



universität  
wien

# MASTERARBEIT

Titel der Masterarbeit

„Dust Radiative Transfer in Protoplanetary Disks with  
PHOENIX/3D“

Verfasser

Christian Rab, Bakk. rer. nat.

angestrebter akademischer Grad

Master of Science (M.Sc.)

Wien, 2012

Studienkennzahl lt. Studienblatt:

A 066 861

Studienrichtung lt. Studienblatt:

Astronomie

Betreuerin / Betreuer:

Univ. Prof. Dr. Manuel Güdel



# Abstract

Disks around young low/intermediate mass stars are often called protoplanetary disks as they are considered to be the birthplaces of planets. To understand their complex structure and composition accurate radiative transfer modelling is necessary.

This thesis is part of a larger project for modelling of passive, irradiated protoplanetary disks with the stellar atmosphere code PHOENIX/3D. The PHOENIX/3D disk models will include accurate 3D radiative transfer (continuum, lines, non-LTE), gas chemistry, heating and cooling balance of the gas and the hydrostatic disk structure.

This work is limited to the 3D dust continuum radiative transfer. The aims are to test the capability of PHOENIX/3D for dust radiative transfer in protoplanetary disks, identify problems and try to solve them (if possible). For this we use the benchmark problem for calculating the temperature structure in a protoplanetary disk as defined in Pinte et al. (2009). The implementation of a method for determining the temperature structure under the assumption of radiative equilibrium is therefore also part of this work.

To apply PHOENIX/3D to the configuration of a protoplanetary disk several adaptations were necessary. PHOENIX/3D is a so called “discrete ordinate” code. The radiative transfer equation is solved along a finite number of rays with distinct directions. The drawback of this method is that small scale structure may not be resolved correctly. This is especially true for protoplanetary disks as the star ( $\approx 10^{-2}$  AU) but also the inner rim of the disk ( $\approx 10^{-1}$  AU) are small compared to the typical dimension of the disks (several 100 AU). To secure that the radiation of the star reaches all areas of the disk it was necessary to change the distribution of the rays. Because of the limited number of rays it was necessary to correct the weights for the integration of the mean intensity depending on the size of the star and the distance to the star. For the inner rim this is not possible because the spatial dimensions of the inner rim are not as well defined as for the star. To correctly resolve the inner rim it was necessary to align the grid for the rays and the spatial grid so that the distribution of the grid points is equal for both grids.

To estimate the temperature structure of a protoplanetary disk a new method for solving the radiative equilibrium equation in combination with the radiative transfer equation was implemented. This method is based on the approximate  $\Lambda$ -operator technique already used in PHOENIX/3D. The approximate  $\Lambda$ -operator is used for solving the radiative equilibrium equation via a Newton-Raphson iteration.

The correctness of the new method was verified by the results presented in the benchmark-paper. The benchmark includes four test cases with different optical depths in the mid-plane of the disk, ranging from  $\approx 10^3$  to  $\approx 10^6$  for the  $0.81 \mu\text{m}$  wavelength. The new temperature correction scheme shows good convergence properties and converges for all four test cases. For large areas of the disk the results of the new scheme are satisfactory. However, in the deep inner region of the disk, where the stellar radiation does not penetrate, the determined temperatures are always too high. The deviations depend on

the optical depth, ranging from +20%, for the lowest optical depth test case to +65% for the highest optical depth test case. The reason for this is probably numerical diffusion, caused by the too inaccurate interpolation methods used in PHOENIX/3D for the radiative transfer. To overcome this problem large spatial grids are necessary, which increases the computational needs, and therefore becomes unreasonable at least for the highest optical depth test cases. So, further improvements concerning the accuracy (e.g. different kind of interpolation method) and the performance are required.

# Zusammenfassung

Scheiben um junge Sterne im niedrigen und mittleren Massenbereich werden oft protoplanetare Scheiben genannt, da sie als die Geburtsstätte von Planeten gelten. Für das Verständnis der komplexen Struktur dieser Scheiben und deren Zusammensetzung sind aufwendige und genaue Strahlungstransportmodelle notwendig.

Diese Arbeit ist Teil eines größeren Projektes, das sich eine möglichst umfassende Modellierung von „statischen“, bestrahlten protoplanetaren Scheiben mit Hilfe des 3D Strahlungstransportcodes PHOENIX/3D zum Ziel gesetzt hat. Diese Modelle sollen die Modellierung der 3D Scheibenstruktur, relevanter Heiz- und Kühlprozesse des Gases, exakten 3D Strahlungstransport (non-LTE, Linien und Kontinuum) und chemische Netzwerke beinhalten.

Diese Arbeit beschränkt sich auf den 3D Strahlungstransport für die Staubkomponente der protoplanetaren Scheibe. Ziel ist die Anwendung von PHOENIX/3D auf ein typisches „statisches“ Staubscheibenmodell, um eventuelle Probleme für den Scheiben - Strahlungstransport zu identifizieren und soweit als möglich auch zu lösen. Für die Tests wurde der Benchmark von Pinte et al. (2009) ausgewählt. Dieser Benchmark wurde speziell für protoplanetare Staubscheiben konzipiert und beinhaltet die Ermittlung der Temperaturstruktur und die Modellierung der resultierenden spektralen Energieverteilung. Diese Arbeit konzentriert sich auf die Ermittlung der Temperaturstruktur. Dafür war es notwendig eine neue Methode zur Bestimmung der Staubtemperatur unter der Annahme von Strahlungsgleichgewicht in PHOENIX/3D zu implementieren.

