \int_a^b f(x)\,dx, \tag{C.2}$$

where the subscript indicates the index of a sub-interval and the superscript indicates the level of refinement.

Since the value $\tilde{I}_1^i$ is an approximation of the integral, its error with respect to the true value of the integral I is unknown. For this motivation, it is necessary to have a function that is capable of approximating the integration error in the interval $[a,b]^i$, i.e.

$$\tilde{\epsilon}^i = \text{error}\left([a,b]^i, f(\cdot)\right). \tag{C.3}$$

The approximation of the quadrature error is an active field of research, see e.g. Gonnet (2012).

In an adaptive algorithm, if the error is greater than a given tolerance δ, we refine the main interval $[a,b]^i$, which is divided into two adjacent subintervals $[a,m]^{i+1}$ and $[m,b]^{i+1}$, where m can be the midpoint. Finally, we compute two approximate integrals:

$$\tilde{I}_1^{i+1} \approx \int_a^m f(x)\,dx \qquad \tilde{I}_2^{i+1} \approx \int_m^b f(x)\,dx, \tag{C.4}$$

where we assume that the numerical integration resulting from the subdivision of the integration interval is a better approximation of I with respect to $\tilde{I}_1^i$, i.e:

$$\left|I - \left(\tilde{I}_1^{i+1} + \tilde{I}_2^{i+1}\right)\right| \leq \left|I - \tilde{I}_1^i\right|. \tag{C.5}$$

In a very general way, an adaptive quadrature algorithm can be described as a recursive function that interactively refines the approximate solution of the integral by progressively dividing the integration intervals only in regions where the estimated error $\tilde{\epsilon}$ is greater than the given tolerance δ. Hence, an adaptive quadrature can be summarized in the following recursive function:

$$\text{adQuad}\left([a,b]^i\right) := \begin{cases} \tilde{I}_1^i & \text{if} \quad \tilde{\epsilon}^i < \delta \\ \text{adQuad}\left([a,m]^{i+1}\right) + \text{adQuad}\left([m,b]^{i+1}\right) & \text{otherwise,} \end{cases} \tag{C.6}$$

where $\tilde{I}_1^i$ is the approximation of the integral on the interval $[a,b]^i$, δ is the given tolerance, and $[a,m]^{i+1}$ and $[m,b]^{i+1}$ are the intervals on the next refinement level $i+1$.

The algorithm described in Equation C.6 is based on a *depth-first* purely recursive approach. In this method, the integral is solved from left to right by progressively refining the interval on the left, and only when the acceptance condition is met, it moves to the next adjacent interval.

Then, in the algorithm C.6, the intermediate values of the approximated value of the integral during the computation are always an approximation of the integral over an interval $[a, m_j]$, i.e.

$$\tilde{I}_j \approx \int_a^{m_j} f(x)\, dx,$$

so that the recursion (i.e. *depth-first*) method sequentially solves the integral in the intervals: $[a, m_1]$, $[a, m_2]$, $\ldots$, $[a, b]$.

Another possible approach is to use the *breadth-first* strategy, where we progressively compute the corrections of $\tilde{I}$ level by level. Then, the subintervals in the next level of refinement $(i+1)$ will only be calculated when all integrals over the intervals in the current level i have been completed, i.e.

$$\tilde{I}^i \approx \int_{a^i}^{b^i} f(x)\, dx,$$

where we must assume that the approximation at the deepest level: $\tilde{I}^{i+1}$ is more accurate than $\tilde{I}^i$. With this method, we continue until all sub-intervals of a level have achieved the desired tolerance δ.

Introduction to depth-first and breadth-first

The *depth-first* search is a generic algorithm that, starting from the root node of a tree data structure, always recursively explores the first child node, along a branch, until a leaf is reached, before backtracking.

The *breadth-first search* is a generic algorithm that, starting from the root node of a tree data structure, it first explores all neighboring nodes at the current depth. And only after all nodes of the current depth have been explored, it move to the next level.

In algorithm theory, the *depth-first* search is usually associated with the *stack* data structure, while the *breadth-first* search is associated with the *queue* data structure.

By *stack* we mean a data structure with two operators: *push* $(\cdot)$ and *pop* $(\cdot)$, where *push* $(\cdot)$ inserts an element into the stack, while *pop* $(\cdot)$ returns and removes the **last** inserted element.

While by *queue* we mean a data structure with the two operators *push* $(\cdot)$ and *pop* $(\cdot)$, but in this case *pop* $(\cdot)$ returns and removes the **first** inserted element.

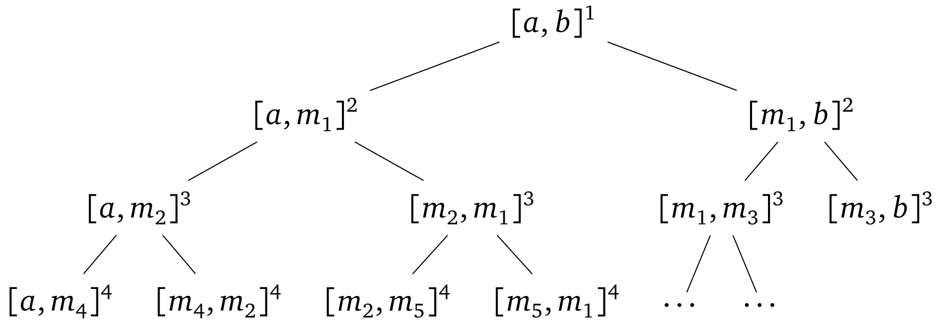

Figure C.1. *The refinement levels of an integration interval can be interpreted as a binary tree. An adaptive algorithm searches all leaves where the local errors are less than a given tolerance.*

In an iterative adaptive quadrature algorithm, if we put the intervals of the next level $i+1$ into a stack, we reproduce the recursive (or depth-first) behavior described in Equation C.6, while if we use the queue, we have an algorithm based on the breadth-first strategy.

The adaptive quadrature based on the depth-first strategy using a stack is described in Algorithm 12, where we observe that the result of the quadrature

$\tilde{I}$ is a cumulative sum obtained from the approximations on the intervals at the deepest levels of refinement.

Algorithm 12: Generic depth-first adaptive quadrature algorithm based on a stack

```
Input: $f(\cdot)$, $[a,b]$, $\delta$
Data: stack
$\tilde{I} := 0$
stack.push($[a,b]$)
while stack is not empty do
    $[a^l, b^l] :=$ stack.pop()
    if error$\left([a^l, b^l], f(\cdot)\right) \geq \delta$ then
        $m^l := \frac{a^l+b^l}{2}$
        stack.push($[a^l, m^l]$)
        stack.push($[m^l, b^l]$)
    else
        $\tilde{I}^l \approx \int_{a^l}^{b^l} f(x)\,dx$
        $\tilde{I} := \tilde{I} + \tilde{I}^l$
return $\tilde{I}$
```

While in Algorithm 13 is possible to observe that the method based on the breadth-first strategy progressively corrects an initial raw computation of $\tilde{I}$ until it achieves the desired accuracy on all subintervals.

The asymptotic time complexity of an adaptive quadrature algorithm is $O\left(2^{\ell}\right)$, where ℓ is the maximum number of refinement levels; in other words, the number of function evaluations required in the worst case grows exponentially with the number of refinement levels. For this reason, it is crucial to use an accurate method for approximating the integral and the error. Note that the reliability of the error estimator (e.g. Deuflhard and Hohmann, 2003; Gonnet, 2012) strongly influences the efficiency of an adaptive quadrature algorithm, then, it is the most critical component.

Algorithm 13: Generic breadth-first adaptive quadrature based on a queue

```
Input: $f(\cdot)$, $[a,b]$, $\delta$
Data: queue
$\tilde{I} \approx \int_{a^l}^{b^l} f(x)\,dx$
queue.push($[a,b]$)
while queue is not empty do
    $[a^l, b^l] :=$ queue.pop()
    $m^l := \frac{a^l+b^l}{2}$
    if error$\left([a^l, b^l], f(\cdot)\right) \geq \delta$ then
        $\tilde{I}^l \approx \int_{a^l}^{b^l} f(x)\,dx$
        $\tilde{I}^l_1 \approx \int_{a^l}^{m^l} f(x)\,dx$
        $\tilde{I}^l_2 \approx \int_{m^l}^{b^l} f(x)\,dx$
        $\tilde{I} := \tilde{I} - \tilde{I}^l + \tilde{I}^l_1 + \tilde{I}^l_2$
        queue.push($[a^l, m^l]$)
        queue.push($[m^l, b^l]$)
return $\tilde{I}$
```

C.2 error estimate in the Gauss-Kronrod rule

Since in a quadrature method the analytical solution of the integral is unknown, the exact quadrature error ϵ is also unknown. Therefore, it is necessary to define a function (hereafter: error estimator) that can compute a reliable approximation of the error $\tilde{\epsilon}$. An error estimator should be efficient in such a way that its procedure does not significantly affect the number of function evaluations of the integrand $f(\cdot)$, ideally, the error estimation should be computed by using only the function evaluations performed for the quadrature computation. Considering the exact quadrature error ϵ and an estimated error $\tilde{\epsilon}$, we can state that

$$(1 - c_1)\,\epsilon \leq \tilde{\epsilon} \leq (1 + c_2)\,\epsilon, \tag{C.7}$$

where $c_1, c_2 \geq 0$ and $c_1 \leq 1$ are unknown constants that define the "lower bound" and "upper bound" of the estimated error. If $c_1, c_2 = 0$, the *estimated error* $\tilde{\epsilon}$ is equal to the *exact error* (ϵ). An error estimator can be considered reliable if c_1 and c_2 are small.

In the scientific literature, the criteria that an error estimator should have are widely discussed, for example, in finite element analysis Grätsch and Bathe

(2005), and numerical quadrature methods (Laurie, 1985) numerous criteria have been proposed, i.e.

- For a quadrature interval with a width larger than 0, if the estimate of the error is 0 the exact error must also be 0.

- The error estimate must be asymptotically correct; i.e., for quadrature subintervals tending to zero, the estimate of the error must tend to zero with the same rate as the exact error.

- The error estimator must return results that are guaranteed to be within a lower bound and an upper bound. Or, similarly, the exact error must be bounded as a function of the estimated error.

- It must be robust, meaning that it returns a sufficiently reliable estimate even in the case of non-smooth or non-continuous functions. In the case of error estimators used in quadrature, it is also important to evaluate the ability to detect very sharp peaks, singularities (e.g., divisions by zero), or other perturbations in the function (e.g., local oscillations).

- An error estimation should be used in an adaptive algorithm as a tool to decide whether to continue refining the quadrature interval or accept the current result.

Therefore, an ideal quadrature error estimator is a function that should be defined as follows:

$$\text{error}\left([a,b], f(\cdot)\right) = \begin{cases} 0 & \text{if} \quad Q_N[a,b] - I[a,b] = 0 \\ \tilde{\epsilon} & \text{s.t.} \quad (1-c_1)\epsilon \le \tilde{\epsilon} \le (1+c_2)\epsilon \end{cases} \tag{C.8}$$

where $Q_N[a,b]$ is the result of the quadrature operation, $I[a,b]$ is the real value of the integral (which by definition is unknown). While c_1 and c_2 are the values that define the upper and lower bound of the quadrature error.

Considering that it is impossible for an ar error estimator to satisfy all the aforementioned criteria, both in the scientific literature and in software applications, numerous strategies have been proposed (e.g. Gonnet, 2012). In the following sections, we will present three methods that are suitable to be used in an adaptive quadrature based on the Gauss-Kronrod rule.

C.2.1 Preliminary notes and used notation

As we saw in Appendix B.3 on page 243, a Gauss-Kronrod quadrature rule consists of two nested quadrature rules on a set of $2N+1$ nodes x_j: a Gaussian rule (defined only on even nodes) $\mathcal{G} = \left\{x_i, w_i^G\right\}_{i=2,4,\dots}^{2N}$, and the Kronrod rule (defined on all nodes) $\mathcal{K} = \left\{x_j, w_j^K\right\}_{j=1}^{2N+1}$, where the Gaussian quadrature has an order of accuracy of $2N-1$ (see Appendix B on page 237) and for Kronrod the order of accuracy is $3N+1$ or $3N+2$ (see Appendix B.3 on page 243). The N nodes of the Gaussian rule are indexed with the even i, while the Kronrod rule is based on all $2N+1$ nodes.

Then, given a Gauss-Kronrod rule, we have two estimates of the integral $I[a,b]$, one performed on the N nodes of the nested Gaussian rule $\mathcal{G}$, i.e:

$$G_N[a,b] = \sum_{i=2,4,\dots}^{2N} w_i^G f\left(x_i\right), \tag{C.9}$$

and the second one performed with the $2N+1$ nodes of the Kronrod rule $\mathcal{K}$, i.e.

$$K_{2N+1}[a,b] = \sum_{j=1}^{2N+1} w_j^K f\left(x_j\right), \tag{C.10}$$

where, if in the interval of integration $f \in C^{3N+2}$ (the integrand is *smooth enough*), we can assume that $K_{2N+1}[a,b]$ is a better approximation of I with respect to $G_N[a,b]$, i.e.:

$$\left|I[a,b] - K_{2N+1}[a,b]\right| < \left|I[a,b] - G_N[a,b]\right| .$$

Then algorithms based on the Gauss-Kronrod rule will then use $K_{2N+1}[a,b]$ as the better approximation of $I[a,b]$.

C.2.2 Absolute difference and local-to-global errors estimators for a Gauss-Kronrod quadrature

In a Gauss-Kronrod quadrature rule we have two results, one calculated with the Gaussian rule $G_N[a,b]$ and the second one is calculated with the Kronrod rule $K_{2N+1}[a,b]$ (B.27), where we assume that the quadrature calculated with the Kronrod rule is more accurate than the one calculated with the Gaussian rule. Therefore, the most direct approach is to use the absolute difference between the Gaussian and Kronrod quadratures as the error estimate:

$$\epsilon_{\text{k}} = \left|K_{2N+1}[a,b] - G_N[a,b]\right| . \tag{C.11}$$

Despite its simplicity, this error estimator has proven to be very stable and provides good accuracy in adaptive quadrature algorithms. In (Gonnet, 2012) this estimator has proven to be superior to other and more sophisticated methods (Gonnet (2012) claims in his conclusion that this estimator is the best). This estimator is also used in Matlab's `integral` and `quadgk` functions. According to our tests (see C.3.1), we can confirm that this error estimator, when used in an adaptive algorithm based on the Gauss-Kronrod rule, guarantees a good accuracy of the results.

The absolute difference estimator can be easily used to calculate a local error relative to a global raw estimate of the integral:

$$\epsilon_{\mathrm{kg}} = \left| \frac{K_{2N+1}[a^j, b^j] - G_N[a^j, b^j]}{Q^{Raw}[a, b]} \right|, \tag{C.12}$$

In a practical numerical application, it is necessary to redefine the global error estimator in such a way as to handle cases where the result of the integral is zero. To solve this problem, we propose to use the following modified equation:

$$\epsilon_{\mathrm{kg}} = \left| \frac{K_{2N+1}[a^j, b^j] - G_N[a^j, b^j]}{1 + |Q^{Raw}[a, b]|} \right|, \tag{C.13}$$

where the absolute value of the raw estimate of the global quadrature is summed by 1. This method, in practice, solves the problem of division by zero without affecting the quality of error estimation.

Reliability of the absolute difference error

If we consider the property of being able to return a zero-error estimate when the exact error is zero, we can observe that this property is only partially satisfied, since it requires that both the Gaussian and Kronrod quadratures must be exact.

If we assume that the result returned by the quadrature algorithm is the one computed with the Kronrod rule $K_{2N+1}[a^j, b^j]$ (which has an order of accuracy of $3N + 1$), it is possible that the result computed with the associated Gaussian nested rule $G_N[a^j, b^j]$ (which has an order of accuracy of $2N - 1$) is not exact. Then, for this estimator, there may be situations where the error estimates are non-zero for exact results. On the other hand, if $G_N[a^j, b^j]$ is exact, then $K_{2N+1}[a^j, b^j]$ is exact, but if $K_{2N+1}[a^j, b^j]$ is exact, then $G_N[a^j, b^j]$ is not necessarily exact. This situation could appear if the integrand is a polynomial P_D with a degree $2N - 1 < D \leq 3N + 1$.

For the error estimators based on the absolute difference, it is impossible to define an interval within which the real quadrature error lies. Then we can conclude that the error estimators are only partially reliable, in the sense that there are situations where the reliability conditions described in Appendix C.2 on page 252 are not fully satisfied.

C.2.3 Sharper error estimator

This error estimator was originally introduced in (Laurie, 1983). In the review of (Gonnet, 2012) it has been shown to perform well. Although Laurie's paper (Laurie, 1983) presents this error estimator as a generic method that can be used with different quadrature rules (Laurie suggests applications based on Gaussian and trapezoidal rules), the sharper error estimator fits very well with the Gauss-Kronrod rule. For this reason, this section discusses the sharper error estimator as an application for the Gauss-Kronrod rule.

To compute this error estimator, we divide the original integration interval $[a,b]$ into two adjacent intervals, $[a^1,b^1]$ and $[a^2,b^2]$, such that:

$$I=\int_a^b f(x)\,dx=\int_{a^1}^{b^1} f(x)\,dx+\int_{a^2}^{b^2} f(x)\,dx.$$

In the main interval $[a,b]$, the quadrature performed with the Gauss-Kronrod rule is denoted with $K_1=K_{2N+1}[a,b]$, while the quadrature performed with the Gaussian rule is $G_1=G_N[a,b]$. Where it is possible to assume that K_1 is more accurate than G_1, since the order of accuracy of K_1 is greater than that of G_1.