Tests mit dem neu aufgesetzten Scheibenmodell zeigten, dass für korrekten Scheibenstrahlungstransport Anpassungen in PHOENIX/3D notwendig waren. PHOENIX/3D verwendet zur Lösung der Strahlungstransportgleichung die sogenannte „Methode der diskreten Ordinaten“. Die Strahlungstransportgleichung wird entlang einer endlichen Anzahl von Strahlen mit bestimmten Richtungen gelöst. Der Nachteil dieser Methode ist, dass kleinskalige Objekte eventuell nicht richtig aufgelöst werden. Im Fall der protoplanetaren Scheiben mit einer typischen Ausdehnung von mehreren 100 AU trifft dies auf den Stern ( $\approx 10^{-2}$  AU) und den inneren Rand der Scheibe ( $\approx 10^{-1}$  AU) zu. Um den Stern richtig aufzulösen, war es notwendig, die Strahlen so zu verteilen, dass garantiert ist, dass das Licht des Sterns alle Bereiche des Scheibenmodells erreichen kann. Ein weiteres Problem ist die zu starke Gewichtung der Strahlung des Sterns für die Integration der durchschnittlichen Intensität aufgrund der begrenzten Anzahl der Strahlen. Für diese Problem wurde ein Algorithmus implementiert, der die Gewichtung abhängig von der Sterngröße und der Entfernung zum Stern korrigiert. Um den inneren Bereich der Scheibe korrekt aufzulösen, war es notwendig, das Gitter für die Strahlen und das räumliche Gitter für die Scheibe aufeinander abzustimmen (gleiche Verteilung der Gitterpunkte).

Zur Bestimmung der Temperaturstruktur ist zusätzlich zur Lösung der Strahlungstransportgleichung die Lösung der Strahlungsgleichgewichtsgleichung erforderlich. Dafür wurde

ein Newton-Raphson Verfahren verwendet. Die Implementierung des Verfahrens basiert auf der Benutzung des „Approximate Lambda“-Operators, der in PHOENIX/3D für das Lösen der Strahlungstransportgleichung verwendet wird. Das garantiert eine gute Konvergenzrate und erlaubt eine nahtlose Integration des Verfahrens in die bestehende Methode zur Lösung der Strahlungstransportgleichung.

Die Korrektheit der Methode wurde anhand der Resultate aus dem Benchmarkpaper verifiziert. Das Benchmarkpaper beinhaltet vier verschiedene Testfälle mit optischen Tiefen in der Mittelebene der Scheibe von  $\approx 10^3$  bis  $\approx 10^6$  für eine Wellenlänge von  $0.81 \mu\text{m}$ . Das neue Temperaturkorrekturverfahren zeigt gute Konvergenzeigenschaften und konvergiert für alle vier Testfälle. Für den Großteil der Scheibe wurde eine zufriedenstellende Übereinstimmung der Resultate festgestellt. In den tiefen inneren Regionen der Scheibe, in die die Strahlung des Sternes nicht vordringen kann, sind die von PHOENIX/3D ermittelten Temperaturen aber zu hoch. Für den Testfall mit der geringsten optischen Tiefe liegt die Abweichung im Bereich von +20%, für den Fall mit der höchsten optischen Tiefe beträgt die Abweichung +65%. Der Grund dafür liegt wahrscheinlich in der zu geringen Genauigkeit der Interpolationsmethoden, die in PHOENIX/3D für den Strahlungstransport verwendet werden. Momentan kann die Abweichung nur mit einer größeren Anzahl von räumlichen Gitterpunkten verringert werden. Das erfordert aber einen großen Rechenaufwand. Für die praktische Anwendung des Verfahrens ist deshalb eine Verbesserung der Genauigkeit (z.B.: durch eine andere Art der Interpolation) und der Performance notwendig.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Radiative Transfer</b>	<b>7</b>
2.1	Radiative Transfer Equation . . . . .	7
2.1.1	Formal Solution . . . . .	10
2.1.2	Schwarzschild-Milne Equation . . . . .	12
2.2	Radiative Equilibrium . . . . .	13
2.3	The Overall Problem . . . . .	14
<b>3</b>	<b>3D Radiative Transfer in PHOENIX/3D</b>	<b>15</b>
3.1	Solving the Radiative Transfer Equation . . . . .	15
3.1.1	Lambda-Iteration . . . . .	16
3.1.2	Operator Splitting/Accelerated Lambda-Iteration . . . . .	17
3.2	ALI in PHOENIX/3D . . . . .	18
3.2.1	Formal Solution . . . . .	18
3.2.2	Construction of $\Lambda^*$ . . . . .	21
3.2.3	Solving the Operator Splitting Equation . . . . .	22
3.2.4	Convergence Properties and Performance . . . . .	23
<b>4</b>	<b>Disk Radiative Transfer with PHOENIX/3D</b>	<b>25</b>
4.1	Modelling of the Disk . . . . .	25
4.1.1	The Spatial Grid . . . . .	25
4.1.2	The Star . . . . .	27
4.1.3	The Disk . . . . .	27
4.2	The Resolution Problem . . . . .	29
4.2.1	Resolving the Star . . . . .	29
4.2.2	Resolving the Inner Rim . . . . .	32
<b>5</b>	<b>Temperature Correction</b>	<b>35</b>
5.1	Solving the Radiative Equilibrium Equation . . . . .	35
5.2	Implementation . . . . .	37
<b>6</b>	<b>Tests and Results</b>	<b>41</b>
6.1	Benchmark Problem . . . . .	41
6.2	Results . . . . .	42
6.2.1	Test Configuration . . . . .	42
6.2.2	Discussion . . . . .	42
6.2.3	Symmetry Test . . . . .	50
6.3	Convergence and Performance . . . . .	50
6.3.1	Convergence . . . . .	50
6.3.2	Performance . . . . .	53

<b>7 Conclusions and Outlook</b>	<b>55</b>
<b>References</b>	<b>57</b>
<b>List of Figures</b>	<b>61</b>
<b>Acknowledgements</b>	<b>63</b>
<b>Curriculum Vitae</b>	<b>65</b>

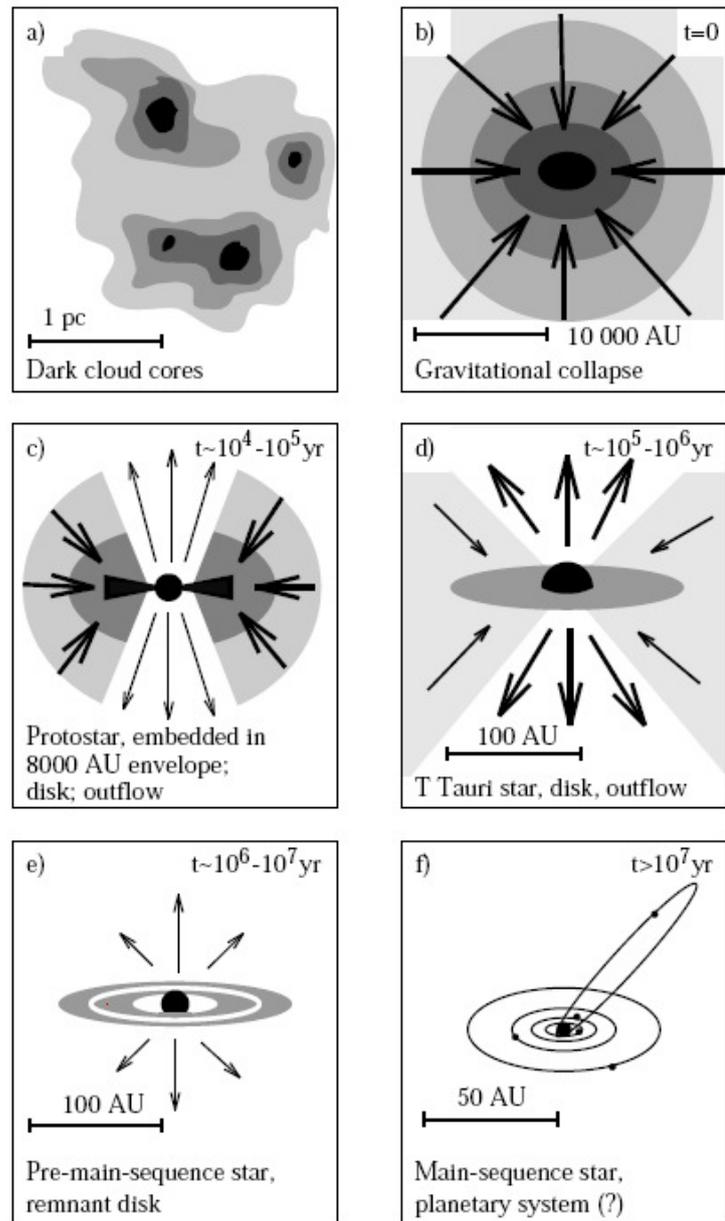
# Chapter 1

## Introduction

Circumstellar disks nowadays are commonly observed around young low mass (T Tauri) and intermediate mass (Herbig Ae/Be) stars (Watson et al. 2007). Disks are a natural outcome of the low/intermediate mass star formation process. Following Shu et al. (1987) and Hogerheijde (1998) this process may be separated into the following stages (see Fig. 1.1, panels *a* to *f*):

- a) Fragmentation of the initial molecular cloud into dark, dense and cold cloud cores.
- b) Gravitational collapse of these cores lead to the formation of a central hydrostatic structure surrounded by a large envelope from which material continues to fall onto the central structure. Caused by angular momentum conservation a disk is forming in the envelope.
- c) With enough mass accreted, deuterium will eventually ignite in the central region - the protostar. Most of these protostars will produce a stellar wind. This stellar wind in combination with the interaction of the protostar with the accretion disk and magnetic fields leads to collimated bipolar outflows and jets in the direction of the rotational pole (the direction of weakest resistance). At the end of this stage, most of the still infalling material from the envelope is now accreted by the disk.
- d) After losing nearly all its surrounding material the so-called T Tauri star becomes visible in the optical and the accretion on the disk slows down.
- e) After the mass reservoir of the envelope is exhausted accretion on the disk stops, the star continues to contract and evolves towards the main sequence. During this stage the disk disperses caused by accretion of disk material by the star, stellar winds, photo-evaporation and possible planet formation.
- f) The star has reached the main sequence and may be orbited by planets.

This evolutionary process is also visible to observations, although for stages *b* and *c* the forming protostar is not directly observable, only its envelope becomes visible in the longer wavelength range. Common to the spectral energy distribution (SED) of these young stellar objects (YSOs) is a strong infrared component (*infrared excess*) additional to radiation of the pre-stellar object. This is caused by the thermal emission of the heated envelope and/or disk (see Fig. 1.2).