In the sub-interval $[a^1,b^1]$ we have the Gauss-Kronrod quadrature $K_1^2=K_{2N+1}^1[a^1,b^1]$ and the Gaussian quadrature $G_1^2=G_N^1[a^1,b^1]$. Similarly, over the interval $[a^2,b^2]$ we have $K_2^2=K_{2N+1}^2[a^2,b^2]$ and $G_2^2=G_N^2[a^2,b^2]$.

Finally, we define two quadratures on the interval $[a,b]$ as the sums of the Gauss-Kronrod quadratures on the two subintervals $K_2=K_1^2+K_2^2$ and the other as the sum of the Gauss quadratures $G_2=G_1^2+G_2^2$ over the two subintervals.

Therefore, the sharper error estimator $\tilde{\epsilon}_{\text{sh}}$ is defined as follows:

$$\tilde{\epsilon}_{\text{sh}}=\frac{(K_2-G_2)(K_2-K_1)}{(G_2-G_1)-(K_2-K_1)}, \tag{C.14}$$

Where $\left|\tilde{\epsilon}_{\text{sh}}\right|$ is a reliable estimate, since the real value of the integral I is bounded within a certain interval as a function of the error.

Theorem 3. *If K_1, K_2, G_1 and G_2 are numbers such that the conditions in Equation (C.15) hold:*

$$\begin{aligned}&(i)\quad 0\le\frac{I-K_2}{I-K_1}\le\frac{I-G_2}{I-G_1}\le 1\\ &(ii)\quad |K_2-K_1|<|G_2-G_1|.\end{aligned} \tag{C.15}$$

The real value of the integral lies in the interval bounded by K_2 and $K_2 + \tilde{\epsilon}_{sh}$ (Laurie, 1983):

$$I = \int_a^b f(x)\,dx \in [K_2, K_2 + \tilde{\epsilon}_{sh}], \tag{C.16}$$

or equivalently we can write:

$$\frac{\tilde{\epsilon}_{sh}}{(I - K_2)} \geq 1. \tag{C.17}$$

Proof. We follow the procedure introduced in (Laurie, 1983) where he proved that equation (C.17) is true if the conditions of equation (C.15) hold.

Let us define the variables $k_1 = I - K_1$, $k_2 = I - K_2$, $g_1 = I - G_1$ and $g_2 = I - G_2$. By substituting the new variables into the definition of $\tilde{\epsilon}_{\text{sh}}$ (C.14), and by substituting it into the equation (C.17), with some algebraic manipulation, we can obtain:

$$\frac{\tilde{\epsilon}_{\text{sh}}}{(I - K_2)} - 1 = \left(\frac{g_2}{g_1} - \frac{k_2}{k_1}\right)\left(\frac{k_1}{k_2}\right)\left(\frac{g_1}{g_1 - g_2 - k_1 + k_2}\right) \geq 0. \tag{C.18}$$

If we assume that K_2 is a more accurate approximation than K_1 and equivalently G_2 is more accurate than G_1 we can say that $g_1 - g_2 - (k_1 - k_2)$ has the same sign as $g_1 - g_2$ which, in turn, has the same sign as g_1, then, the last term is positive. The second term is positive because we assume that k_1 and k_2 have the same sign of the integral. And the first term it true because of the Condition (i) in the Equation (C.15).

Then, Theorem 3 is true.

□

In the case of a smooth integrand, and considering that K_2 has a higher degree than K_1, $\left|\tilde{\epsilon}_{\text{sh}}\right|$ is usually less than $|K_2 - G_2|$, because it is typical that $|K_2 - K_1| \ll |G_2 - G_1|$.

In a numerical integration algorithm, condition (ii) can be easily verified, while condition (i), because it involves the analytical value of the integral I, is impossible to be verified. The heuristic proposed by Laurie to check the condition (i) is to replace I by $K_2 + \tilde{\epsilon}_{\text{sh}}$. Thus, in the case of a Gauss-Kronrod rule where the degrees of the two quadratures are different, the conditions (i) and (ii) become:

$$\begin{aligned} &(i^*) \quad 0 \leq \frac{K_2 - G_2}{K_2 - G_1} \leq 1 \\ &(ii^*) \quad |K_2 - K_1| < |G_2 - G_1| \end{aligned} \tag{C.19}$$

The new condition (i^*) can be easily and intuitively interpreted with the fact that if the integrand is smooth enough, G_2 is a better approximation of G_1, then the ratio is less or equal to 1.

In a practical implementation of the sharper error estimator in an adaptive quadrature, it is necessary to establish a policy to be adopted if at least one of the two conditions in equation (C.19) fails.

In general, we have observed that if the integrand is smooth enough, the rules in equation (C.19) can fail only if the estimated error $\tilde{\epsilon}_{\text{sh}}$ is large and others adaptive refinements are required in any case. Then, in cases of *smooth enough* integrands, the better policy is "*do nothing*", because the problem usually disappears with the adaptive and progressive refinement of the subintervals.

This is because, if the integrand is smooth enough, the Kronrod rule will in any case give a more accurate estimate of the integral than that obtained with the Gaussian rule, thus ensuring convergence. Consequently, the failure of one of the two conditions will only occur if the estimation of the integral is coarse and done on a very large integration interval (or, in other words, it will fail in situations where the sampling of the integrand is not dense enough to perceive the shape and smoothness of the integrand).

In more general cases where the integrand may be non-smooth or non-continuous, a good policy we have adopted in the case of failure of the conditions (C.19) is to replace the sharper error estimator with the QUADPACK error estimator (C.21), which has been shown to work well even with non-smooth or non-continuous integrands.

Reliability of the Sharper error estimator

From Equation C.16 we can conclude that the sharper error estimate bounds the real value of the integral I, then consequently the exact error of quadrature is bounded.

Theorem 4. *If the real quadrature error is zero, then the estimated error $\tilde{\epsilon}_{sh}$ is also zero.*

Proof. (wlog) we can assume that the two results obtained with the Kronrod rule K_2 and K_1 are based on an odd N, then they are exact for all polynomials $P \in \mathbb{P}_{3N+2}$. Consequently, if K_1 is exact, then K_2 is exact (since the exactness depends only on N and not on the intervals). We must also note that if K_1 and K_2 are exact, the integrand is a polynomial. Then the difference between K_1 and K_2 in the numerator of the formulation of the sharper error estimator (see

Equation C.14) is zero, so the estimated error is zero. This proves that the initial assumption is true. □

Theorems 3 and 4 ensure that the sharper error estimator is accurate, bounded, and asymptotically correct, then it has a high degree of reliability, at least for smooth enough integrands.

C.2.4 QUADPACK error estimator

This error estimator is the one used in the generic adaptive quadrature algorithm available in the QUADPACK library (Piessens et al., 2012) (which is based on a 21-node Gauss-Kronrod rule).

This method was developed empirically, and no theorem explains a real mathematical motivation of its effectiveness (Gonnet, 2012; Piessens et al., 2012). Nevertheless, this estimator has been shown to work well with all kinds of functions. Thus, it was successfully adopted in the QUADPACK library as a default method.

The QUADPACK estimator is defined by the equations (C.20) and (C.21), where the integral $\tilde{I}_k$ is estimated using the values of the integrand at the Kronrod nodes, and consequently $\tilde{I}_k$ is solved using the Kronrod extensions weights.

$$\tilde{I}_k = \int_a^b \left| f(x) - \frac{K_{2N+1}[a,b]}{b-a} \right| dx \tag{C.20}$$

$$\tilde{\epsilon}_{qp} = \tilde{I}_k \min\left(1, \left(200 \frac{|G_N[a,b] - K_{2N+1}[a,b]|}{\tilde{I}_k}\right)^{\frac{3}{2}}\right) \tag{C.21}$$

A qualitative mathematical interpretation of this estimator has been proposed in Krommer and Überhuber who gave these explanations (Gonnet, 2012; Krommer and Ueberhuber, 1998):

> "a measure for the smoothness of $f(\cdot)$ on $[a,b]$"

> "If this ratio is small, the difference between the two quadrature formulas is small compared to the variation of $f(\cdot)$ on $[a,b]$; i.e., the discretization of $f(\cdot)$ in the quadrature formulas $G_N[a,b]$ and K_{2N+1} is fine with respect to its variation. In this case, K_{2N+1} can indeed be expected to yield a better approximation for I than G_N."

In the test that we have done C.3.1 this estimator has proved his effectiveness with all functions we used in the tests.

Reliability of the QUADPACK error estimator

The QUADPACK error estimator is reliable in the sense that if the real quadrature error is zero: $\epsilon_{\text{abs}} = 0$, then the estimated error is also zero: $\tilde{\epsilon}_{qp} = 0$. This can be easily proved by assuming that the result of the Kronrod quadrature is exact: $I = K_{2N+1}[a,b]$.