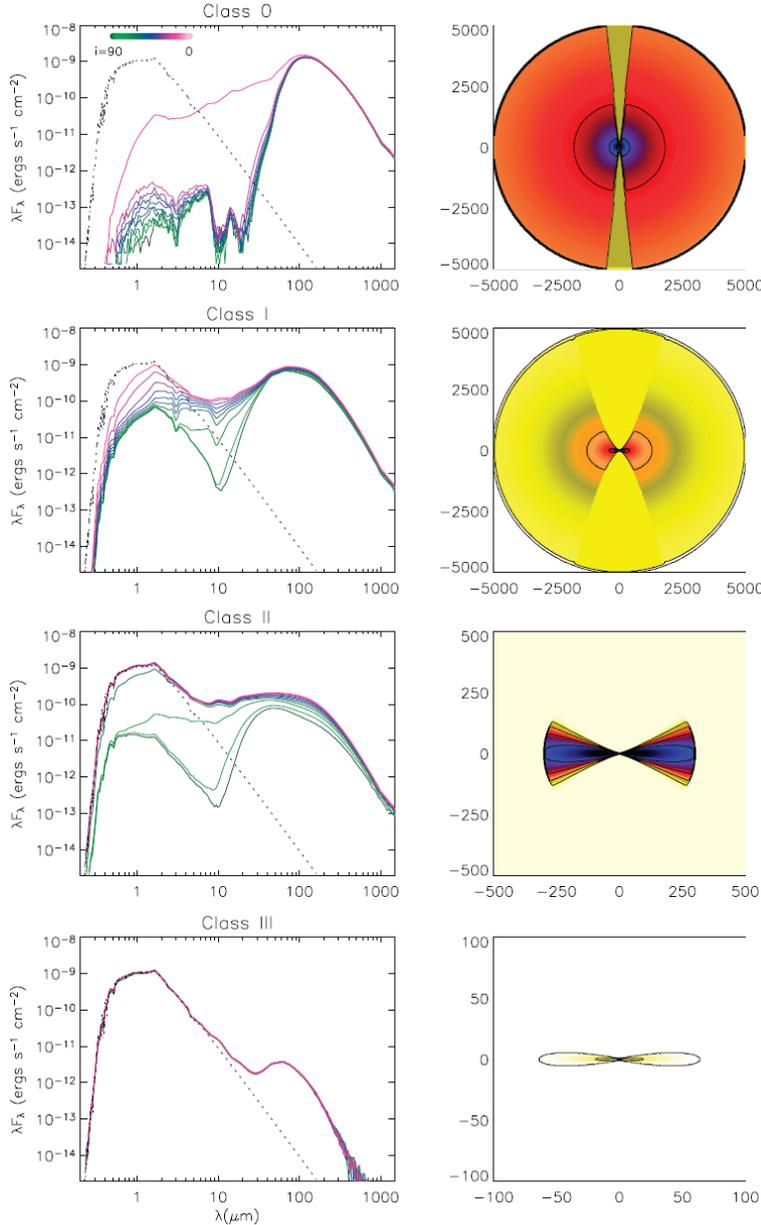
**Figure 1.1** – Overview of the different stages of low mass star formation. Also the typical spatial and time scales for each stage of the star formation process are given. Adapted from Hogerheijde (1998).

Lada (1987) established a classification scheme according to the slope of the spectral energy distribution (SED) for these YSOs. Three classes (I-II-III) were defined based on the *spectral index*  $\alpha_{IR}$  which is defined as

$$\alpha_{IR} := \frac{d \log(\lambda F_{\lambda})}{d \log \lambda} \quad (1.1)$$

where  $\lambda$  is the wavelength and  $F_{\lambda}$  is the flux in the wavelength range 2 to  $25 \mu\text{m}$ .

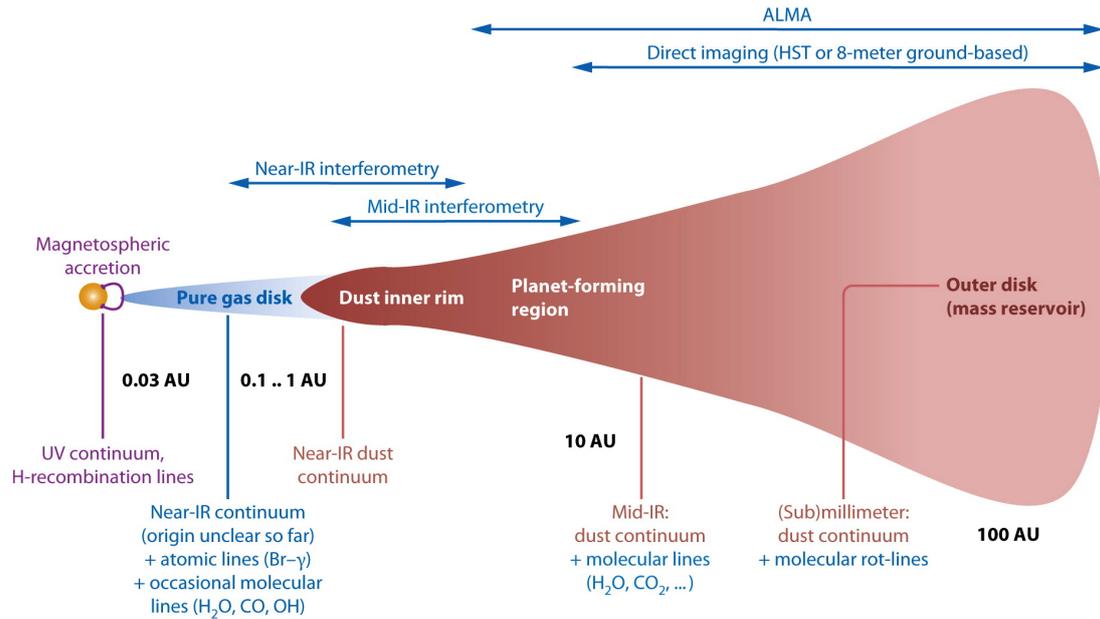
Owing to ongoing improvements in observational techniques this scheme was refined by various intermediate classes (e.g. Greene et al. 1994; Whitney et al. 2003) and André et al. (1993) introduced an additional Class 0 derived from observations in the millimetre wavelength regime.



**Figure 1.2** – Model calculations of YSOs. Models are shown for the corresponding observational classes 0,I,II,III. The *left panels* show the SED of the YSO. The *black dotted lines* show the spectra of the star. The *solid lines* show the composite SED of the star and the envelope/disk for different inclinations  $i$  from  $i = 0$  (*magenta*) to  $i = 90$  (*green*). In the *right panels* the corresponding density distribution is given. For the density a log scale is used (*white*: low density to *black*: high density). The *black solid lines* are isodensity contours. The axis shows the dimension in AU. Adapted from Whitney et al. (2003).

Although a one-to-one mapping is not possible these classes can be assigned to the stages shown in Fig. 1.1 (Hogerheijde 1998). Class 0 and I ( $\alpha_{IR} > 0.3$ ) are representing stage *c*, where for Class 0 the star is still completely hidden in the envelope (no optical or infrared emission). Stage *d* corresponds to Class II ( $-1.6 < \alpha_{IR} < -0.3$ ) where the SED for longer wavelengths is still dominated by the thermal emission from the disk. Class III ( $\alpha_{IR} < -1.6$ ) can be attributed to stage *e* where the disk becomes optically thin and the SED is already dominated by the stellar radiation. The numerical values for  $\alpha_{IR}$  are from Williams and Cieza (2011) and Greene et al. (1994).

In Fig. 1.2 model calculations of YSO and their resulting SEDs for the various observational classes are shown. This figure also shows the dependency of the SED on the inclination of the object, consequently a observational classification scheme cannot be applied directly to the real evolutionary process (see Williams and Cieza 2011; Whitney et al. 2003).



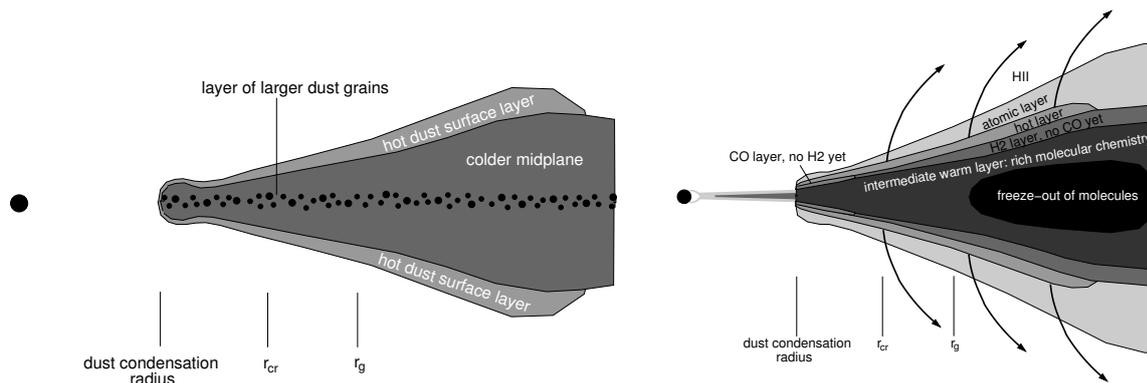
**Figure 1.3** – Sketch of a typical protoplanetary disk. The blue horizontal arrows indicate by which observational method the different regions of the disk can be spatially resolved (top of the image). On the bottom side the kind of emission arising in different regions are shown, where red refers to the dust (continuum emission) and blue to the gas component (line emission). The radial dimension is not to scale. Adapted from Dullemond and Monnier (2010).

For disks around YSOs often the term protoplanetary disks is used, as they can be seen as the “cradle of future planetary systems” (Dullemond and Monnier 2010). Fig 1.3 shows a sketch of “typical” protoplanetary disk. Although the word typical does not really fit for protoplanetary disk as they show large variances in their properties (e.g. size, mass, structure, chemistry). Here we want to give just a very short overview of some of their properties, for a more complete overview see the reviews of Williams and Cieza (2011) and Dullemond and Monnier (2010).

Outer radii of disks derived from scattered light images can be anywhere between 10 and several 1000 AU (Watson et al. 2007). Mass estimates from millimetre observations for disks around Class II YSOs yield masses between  $10^{-4}$  and  $10^{-1}M_{\odot}$  (Williams and Cieza 2011). Disks are composed of dust and gas, and although the dust dominates the opacity,  $\approx 99\%$  of the mass is in the gaseous component and only 1% in dust (this ratio may change with the evolution of the disk). Spectroscopic observations of the gaseous component reveal various atomic and molecular species implying processes like photo-ionization, photo-evaporation and a rich chemistry (Williams and Cieza 2011). They can have a flared or a flatter structure (self-shadowed) influenced by a “puffed up” inner rim (see Fig. 1.4), dust settling and growing (Dullemond and Dominik 2004).

## Modelling Protoplanetary Disks

As we have seen protoplanetary disks are quite complex structures. There exist mainly two theoretical approaches for modelling them. The first one deals with the dynamic evolution of the disk including the formation of planets. For this the disk is often treated as a flat two dimensional structure: a geometrically thin accretion disk (Pringle 1981). This approach



**Figure 1.4** – Dust and gas structure of protoplanetary disks. The *left panel* shows the dust structure with the puffed up inner rim and the hot surface layer of the disk (caused by direct irradiation). The *right panel* shows the gas structure with the various molecular and atomic layers and the pure gas disk inside the dust sublimation radius.  $r_c$  and  $r_g$  indicate typical radii for photoevaporation. Adapted from Dullemond et al. (2007).

is mainly used to study the formation and early evolution of disks where accretion is still dominant (Class I to II YSOs). To understand the evolution of a disk the main question that has to be answered is: “How does the disk lose its angular momentum”? Several mechanism like gravitational instabilities, magnetic fields, mixing and photoevaporation are identified although it is still unclear which process is the most efficient one in which state of disk evolution. For a recent review on the dynamics of protoplanetary disks see Armitage (2011).