Without loss of generality, in this proof we can remove the absolute value from the definition of $\tilde{I}_k$, then we have

$$\tilde{I}_k = \int_a^b f(x) - \frac{I}{b-a} dx, \tag{C.22}$$

by using the linearity of the integral, we can prove the theorem with some simple steps:

$$\begin{aligned} \tilde{I}_k &= \int_a^b f(x)\,dx - \int_a^b \frac{I}{b-a} dx \\ \tilde{I}_k &= I - \int_a^b \frac{I}{b-a} dx \\ \tilde{I}_k &= I - I\left(\frac{b}{b-a} - \frac{a}{b-a}\right) = 0, \end{aligned} \tag{C.23}$$

then the QUADPACK error estimator returns zero if the value of the quadrature is exact.

Because of the nature of this error estimator, it is impossible to define an upper and lower bound for the real value of the integral.

C.3 Gauss-Kronrod adaptive quadrature algorithm

The algorithm we developed uses the error estimates described in sections C.2.3 and C.2.4, combined with a branch-cutting strategy based on a local-to-global error estimate Algorithm 13

In the algorithm, we start with a raw estimate of $I = \int_a^b f(x)\,dx$ (as in Algorithm 13) and the order of the next interval to be processed is decided by a priority queue (heap), namely the sub-interval where the estimated error is maximal has the maximal priority and is processed before the others. This strategy allows us to quickly correct the initial estimation. The local-to-global error (at line 21) allows stopping the iterative refinement if the absolute error of a given sub-interval is not globally significant.

Algorithm 14: Gauss-Kronorod adaptive algorithm

```
Input: $f(\cdot)$
Input: $epsilon := 1 \cdot 10^{-17}$, $tolerance := 1 \cdot 10^{-17}$
Output: $I$
struct {
    $[a^l, b^l]$   : Local integration interval
    $I^l$          : Local quadrature
    $\epsilon_l$   : Quadrature error on the local interval.
    $priority$     : A positive number proportional to the error $\epsilon_l$
} Quadrature Task;
$Q$ := Priority Queue of Quadrature Tasks.
Function AddTaskToQueue ($[c, d]$, $\epsilon$, $I^l$) :
    if $\epsilon > tolerance$ then
        $task :=$ MakeQuadratureTask ( $[c, d]$, $I^l$, $\epsilon$ )
        PushWithPriority ( $Q$, $task$ )
$I, \epsilon_0 :=$ PerformGKQuadrature ( $f(\cdot)$, $a$, $b$ )
AddTaskToQueue ($[a, b]$, $\epsilon_0$, $I$ )
while Size ($Q$) $> 0$ do
    $task :=$ PopMaxPriority ($Q$)
    $I^l_{old} := task.I^l$
    $a^l, b^l := task.[a^l, b^l]$
    $m^l := \frac{a^l + b^l}{2}$
    $I^l_1, \epsilon_1 :=$ PerformGKQuadrature ( $f(\cdot)$, $a^l$, $m^l$ )
    $I^l_2, \epsilon_2 :=$ PerformGKQuadrature ( $f(\cdot)$, $m^l$, $b^l$ )
    $I^l_{new} := I^l_1 + I^l_2$
    $I := I - I^l_{old} + I^l_{new}$
    $d_q := \left| I^l_{old} - I^l_{new} \right|$
    if $\frac{d_q}{1 + d_q + |I|} < epsilon$ then
        Continue
    AddTaskToQueue ($[a^l, m^l]$, $\epsilon_1$, $I^l_1$ )
    AddTaskToQueue ($[m^l, b^l]$, $\epsilon_2$, $I^l_2$ )
return $I$
```

The main algorithm is controlled by the priority queue Q (line 4), which stores the set of quadrature tasks, and it only stops when Q is empty.

To avoid over-refinement, adaptive algorithms usually implement a mechanism to limit the maximum depth of the iterative process. This mechanism is useful in the case of non-smooth integrands. In the algorithm presented here, we do not include this mechanism since practical tests showed that it was never necessary, since the error estimators presented here can control the algorithm with very good reliability.

C.3.1 Test procedure

In the test procedure, we used the adaptive quadrature described in the algorithm 14 with the corresponding error estimators. For making a comparison with other other implementations, we chosen as reference the generic adaptive quadrature from the library `QUADPACK` (Piessens et al., 2012) and the built-in adaptive Romberg quadrature from the library `scipy`[1].

All algorithms are configured so that there are no restrictions on the depth of recursion. In particular, for `QUADPACK` and Romberg we set a very large number of maximum recursion calls.

In order to force all algorithms to return a result that is as accurate as possible, the tolerances (absolute or relative) are set to 10^{-17} (the algorithms use the tolerance to stop the iteration process if the estimate of the error is less than this value). Note that different algorithms and error estimators may interpret and use these tolerances differently.

In the results Table C.1 the algorithms are referred as follow:

- **Adaptive GK SH err:** The Gauss-Kronrod adaptive algorithm 14 with the sharper error estimator C.2.3.
- **Adaptive GK ABS err:** The GK adaptive algorithm 14 with the absolute difference error estimator C.2.2.
- **Adaptive QP err:** The GK adaptive algorithm 14 with the QUADPACK error estimator C.2.4.
- **QUADPACK:** Generic adaptive quadrature from `QUADPACK`. In this case we use the builtin interface `integrate.quad` form the `scipy` python library.

[1] scipy Romberg

- **Romberg:** Adaptive Romberg quadrature. We have used the builtin function `integrate.romberg` from the `scipy` library.

In Table C.1 in columns are reported the following quantities:

- **f-evaluations:** The total number of functions evaluations.
- **relative error vs. reference:** The relative errors with respect to the reference results.
- **estimated error:** The estimated errors returned by the algorithms. *It should be noted that these errors are calculated according to the internal heuristics of each algorithm and their meaning is different.*

Note that the Romberg algorithm from `scipy` does not return an estimate of the error. Therefore, the corresponding fields in the table are left blank.

C.3.2 Generic test functions

The following generic functions were used to test the adaptive quadratures. We report the corresponding integrals and the reference results.

$$
\begin{aligned}
f_1(x) &= \sin\left(\frac{\alpha}{\alpha^2+x^2}\right) - \cos^5\left(5x^2\right) + \frac{1}{2} \\
I_1 &= \int_{-1}^{10} f_1(x)\,dx = 5.6092404089309748290862030455 1 \quad \text{if } \alpha = \frac{1}{20}
\end{aligned}
$$

$$
\begin{aligned}
f_2(x) &= \exp\left(\sin\left(\frac{\alpha}{\alpha^2+x^2}\right)\right) + \exp\left(\cos^5\left(5x^2\right)\right) - \frac{1}{2} \\
I_2 &= \int_{-1}^{10} f_2(x)\,dx = 19.155166818498490203864540596 5 \quad \text{if } \alpha = \frac{1}{30}
\end{aligned}
$$

$$
\begin{aligned}
f_3(x) &= \frac{\alpha}{\pi\left(\alpha^2+x^2\right)} \\
I_{3.1} &= \int_{-10}^{10} f_3(x)\,dx = \frac{2\arctan(200)}{\pi} \quad \text{if } \alpha = \frac{1}{20} \\
I_{3.2} &= \int_{-10}^{10} f_3(x)\,dx = \frac{2\arctan(100000)}{\pi} \quad \text{if } \alpha = \frac{1}{10000}
\end{aligned}
$$

$$
\begin{aligned}
f_4(x) &= \exp\left(\arcsin(\sin(x))\right) - 1 \\
I_4 &= \int_{-10}^{10} f_4(x)\,dx = 8.8305320509863889179186305166 9
\end{aligned}
$$

$$f_5(x) = \begin{cases} 1 & \text{if } |x| < \alpha \\ 0 & \text{otherwise} \end{cases}$$
$$I_5 = \int_{-10}^{10} f_5(x)\,dx = 1 \quad \text{if } \alpha = \frac{1}{2}$$

$$f_6(x) = \exp(\sin(x) + \cos(20x))$$
$$I_6 = \int_{0}^{2\pi} f_6(x)\,dx = 10.07146102827885793361509234466$$

$$f_7(x) = h^{\pi}_{\beta}(x)$$
$$I_7 = \int_{u-12\sqrt{2}}^{u+12\sqrt{2}} h^{\pi}_{\beta}(x)\,dx = 1.2246089728591880163577343 41 \quad \text{if } u = 0.7$$

Properties of the functions and observatons:

- $f_1, f_2, f_3, f_6, f_7 \in C^{\infty}$,
  $f_4 \in C^{0}$,
  $f_5(\cdot)$ is the rectangular pulse function that is not continuous.
- $f_7(\cdot)$ is the limit case of the $\mathcal{R}_{Q}^{\text{II},KK'}$ for $\Theta = \pi$.
- $f_3(\cdot)$ is the Lorenzian distribution.
- $f_6(\cdot)$ is a periodic function with a period of 2π.
  Moreover $f_6(0) = f_6(\pi) = f_6(2\pi) = e$
- The reference results for $f_1(\cdot), f_2(\cdot), f_4(\cdot)$, $f_6(\cdot)$, and $f_7(\cdot)$ have been calculated with the function `NIntegrate` from `Mathematica` [2] by using the variable precision floating point format, in these specific cases we have used 200 digits of precision.
- For the functions $f_3(\cdot)$ and $f_5(\cdot)$ the reference results are analytical.

C.3.3 Results

In Table C.1 we report the results related to the functions described in the preview Section C.3.2. The full descriptions of rows and columns can be found in Section: C.3.1.

[2] reference.wolfram.com/language/ref/NIntegrate.html

Test	**Algorithm**	**f-evaluations**	**relative error vs. ref**	**estimated error**
I_1	Adaptive GK SH err	10647	$7.92 \cdot 10^{-16}$	$3.37 \cdot 10^{-17}$
	Adaptive GK ABS err	33495	$6.33 \cdot 10^{-16}$	$1.22 \cdot 10^{-11}$
	Adaptive GK QP err	16905	$1.74 \cdot 10^{-15}$	$3.20 \cdot 10^{-13}$
	QUADPACK	14805	$1.27 \cdot 10^{-15}$	$8.79 \cdot 10^{-14}$
	Romberg	32769	$5.67 \cdot 10^{-14}$	n.a.
I_2	Adaptive GK SH err	24507	$2.41 \cdot 10^{-15}$	$8.50 \cdot 10^{-17}$
	Adaptive GK ABS err	35763	$2.60 \cdot 10^{-15}$	$3.82 \cdot 10^{-12}$
	Adaptive GK QP err	32319	$2.04 \cdot 10^{-15}$	$6.36 \cdot 10^{-14}$
	QUADPACK	28413	$5.56 \cdot 10^{-16}$	$2.43 \cdot 10^{-13}$
	Romberg	65537	$9.50 \cdot 10^{-12}$	n.a.
$I_{3.1}$	Adaptive GK SH err	781	$5.57 \cdot 10^{-16}$	$4.90 \cdot 10^{-16}$
	Adaptive GK ABS err	1485	$5.57 \cdot 10^{-16}$	$6.66 \cdot 10^{-13}$
	Adaptive GK QP err	1485	$5.57 \cdot 10^{-16}$	$9.97 \cdot 10^{-15}$
	QUADPACK	1155	0	$1.09 \cdot 10^{-14}$
	Romberg	16385	$8.33 \cdot 10^{-13}$	n.a.
$I_{3.2}$	Adaptive GK SH err	1573	0	$1.37 \cdot 10^{-16}$
	Adaptive GK ABS err	2365	$2.22 \cdot 10^{-16}$	$2.53 \cdot 10^{-12}$
	Adaptive GK QP err	2365	$2.22 \cdot 10^{-16}$	$5.31 \cdot 10^{-13}$
	QUADPACK	1911	$2.22 \cdot 10^{-16}$	$2.83 \cdot 10^{-13}$
	Romberg	8388609	$7.43 \cdot 10^{-13}$	n.a.
I_4	Adaptive GK SH err	9889	$2.01 \cdot 10^{-16}$	$3.37 \cdot 10^{-15}$
	Adaptive GK ABS err	6325	$4.02 \cdot 10^{-16}$	$3.32 \cdot 10^{-14}$
	Adaptive GK QP err	4895	$4.02 \cdot 10^{-16}$	$8.67 \cdot 10^{-14}$
	QUADPACK	6279	$2.01 \cdot 10^{-16}$	$2.57 \cdot 10^{-13}$
	Romberg	65537	$4.33 \cdot 10^{-10}$	n.a.
I_5	Adaptive GK SH err	2961	$4.44 \cdot 10^{-16}$	$1.75 \cdot 10^{-17}$
	Adaptive GK ABS err	1505	$1.11 \cdot 10^{-16}$	$3.54 \cdot 10^{-16}$
	Adaptive GK QP err	1505	$1.11 \cdot 10^{-16}$	$9.32 \cdot 10^{-16}$
	QUADPACK	6237	0	$1.19 \cdot 10^{-14}$
	Romberg	536870913	$3.68 \cdot 10^{-8}$	n.a.
I_6	Adaptive GK SH err	2667	$1.06 \cdot 10^{-15}$	$1.02 \cdot 10^{-18}$
	Adaptive GK ABS err	6363	$1.41 \cdot 10^{-15}$	$4.74 \cdot 10^{-14}$
	Adaptive GK QP err	2919	$5.29 \cdot 10^{-16}$	$1.25 \cdot 10^{-15}$
	QUADPACK	3087	$1.76 \cdot 10^{-16}$	$1.14 \cdot 10^{-13}$
	Romberg	3 (see Obs.: 1)	0.70	n.a.
$I_{6,u=0.7}$	Adaptive GK SH err	1331	$5.44 \cdot 10^{-16}$	$1.83 \cdot 10^{-18}$
	Adaptive GK ABS err	3883	$3.63 \cdot 10^{-16}$	$1.00 \cdot 10^{-12}$
	Adaptive GK QP err	2409	0	$1.00 \cdot 10^{-12}$
	QUADPACK	1701	$3.26 \cdot 10^{-15}$	$1.33 \cdot 10^{-14}$
	Romberg	524289	$1.97 \cdot 10^{-11}$	n.a.

Table C.1. *Results of the tests on the functions in C.3.2*

Observations

1. The Romberg algorithm has failed the quadrature This is due to the fact that the value of $f_6(\cdot)$ at the mid-point is equal to the values on the boundaries of the integration interval C.3.2. This specific implementation of the Romberg quadrature is therefore unable to handle this particular case.

2. The quadrature algorithm, when based on the QUADPACK error estimator, appears to perform better on functions that are not continuous or smooth.

3. The quadrature algorithm, when based on the sharper error estimator, appears to work better with smooth functions.

C.3.4 General conclusions regarding the tests

From table C.1 we can conclude that, regardless of the used error estimator, adaptive quadratures based on the Gauss-Kronrod scheme generally perform very well (in comparison to the Romberg algorithm). Regarding the comparison between the error estimators, we can observe that the sharper error tends to work better on smooth integrands, while the QUADPACK error appears to be more effective on non-smooth integrands.

From these tests we can conclude that the choice of Gauss-Kronrod adaptive quadrature is ideal for performing the computation of $\mathcal{R}_Q^{\mathrm{III},KK'}$ quantities (4.3) or for generating reference results.

Bibliography

Adams, T., Hummer, D., and Rybicki, G. (1971). Numerical evaluation of the redistribution function RII and of the associated scattering integral. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 11(9):1365 – 1376.

Alsina Ballester, E., Belluzzi, L., and Trujillo Bueno, J. (2016). The Magnetic Sensitivity of the Mg II k Line to the Joint Action of Hanle, Zeeman, and Magneto-optical Effects. *ApJ*, 831(2):L15.

Alsina Ballester, E., Belluzzi, L., and Trujillo Bueno, J. (2017). The Transfer of Resonance Line Polarization with Partial Frequency Redistribution in the General Hanle–Zeeman Regime. *ApJ*, 836(1):6.

Alsina Ballester, E., Belluzzi, L., and Trujillo Bueno, J. (2018). Magneto-optical Effects in the Scattering Polarization Wings of the Ca I 4227 Å Resonance Line. *ApJ*, 854(2):150.

Alsina Ballester, E., Belluzzi, L., and Trujillo Bueno, J. (2022). The transfer of polarized radiation in resonance lines with partial frequency redistribution, J-state interference, and arbitrary magnetic fields. A radiative transfer code and useful approximations. *A&A*, 664:A76.

Anusha, L. and Nagendra, K. (2011). Polarized Line Formation in Multi-dimensional Media. IV. A Fourier Decomposition Technique to Formulate the Transfer Equation with Angle-dependent Partial Frequency Redistribution. *ApJ*, 739(1):40.

Anusha, L. S. and Nagendra, K. N. (2012). Polarized Line Formation in Multi-dimensional Media. V. Effects of Angle-dependent Partial Frequency Redistribution. *ApJ*, 746(1):84.

Anusha, L. S., Nagendra, K. N., Bianda, M., Stenflo, J. O., Holzreuter, R., Sampoorna, M., Frisch, H., Ramelli, R., and Smitha, H. N. (2011). Analysis of the Forward-scattering Hanle Effect in the Ca I 4227 Å Line. *ApJ*, 737(2):95.

Bakhvalov, D., Dawson, M., Lakshmanamurthy, S., and Rotem, N. (2020). *Performance Analysis and Tuning on Modern CPUs: Squeeze the Last Bit of Performance from Your Application*. Independently Published.

Beentjes, C. H. (2015). Quadrature on a spherical surface. *Working note available on the website http://people.maths.ox.ac.uk/beentjes/Essays*.

Belluzzi, L. and Landi Degl'Innocenti, E. (2009). A spectroscopic analysis of the most polarizing atomic lines of the second solar spectrum. *A&A*, 495(2):577–586.