The second approach neglects the dynamical processes responsible for the evolution of the disk but considers the vertical structure and the detailed composition of the disk. This kind of models are mainly used for the late evolutionary stages of protoplanetary disks (Class II to III YSOs) where accretion is not any longer the dominant process. The disk is treated as a passive structure in hydrostatic, thermal and chemical equilibrium where the material moves around the star on Keplerian orbits. The main questions addressed are on the detailed vertical structure and the chemical composition (dust, gas) of the disk. The main physical driver at this stage of the disk is radiation. The most important radiation source is the star but also high-energy radiation from external sources like cosmic rays or jets are important. Therefore accurate radiative transfer models are crucial for this kind modelling. Fig. 1.4 shows the typical structure and composition of a protoplanetary disk derived from these kind of models.

Mainly two kinds of numerical codes exist for this “static” modelling. The first group concentrates on the dust component of the disk as the dust emission mainly defines the shape of the SED of YSOs. This field is dominated by the so called Monte Carlo radiative transfer codes (e.g. Pinte et al. 2006; Wolf 2003; Min et al. 2009; Dullemond 2000). Monte Carlo codes use a stochastic approach to solve the radiative transfer problem. This method starts photons at the sources of the radiation field (e.g. star or dust grain) and follows them through the model space by a random walk. It is common for Monte Carlo codes to use the Stokes formalism, where the radiation field is not only described by a scalar value of the intensity, but by a Stokes vector also including information about the polarisation of the radiation. Maybe the biggest advantage of this method is that it perfectly deals with inhomogeneity (Auer 2003b). Both in the distribution of the sources (it does not

matter where the photon is emitted and how large the source is) and also in the radiation field itself (anisotropic scattering). These codes are widely used for fitting observational data like SED, scattering images and polarisation maps. This allows a detailed study of the composition of the dust (size distribution, chemical composition) and the influence of the vertical structure on the SED.

The second group of codes extend this approach to the gaseous component (e.g. Glassgold et al. 2004; Gorti and Hollenbach 2008; Woitke et al. 2009; Bruderer et al. 2012). The gas component plays an important role as it is the dominant mass component of the disk and also influences the dust dynamics (e.g. settling of the dust) and the structure of the disk (e.g. decoupling of gas and dust temperature influences the hydrostatic structure; Glassgold et al. 2004; Woitke et al. 2009). Modern infrared space telescopes like *Spitzer* and *Herschel* allow now also the direct observation of line emission of the gas. With *ALMA* (Atacama Large Millimeter Array), which makes the observation of (sub)millimeter lines possible, and in the future the *JWST* (James Webb Space Telescope), this kind of codes becomes more and more important. Although the codes differ in many details, most of them include the modelling of dust continuum radiative transfer, gas chemistry, heating and cooling balance of the gas, the hydrostatic disk structure and (approximate) non-LTE (local thermodynamical equilibrium) line transfer. This allows to study the distribution and abundance of atomic and molecular species throughout the disk and producing synthetic spectra for comparison with observational data.

This thesis is part of a larger project for modelling of protoplanetary disks in the spirit of the codes described above. The main idea is to use the existing stellar atmosphere code PHOENIX/3D (Hauschildt and Baron 2006) and especially its ability for detailed non-LTE line transfer. The overall goal of this PHOENIX/3D-based program is to develop a 3D disk model code that includes the modelling of a self-consistent 3D disk structure, all relevant heating and cooling processes, accurate 3D radiative transfer (lines, continuum) and chemical networks.

## Aims and Motivation for this Thesis

This master thesis is limited to the 3D dust continuum radiative transfer. The aims are to test the capability of PHOENIX/3D for disk radiative transfer, identify problems and try to solve them (if possible). For this we use the benchmark problem defined in Pinte et al. (2009) for calculating the temperature structure in a protoplanetary disk.

This thesis is structured as follows. We start with an introduction to radiative transfer and define the theoretical problem (chapter 2). In chapter 3 we give a brief overview of the method used in PHOENIX/3D for solving the radiative transfer equation. Chapter 4 describes the necessary adaptations to PHOENIX/3D for disk radiative transfer. In chapter 5 a new method for calculating the equilibrium dust temperature and its implementation in PHOENIX/3D is presented. Our tests and results are presented in chapter 6. Finally we present our conclusions and discuss possible improvements and future work (chapter 7).

# Chapter 2

## Radiative Transfer

In this chapter we will give a short introduction to radiative transfer and define the theoretical problem we have to deal with. In section 2.1 we discuss the radiative transfer equation and define important quantities used throughout this thesis, further we describe the conditions for radiative equilibrium and formulate the radiative equilibrium equation (section 2.2). In the last section of this chapter (section 2.3) we combine these two equations to define our overall problem.

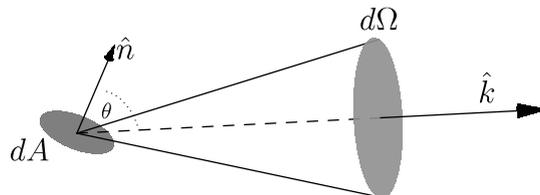
This chapter mainly follows the books of Shu (1991, chap. 1,3) and Mihalas (1978, chap. 1,2,6).

### 2.1 Radiative Transfer Equation

Radiative transfer is about the interaction of radiation with matter. During this interaction energy can be removed from (e.g. by absorption) or added to (e.g. thermal emission) the radiation field.