Belluzzi, L. and Trujillo Bueno, J. (2012). The Polarization of the Solar Mg II h and k Lines. *ApJ*, 750(1):L11.

Benedusi, P., Janett, G., Belluzzi, L., and Krause, R. (2021). Numerical solutions to linear transfer problems of polarized radiation. II. Krylov methods and matrix-free implementation. *A&A*, 655:A88.

Benedusi, P., Janett, G., Riva, S., Krause, R., and Belluzzi, L. (2022). Numerical solutions to linear transfer problems of polarized radiation. III. Parallel preconditioned Krylov solver tailored for modeling PRD effects. *A&A*, 664:A197.

Benedusi, P., Riva, S., Zulian, P., Štěpán, J., Belluzzi, L., and Krause, R. (2023). Scalable matrix-free solver for 3d transfer of polarized radiation in stellar atmospheres. *Journal of Computational Physics*, page 112013.

Berdyugina, S. V. and Fluri, D. M. (2004). Evidence for the Hanle effect in molecular lines. *A&A*, 417:775–784.

Berdyugina, S. V., Stenflo, J. O., and Gandorfer, A. (2002). Molecular line scattering and magnetic field effects: Resolution of an enigma. *A&A*, 388:1062–1078.

Bianda, M., Ramelli, R., Anusha, L. S., Stenflo, J. O., Nagendra, K. N., Holzreuter, R., Sampoorna, M., Frisch, H., and Smitha, H. N. (2011). Observations of the forward scattering Hanle effect in the Ca I 4227 Å line. *A&A*, 530:L13.

Bianda, M., Stenflo, J., Gandorfer, A., and Gisler, D. (2003). Enigmatic magnetic field effects in the scattering polarization of the ca i 4227 å line. In *Current Theoretical Models and Future High Resolution Solar Observations: Preparing for ATST*, volume 286, page 61.

Blackford, L. S., Petitet, A., Pozo, R., Remington, K., Whaley, R. C., Demmel, J., Dongarra, J., Duff, I., Hammarling, S., Henry, G., et al. (2002). An updated set of basic linear algebra subprograms (blas). *ACM Transactions on Mathematical Software*, 28(2):135–151.

Bommier, V. (1997a). Master equation theory applied to the redistribution of polarized radiation, in the weak radiation field limit. I. Zero magnetic field case. *A&A*, 328:706–725.

Bommier, V. (1997b). Master equation theory applied to the redistribution of polarized radiation, in the weak radiation field limit. II. Arbitrary magnetic field case. *A&A*, 328:726–751.

Bourquin, R. (2015-04). Exhaustive search for higher-order kronrod-patterson extensions. *Research Report, SAM-ETHZ*, 2015-11.

Capozzi, E., Alsina Ballester, E., Belluzzi, L., Bianda, M., Dhara, S. K., and Ramelli, R. (2020). Observational indications of magneto-optical effects in the scattering polarization wings of the Ca I 4227 Å line. *A&A*, 641:A63.

Carlin, E. S., Asensio Ramos, A., and Trujillo Bueno, J. (2013). Temporal Evolution of the Scattering Polarization of the Ca II IR Triplet in Hydrodynamical Models of the Solar Chromosphere. *ApJ*, 764(1):40.

Carlin, E. S. and Bianda, M. (2017). Spatiotemporal Evolution of Hanle and Zeeman Synthetic Polarization in a Chromospheric Spectral Line. *ApJ*, 843(1):64.

Carlin, E. S., Manso Sainz, R., Asensio Ramos, A., and Trujillo Bueno, J. (2012). Scattering Polarization in the Ca II Infrared Triplet with Velocity Gradients. *ApJ*, 751(1):5.

Carlsson, M., Hansteen, V. H., Gudiksen, B. V., Leenaarts, J., and De Pontieu, B. (2016). A publicly available simulation of an enhanced network region of the Sun. *A&A*, 585:A4.

Clapp, R., Dimitrov, M., Kumar, K., Viswanathan, V., and Willhalm, T. (2015). Quantifying the performance impact of memory latency and bandwidth for big data workloads. In *2015 IEEE International Symposium on Workload Characterization*, pages 213–224.

Davis, P. J. and Rabinowitz, P. (2007). *Methods of numerical integration*. Courier Corporation.

del Pino Alemán, T., Casini, R., and Manso Sainz, R. (2016). Magnetic Diagnostics of the Solar Chromosphere with the Mg II h-k Lines. *ApJ*, 830(2):L24.

del Pino Alemán, T. and Trujillo Bueno, J. (2021). The Impact of Limited Time Resolution on the Forward-scattering Polarization in the Solar Sr I 4607 Å Line. *ApJ*, 909(2):180.

del Pino Alemán, T., Trujillo Bueno, J., Casini, R., and Manso Sainz, R. (2020). The Magnetic Sensitivity of the Resonance and Subordinate Lines of Mg II in the Solar Chromosphere. *apj*, 891(1):91.

del Pino Alemán, T., Trujillo Bueno, J., Štěpán, J., and Shchukina, N. (2018). A Novel Investigation of the Small-scale Magnetic Activity of the Quiet Sun via the Hanle Effect in the Sr I 4607 Å Line. *ApJ*, 863(2):164.

Deuflhard, P. and Hohmann, A. (2003). *Numerical analysis in modern scientific computing: an introduction*. Springer.

Dhara, S. K., Capozzi, E., Gisler, D., Bianda, M., Ramelli, R., Berdyugina, S., Alsina, E., and Belluzzi, L. (2019). Observations on spatial variations of the Sr I 4607 Å scattering polarization signals at different limb distances with ZIMPOL. *A&A*, 630:A67.

Ding, N. and Williams, S. (2019). An Instruction Roofline Model for GPUs. In *2019 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, pages 7–18. IEEE.

Domke, H. and Hubeny, I. (1988). Scattering of Polarized Light in Spectral Lines with Partial Frequency Redistribution: General Redistribution Matrix. *ApJ*, 334:527.

Faddeeva, V. N. and Terent'ev, N. M. (1961). *Tables of Values of the Function W (z)*. Pergamon Press.

Faurobert, M. (1987a). Linear polarization of resonance lines in the absence of magnetic fields. I - Slabs of finite optical thickness. *A&A*, 178(1-2):269–276.

Faurobert, M. (1987b). Linear polarization of resonance lines in the absence of magnetic fields. I - Slabs of finite optical thickness. *A&A*, 178(1-2):269–276.

Faurobert, M. (1988a). Linear polarization of resonance lines in the absence of magnetic fields. II - Semi-infinite atmospheres. *A&A*, 194(1-2):268–278.

Faurobert, M. (1988b). Linear polarization of resonance lines in the absence of magnetic fields. II - Semi-infinite atmospheres. *A&A*, 194(1-2):268–278.

Faurobert-Scholl, M. (1992). Hanle effect with partial frequency redistribution. II - Linear polarization of the solar CA I 4227 A line. *A&A*, 258(2):521–534.

Fontenla, J. M., Avrett, E. H., and Loeser, R. (1993). Energy Balance in the Solar Transition Region. III. Helium Emission in Hydrostatic, Constant-Abundance Models with Diffusion. *ApJ*, 406:319.

Frisch, H. (2007). The Hanle effect. Decomposition of the Stokes parameters into irreducible components. *A&A*, 476(2):665–674.

Frisch, H. (2009). The Hanle Effect. Angle-dependent Frequency Redistribution Function. Decomposition of the Stokes Parameters in Irreducible Components. In Berdyugina, S. V., Nagendra, K. N., and Ramelli, R., editors, *Solar Polarization 5: In Honor of Jan Stenflo*, volume 405 of *Astronomical Society of the Pacific Conference Series*, page 87.

Frisch, H. (2010). Spectral line polarization with angle-dependent partial frequency redistribution. I. A Stokes parameters decomposition for Rayleigh scattering. *A&A*, 522:A41.

Gandorfer, A. (2000). *The Second Solar Spectrum: A high spectral resolution polarimetric survey of scattering polarization at the solar limb in graphical representation. Volume I: 4625 Å to 6995 Å*, volume 1. vdf Hochschulverlag AG.

Gandorfer, A. (2002). *The Second Solar Spectrum: A high spectral resolution polarimetric survey of scattering polarization at the solar limb in graphical representation. Volume II: 3910 Å to 4630 Å*, volume 2. vdf Hochschulverlag AG.

Gandorfer, A. (2005). *The Second Solar Spectrum: A high spectral resolution polarimetric survey of scattering polarization at the solar limb in graphical representation. Volume III: 3160 Å to 3915 Å*, volume 3. vdf Hochschulverlag AG.

Gonnet, P. (2012). A review of error estimation in adaptive quadrature. *ACM Comput. Surv.*, 44(4).

Goto, K. and Geijn, R. A. v. d. (2008). Anatomy of high-performance matrix multiplication. *ACM Trans. Math. Softw.*, 34(3).

Gouttebroze, P. (1986). Fast approximations for the R_{II-A} redistribution function. *A&A*, 160(1):195–198.

Grätsch, T. and Bathe, K.-J. (2005). A posteriori error estimation techniques in practical finite element analysis. *Computers & structures*, 83(4-5):235–265.

Gudiksen, B. V., Carlsson, M., Hansteen, V. H., Hayek, W., Leenaarts, J., and Martínez-Sykora, J. (2011). The stellar atmosphere simulation code bifrost-code description and validation. *A&A*, 531:A154.

Guerreiro, N., Janett, G., Riva, S., Benedusi, P., and Belluzzi, L. (2023). Modeling scattering polarization in the solar Ca I 4227 line with angle-dependent PRD effects and bulk velocities. *A&A*.

Hanle, W. (1924). Über magnetische Beeinflussung der Polarisation der Resonanzfluoreszenz. *Zeitschrift fur Physik*, 30(1):93–105.

Hashemi, B. and Trefethen, L. N. (2017). Chebfun in three dimensions. *SIAM Journal on Scientific Computing*, 39(5):C341–C363.

Hennessy, J. L. and Patterson, D. A. (2019). *Computer architecture: a quantitative approach, 6th edition*. Elsevier.

Holzreuter, R., Fluri, D. M., and Stenflo, J. O. (2005). Scattering polarization in strong chromospheric lines. I. Explanation of the triplet peak structure. *aap*, 434(2):713–724.

Hummer, D. G. (1962). Non-coherent scattering: I. The redistribution function with Doppler broadening. *MNRAS*, 125:21–37.

Janett, G., Alsina Ballester, E., Guerreiro, N., Riva, S., Belluzzi, L., del Pino Alemán, T., and Bueno, J. T. (2021a). Modeling the scattering polarization of the solar Ca I 4227 Å line with angle-dependent partial frequency redistribution. *A&A*, 655:A13.

Janett, G., Benedusi, P., Belluzzi, L., and Krause, R. (2021b). Numerical solutions to linear transfer problems of polarized radiation. I. Algebraic formulation and stationary iterative methods. *A&A*, 655:A87.

Janett, G., Carlin, E. S., Steiner, O., and Belluzzi, L. (2017). Formal Solutions for Polarized Radiative Transfer. I. The DELO Family. *ApJ*, 840(2):107.

Janett, G. and Paganini, A. (2018). Formal Solutions for Polarized Radiative Transfer. III. Stiffness and Instability. *ApJ*, 857(2):91.

Janett, G., Steiner, O., Alsina Ballester, E., Belluzzi, L., and Mishra, S. (2019). A novel fourth-order WENO interpolation technique. A possible new tool designed for radiative transfer. *A&A*, 624:A104.

Jaume Bestard, J., Trujillo Bueno, J., Štěpán, J., and del Pino Alemán, T. (2021a). The Effects of Three-dimensional Radiative Transfer on the Resonance Polarization of the Ca I 4227 Å Line. *ApJ*, 909(2):183.

Jaume Bestard, J., Štěpán, J., and Trujillo Bueno, J. (2021b). Improved near optimal angular quadratures for polarised radiative transfer in 3D MHD models. *A&A*, 645:A101.

Johnson, S. (2012). Faddeeva w function implementation. http://ab-initio.mit.edu/Faddeeva.

Kano, R., Trujillo Bueno, J., Winebarger, A., Auchère, F., Narukage, N., Ishikawa, R., Kobayashi, K., Bando, T., Katsukawa, Y., Kubo, M., Ishikawa, S., Giono, G., Hara, H., Suematsu, Y., Shimizu, T., Sakao, T., Tsuneta, S., Ichimoto, K., Goto, M., Belluzzi, L., Štěpán, J., Asensio Ramos, A., Manso Sainz, R., Champey, P., Cirtain, J., De Pontieu, B., Casini, R., and Carlsson, M. (2017). Discovery of Scattering Polarization in the Hydrogen Lyα Line of the Solar Disk Radiation. *ApJ*, 839(1):L10.

Kleint, L., Berdyugina, S. V., Shapiro, A. I., and Bianda, M. (2010). Solar turbulent magnetic fields: surprisingly homogeneous distribution during the solar minimum. *A&A*, 524:A37.

Knuth, D. E. (1968). *The Art of Computer Programming, Volume 1, Fundamental Algorithms*. Addison-Wesley Professional.

Krommer, A. and Ueberhuber, C. (1998). Computational integration (siam, philadelphia).

Kronrod, A. S. (1965). *Nodes and weights of quadrature formulas: sixteen-place tables*.

Landi Degl'Innocenti, E. and Landolfi, M. (2004). *Polarization in spectral lines*, volume 307 of *Astronomy and Space Science Library*. Springer Science & Business Media.

Landi Degl'Innocenti, E. (2007). *Fisica solare*. Springer Science & Business Media.

Lashgar, A., Baniasadi, A., and Khonsari, A. (2012). Dynamic warp resizing: Analysis and benefits in high-performance simt. In *2012 IEEE 30th International Conference on Computer Design (ICCD)*, pages 502–503. IEEE.

Laurie, D. P. (1983). Sharper error estimates in adaptive quadrature. *BIT Numerical Mathematics*, 23(2):258–261.

Laurie, D. P. (1985). Practical error estimation in numerical integration. *Journal of Computational and Applied mathematics*, 12:425–431.

Laurie, D. P. (1997). Calculation of gauss-kronrod quadrature rules. *Math. Comput.*, 66(219):1133–1145.

Lebedev, V. I. (1975). Values of the nodes and weights of ninth to seventeenth order gauss-markov quadrature formulae invariant under the octahedron group with inversion. *USSR Computational Mathematics and Mathematical Physics*, 15(1):44–51.

Leenaarts, J., Pereira, T., and Uitenbroek, H. (2012). Fast approximation of angle-dependent partial redistribution in moving atmospheres. *A&A*, 543:A109.

Manso Sainz, R., Landi Degl'Innocenti, E., and Trujillo Bueno, J. (2004). Concerning the existence of a“turbulent”magnetic field in the quiet sun. *ApJ*, 614(1):L89–L91.

Markovsky, I. (2012). *Low rank approximation: algorithms, implementation, applications*, volume 906. Springer.

McLaren, A. (1963). Optimal numerical integration on a sphere. *Mathematics of Computation*, 17(84):361–383.

Mihalas, D. (1978). *Stellar atmospheres*. W.H. Freeman.

Mohankumar, N. M. and Sen, S. (2019). On the very accurate evaluation of the voigt functions. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 224:192–196.

Moore, C. E., Minnaert, M. G. J., and Houtgast, J. (1966). *The solar spectrum 2935 A to 8770 A, Monograph. 61*. National Bureau of Standards, U.S.

Mucciarelli, A. (2011). Microturbulent velocity from stellar spectra: a comparison between different approaches. *A&A*, 528:A44.

Nagendra, K. N. (2019). Polarized Line Formation: Methods and Solutions. In Belluzzi, L., Casini, R., Romoli, M., and Trujillo Bueno, J., editors, *Solar Polariation Workshop 8*, volume 526 of *Astronomical Society of the Pacific Conference Series*, page 99.

Nagendra, K. N., Frisch, H., and Faurobert, M. (2002). Hanle effect with angle-dependent partial redistribution. *A&A*, 395:305–320.

Nagendra, K. N. and Sampoorna, M. (2011). Spectral line polarization with angle-dependent partial frequency redistribution. IV. Scattering expansion method for the Hanle effect. *A&A*, 535:A88.

Nagendra, K. N., Sowmya, K., Sampoorna, M., Stenflo, J. O., and Anusha, L. S. (2020). Importance of Angle-dependent Partial Frequency Redistribution in Hyperfine Structure Transitions Under the Incomplete Paschen-Back Effect Regime. *ApJ*, 898(1):49.

Notaris, S. E. (1993). Error bounds for Gauss-Kronrod quadrature formulae of analytic functions. *Numerische Mathematik*, 64(1):371–380.

Oeftiger, A., Aviral, A., De Maria, R., Deniau, L., Hegglin, S., Li, K., McIntosh, E., and Moneta, L. (2016). Review of CPU and GPU Faddeeva Implementations. (CERN-ACC-2016-193):WEPOY044. 4 p.

Ofenbeck, G., Steinmann, R., Caparros, V., Spampinato, D. G., and Püschel, M. (2014). Applying the roofline model. In *2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pages 76–85. IEEE.

Padoin, E. L., de Oliveira, D. A., Velho, P., and Navaux, P. O. (2012). Time-to-solution and energy-to-solution: a comparison between arm and xeon. In *2012 Third Workshop on Applications for Multi-Core Architecture*, pages 48–53. IEEE.

Paganini, A., Hashemi, B., Alsina Ballester, E., and Belluzzi, L. (2021). Fast and accurate approximation of the angle-averaged redistribution function for polarized radiation. *A&A*, 645:A4.