The so called radiative transfer equation (RTE) describes these processes. To formulate the radiative transfer equation the following definitions are needed:

#### Specific Intensity



**Figure 2.1** – Definition of the specific intensity  $I_\lambda(\vec{r}, \hat{k}, t)$ .

The *specific intensity*  $I_\lambda(\vec{r}, \hat{k}, t)$  describes the radiation field at a distinct wavelength  $\lambda$  and is defined as

$$dE = I_\lambda(\vec{r}, \hat{k}, t) dA \hat{k} \cdot \hat{n} d\Omega d\lambda dt . \quad (2.1)$$

Here  $dE$  is the amount of energy in the wavelength interval  $[\lambda, \lambda + d\lambda]$  which crosses the area  $dA$ , with unit normal  $\hat{n}$ , in the direction  $\hat{k}$  into the solid angle  $d\Omega$  in time  $dt$ .  $I$  has the units  $\text{erg cm}^{-3} \text{ s}^{-1} \text{ sr}^{-1}$ . Instead of  $\hat{k} \cdot \hat{n}$  often  $\cos \theta$  is used where  $\theta$  is the angle between  $\hat{k}$  and  $\hat{n}$  (see Fig. 2.1).

This definition has the advantage that  $I$  is a conserved quantity as long as there is no interaction with matter (for a proof see Shu 1991, chap. 1). For this case the radiative transfer equation has the simple form of

$$\frac{1}{c} \frac{\partial I_\lambda(\vec{r}, \hat{k}, t)}{\partial t} + \hat{k} \cdot \nabla I_\lambda(\vec{r}, \hat{k}, t) = 0 . \quad (2.2)$$

But usually radiation is absorbed and emitted by matter. One way to describe these processes are the macroscopic quantities called the extinction and emission coefficient.

### Extinction Coefficient

The *extinction coefficient* (or total *absorption coefficient*) describes the removal of energy from the radiation field by matter and is defined as

$$dE = \chi_\lambda(\vec{r}, t) I_\lambda(\vec{r}, \hat{k}, t) dS ds d\Omega d\lambda dt . \quad (2.3)$$

$dE$  is the amount of energy which is removed by a matter element with cross section  $dS$  and length  $ds$ , from the beam normal to  $dS$ , with intensity  $I_\lambda(\vec{r}, \hat{k}, t)$  and wavelength range  $d\lambda$  propagating into  $d\Omega$  in time  $dt$ . This definition is only valid for static media. For moving media  $\chi_\lambda$  has an angular dependence caused by the Doppler shift, but here we only deal with static media and therefore the extinction coefficient is isotropic. The quantity  $1/\chi_\lambda$  is also called the *photon mean-free-path*. It is a measure for the distance a photon can travel before it is removed from the beam by interaction with matter.

The extinction coefficient is a product either of the number density  $n(\vec{r}, t)$  or the mass density  $\rho(\vec{r}, t)$  of the matter with a cross section or cross section per unit mass. We define here  $\chi_\lambda$  as

$$\chi(\vec{r}, t)_\lambda := \rho(\vec{r}, t) \kappa_\lambda^{ext}(\vec{r}, t) . \quad (2.4)$$

With this definition  $\kappa_\lambda^{ext}$  has the unit of a cross-section per unit mass [ $\text{cm}^2 \text{ g}^{-1}$ ]. From now on we call this quantity the *opacity*.

The extinction (the removal of the photon) can happen in two different ways. The photon can be absorbed (true absorption) or scattered. Scattering is the changing of the direction of the photon. The photon is removed from the beam with direction  $\hat{k}$  and propagates into the new direction  $\hat{k}'$ . Extinction is the sum of these two processes (assuming that the processes occur independently).

$$\kappa_\lambda^{ext}(\vec{r}, t) = \kappa_\lambda^{abs}(\vec{r}, t) + \kappa_\lambda^{sca}(\vec{r}, t) . \quad (2.5)$$

The opacity depends on the properties of the matter (e.g. gas, dust), but here we do not go into detail concerning the determination of the opacity. We just assume that the quantity is known and that the coefficient is also corrected for stimulated emission (for more details see Mihalas 1978, chap. 2).

## Emission Coefficient

The macroscopic *emission coefficient* describes the radiation emitted by the media and is defined as

$$dE = \eta_\lambda(\vec{r}, \hat{k}, t) dS ds d\Omega d\lambda dt . \quad (2.6)$$

$dE$  is the energy released from the material with cross-section  $dS$  and length  $ds$ , into a solid angle  $d\Omega$  in the wavelength range  $\lambda$  in direction  $\hat{k}$  in time  $dt$ . It has the dimension  $\text{erg cm}^{-4} \text{s}^{-1} \text{sr}^{-1}$ . The emission coefficient normally has an angular dependence (direction of the ray) caused by scattering and moving media. Here we only consider isotropic scattering and static media, and therefore the angular dependency drops.

In the case of thermodynamic equilibrium a relation between the absorption coefficient  $\kappa^{abs}$  and the emission coefficient applies (*Kirchhoff's law*). This means that the amount of energy absorbed by the material has to be equal to the amount of energy emitted:

$$\eta_\lambda^t = \rho \kappa_\lambda^{abs} I_\lambda(\hat{k}) . \quad (2.7)$$

The intensity of a matter element in thermodynamic equilibrium with temperature  $T$  is given by the *Planck function*  $B_\lambda(T)$  and one can write

$$\eta_\lambda^t = \rho \kappa_\lambda^{abs} B_\lambda(T) . \quad (2.8)$$

This relation is only valid for a whole system in thermodynamic equilibrium. For a protoplanetary disk this is normally not the case (for disks the temperature depends on the location). But the concept of the so called *local thermodynamic equilibrium (LTE)* can be applied. This concept assumes that the thermodynamic properties of the material (e.g. opacity, emissivity) are the same as their thermodynamic equilibrium properties at the local values of  $T$  and  $\rho$  at each point throughout the whole model space (e.g. the disk). This approximation implies that the properties of the matter are only subject to small changes over the mean free path of a photon. If the photon can only travel short distances (e.g. high density/absorption) then the concept of LTE is a good approximation, as collisional processes are dominant in comparison to radiative processes. For a thin medium where collisional interactions are rare and the photon is allowed to travel large distances until an absorption or scattering event occurs, the approximation of LTE is normally not valid. However, here we only consider cases where LTE can be applied. In that case equation 2.8 is also valid locally and one can write

$$\eta_\lambda^t(\vec{r}, t) = \rho(\vec{r}, t) \kappa_\lambda^{abs}(\vec{r}, t) B_\lambda(T(\vec{r}, t)) . \quad (2.9)$$