Patterson, T. N. (1968). The optimum addition of points to quadrature formulae. *Mathematics of Computation*, 22(104):847–856.

Piessens, R., de Doncker-Kapenga, E., Überhuber, C. W., and Kahaner, D. K. (2012). *Quadpack: a subroutine package for automatic integration*, volume 1. Springer Science & Business Media.

Poppe, G. P. and Wijers, C. M. (1990). More efficient computation of the complex error function. *ACM Transactions on Mathematical Software (TOMS)*, 16(1):38–46.

Rachmeler, L. A., Trujillo Bueno, J., McKenzie, D. E., Ishikawa, R., Auchère, F., Kobayashi, K., Kano, R., Okamoto, T. J., Bethge, C. W., Song, D., Alsina Ballester, E., Belluzzi, L., del Pino Alemán, T., Asensio Ramos, A. A., Yoshida, M., Shimizu, T., Winebarger, A., Kobelski, A. R., Vigil, G. D., De Pontieu, B., Narukage, N., Kubo, M., Sakao, T., Hara, H., Suematsu, Y., Štěpán, J., Carlsson, M., and Leenaarts, J. (2022). Quiet Sun Center to Limb Variation of the Linear Polarization Observed by CLASP2 Across the Mg II h and k Lines. *ApJ*, 936(1):67.

Ramelli, R., Balemi, S., Bianda, M., Defilippis, I., Gamma, L., Hagenbuch, S., Rogantini, M., Steiner, P., and Stenflo, J. O. (2010). ZIMPOL-3: a powerful solar polarimeter. In McLean, I. S., Ramsay, S. K., and Takami, H., editors, *Ground-based and Airborne Instrumentation for Astronomy III*, volume 7735 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 77351Y.

Rees, D. E., Murphy, G. A., and Durrant, C. J. (1989). Stokes Profile Analysis and Vector Magnetic Fields. II. Formal Numerical Solutions of the Stokes Transfer Equations. *ApJ*, 339:1093.

Rees, D. E. and Saliba, G. J. (1982). Non-LTE resonance line polarization with partial redistribution effects. *aap*, 115(1):1–7.

Riva, S., Guerreiro, N., Janett, G., Rosinelli, D., Benedusi, P., Krause, R., and Belluzzi, L. (2023). Assessment of the CRD approximation for the observer's frame R^{III} redistribution matrix. *A&A*.

Sampoorna, M. and Nagendra, K. N. (2015). Polarized Line Formation in Moving Atmospheres with Partial Frequency Redistribution and a Weak Magnetic Field. *ApJ*, 812(1):28.

Sampoorna, M., Nagendra, K. N., and Frisch, H. (2011a). Spectral line polarization with angle-dependent partial frequency redistribution. II. Accelerated lambda iteration and scattering expansion methods for the Rayleigh scattering. *A&A*, 527:A89.

Sampoorna, M., Nagendra, K. N., and Frisch, H. (2011b). Spectral line polarization with angle-dependent partial frequency redistribution. II. Accelerated lambda iteration and scattering expansion methods for the Rayleigh scattering. *A&A*, 527:A89.

Sampoorna, M., Nagendra, K. N., and Stenflo, J. O. (2017). Polarized Line Formation in Arbitrary Strength Magnetic Fields Angle-averaged and Angle-dependent Partial Frequency Redistribution. *ApJ*, 844(2):97.

Sipser, M. (1996). *Introduction to the Theory of Computation*. Cengage Learning, USA.

Solihin, Y. (2015). *Fundamentals of parallel multicore architecture*. CRC Press.

Stenflo, J. (1994). *Solar Magnetic Fields: Polarized Radiation Diagnostics*, volume 189. Springer.

Stenflo, J. O. (1982). The Hanle Effect and the Diagnostics of Turbulent Magnetic Fields in the Solar Atmosphere. *Sol. Phys.*, 80(2):209–226.

Stenflo, J. O., Baur, T. G., and Elmore, D. F. (1980). Resonance-line polarization: IV. Observations of non-magnetic line polarization and its center-to-limb variations. *A&A*, 84:60–67.

Stenflo, J. O. and Keller, C. U. (1997). The second solar spectrum. A new window for diagnostics of the Sun. *A&A*, 321:927–934.

Stenflo, J. O., Keller, C. U., and Gandorfer, A. (1998). Differential Hanle effect and the spatial variation of turbulent magnetic fields on the Sun. *A&A*, 329:319–328.

Štěpán, J. and Trujillo Bueno, J. (2013). Porta: A three-dimensional multilevel radiative transfer code for modeling the intensity and polarization of spectral lines with massively parallel computers. *A&A*, 557:A143.

Stoer, J. and Bulirsch, R. (2013). *Introduction to numerical analysis*, volume 12. Springer Science & Business Media.

Supriya, H. D., Nagendra, K. N., Sampoorna, M., and Ravindra, B. (2012). The effect of electron scattering redistribution on atomic line polarization. *MNRAS*, 425(1):527–539.

Supriya, H. D., Sampoorna, M., Nagendra, K. N., Ravindra, B., and Anusha, L. S. (2013). An efficient decomposition technique to solve angle-dependent Hanle scattering problems. *J. Quant. Spectr. Rad. Transf.*, 119:67–76.

Supriya, H. D., Smitha, H. N., Nagendra, K. N., Stenflo, J. O., Bianda, M., Ramelli, R., Ravindra, B., and Anusha, L. S. (2014). Center-to-limb Observations and Modeling of the Ca I 4227 Å Line. *ApJ*, 793(1):42.

Trujillo Bueno, J. (2001). Atomic Polarization and the Hanle Effect. In Sigwarth, M., editor, *Advanced Solar Polarimetry – Theory, Observation, and Instrumentation*, volume 236 of *Astronomical Society of the Pacific Conference Series*, page 161.

Trujillo Bueno, J. (2014). Polarized Radiation Observables for Probing the Magnetism of the Outer Solar Atmosphere. In Nagendra, K. N., Stenflo, J. O., Qu, Z. Q., and Sampoorna, M., editors, *Solar Polarization 7*, volume 489 of *Astronomical Society of the Pacific Conference Series*, page 137.

Trujillo Bueno, J., Landi Degl'Innocenti, E., and Belluzzi, L. (2017). The Physics and Diagnostic Potential of Ultraviolet Spectropolarimetry. *Space Sci. Rev.*, 210(1-4):183–226.

Trujillo Bueno, J., Landi Degl'Innocenti, E., Collados, M., Merenda, L., and Manso Sainz, R. (2002). Selective absorption processes as the origin of puzzling spectral line polarization from the Sun. *Nature*, 415(6870):403–406.

Trujillo Bueno, J., Shchukina, N., and Asensio Ramos, A. (2004). A substantial amount of hidden magnetic energy in the quiet Sun. *Nature*, 430(6997):326–329.

Trujillo Bueno, J., Štěpán, J., Belluzzi, L., Asensio Ramos, A., Manso Sainz, R., del Pino Alemán, T., Casini, R., Ishikawa, R., Kano, R., Winebarger, A., Auchère, F., Narukage, N., Kobayashi, K., Bando, T., Katsukawa, Y., Kubo, M., Ishikawa, S., Giono, G., Hara, H., Suematsu, Y., Shimizu, T., Sakao, T., Tsuneta, S., Ichimoto, K., Cirtain, J., Champey, P., De Pontieu, B., and Carlsson, M. (2018). CLASP Constraints on the Magnetization and Geometrical Complexity of the Chromosphere-Corona Transition Region. *ApJ*, 866(1):L15.

Uitenbroek, H. (2001). Multilevel radiative transfer with partial frequency redistribution. *ApJ*, 557(1):389.

Štěpán, J., Jaume Bestard, J., and Trujillo Bueno, J. (2020). Near optimal angular quadratures for polarised radiative transfer. *aap*, 636:A24.

Wells, R. J. (1999). Rapid approximation to the voigt/faddeeva function and its derivatives. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 62(1):29–48.

Williams, S., Waterman, A., and Patterson, D. (2009). Roofline: an insightful visual performance model for multicore architectures. *Communications of the ACM*, 52(4):65–76.

Zeuner, F., Belluzzi, L., Guerreiro, N., Ramelli, R., and Bianda, M. (2022). Hanle rotation signatures in Sr I 4607 Å. *A&A*, 662:A46.

Zeuner, F., Manso Sainz, R., Feller, A., van Noort, M., Solanki, S. K., Iglesias, F. A., Reardon, K., and Martínez Pillet, V. (2020). Solar Disk Center Shows Scattering Polarization in the Sr I 4607 Å Line. *ApJ*, 893(2):L44.

Zou, Z., Hughes, T., Scott, M., Miao, D., and Sauer, R. (2022). Efficient and robust quadratures for isogeometric analysis: Reduced gauss and gauss–greville rules. *Computer Methods in Applied Mechanics and Engineering*, 392:114722.