With the definition of the opacity and the emissivity we now can add source and sink terms to equation 2.2. Under the assumption of static media, LTE, isotropic scattering and the definitions  $I_\lambda := I_\lambda(\vec{r}, \hat{k}, t)$ ,  $\rho := \rho(\vec{r}, t)$ ,  $\kappa_\lambda^x := \kappa_\lambda^x(\vec{r}, t)$  and  $B_\lambda := B_\lambda(T(\vec{r}, t))$  the radiative transfer equation (RTE) can be written as

$$\frac{1}{c} \frac{\partial I_\lambda}{\partial t} + \hat{k} \cdot \nabla I_\lambda = \underbrace{-\rho \kappa_\lambda^{ext} I_\lambda}_{\text{extinction}} + \underbrace{\rho \kappa_\lambda^{abs} B_\lambda + \rho \kappa_\lambda^{sca} \frac{1}{4\pi} \oint_{4\pi} I_\lambda(\vec{r}, \hat{k}', t) d\Omega'}_{\text{emission}} \quad (2.10)$$

The first term (extinction) at the right-hand side of the equation represents the amount of light removed from the considered beam. The emission term represents the amount of light added to the beam. The first term corresponds to the thermal emission of the matter (equation 2.9). The integral term represents the amount of light scattered into the beam with direction  $\hat{k}$  from all directions  $\hat{k}'$ .

### 2.1.1 Formal Solution

From now on we only deal with the time independent form of equation 2.10. To make equation 2.10 more readable some more definitions are needed.

#### Mean Intensity

One common and useful quantity in radiative transfer is the so called *mean intensity*

$$J_\lambda(\vec{r}) := \frac{1}{4\pi} \oint I_\lambda(\vec{r}, \hat{k}) d\Omega = \frac{1}{4\pi} \oint I_\lambda(\vec{r}, \theta, \phi) \sin\theta d\theta d\phi. \quad (2.11)$$

This is the average of the specific intensity over all solid angles. For the directions of the radiation often polar and azimuthal coordinates  $(\theta, \phi)$  are used.

#### Source Function

Further we define the *source function* for the isotropic case. The source function is the ratio of the total emissivity to total extinction

$$S_\lambda(\vec{r}) := \frac{\eta_\lambda(\vec{r})}{\chi_\lambda(\vec{r})}. \quad (2.12)$$

In the case for isotropic scattering the source function does not depend on the beam direction. The detailed form of the source function depends on the problem (e.g. assumption of LTE, scattering). For our case the source function has the following form

$$S_\lambda = \frac{\kappa_\lambda^{abs} B_\lambda + \kappa_\lambda^{sca} J_\lambda}{\kappa_\lambda^{ext}}. \quad (2.13)$$

Where just the corresponding emission and extinction terms of equation 2.10, and the definition for  $J_\lambda$  (equation 2.11) are used. To bring the source function into the form as it is used in PHOENIX/3D we define the *thermal coupling parameter* (Mihalas 1978, chap. 6)

$$\epsilon_\lambda := \frac{\kappa_\lambda^{abs}}{\kappa_\lambda^{abs} + \kappa_\lambda^{sca}}. \quad (2.14)$$

In the absence of scattering  $\epsilon = 1$ , for stronger scattering  $\epsilon \rightarrow 0$ . In the context of dust radiative transfer, instead of epsilon often the so called *albedo* is used, which is just given by  $1 - \epsilon$ . If we now put this into equation 2.13 (hint: multiply with  $\epsilon/\epsilon$ ) we get

$$S_\lambda = \epsilon_\lambda B_\lambda + (1 - \epsilon_\lambda) J_\lambda. \quad (2.15)$$

By replacing the ray-path derivative  $\hat{k}\nabla$  in equation 2.10 with  $d/ds$ , which is the derivative along the propagation path of the ray (we assume an orthogonal coordinate system and that the path has always a linear slope) and by using the definition of the source function (equation 2.12) for the time independent form of equation 2.10 we get

$$\frac{dI_\lambda}{ds} \frac{1}{\rho\kappa_\lambda^{ext}} = S_\lambda - I_\lambda . \quad (2.16)$$

In the next step we define the *optical depth*

$$\tau_\lambda(s) := \int_{s_0}^s -\rho\kappa_\lambda^{ext} ds = \int_{s_0}^s -\chi_\lambda ds . \quad (2.17)$$

With the minus sign in the definition, the optical depth provides a measure of how deep an outside observer can look into an observed object (e.g. from the border of an atmosphere to the center of the star).

With the definition of  $\tau_\lambda$  we obtain the so-called *standard form* of the radiative transfer equation

$$\frac{dI_\lambda(\hat{k}, \tau_\lambda)}{d\tau_\lambda} = I_\lambda(\hat{k}, \tau_\lambda) - S_\lambda(\tau_\lambda) . \quad (2.18)$$

In this form the radiative transfer equation is a linear differential equation with constant coefficients, and therefore an integrating factor exists. With the integrating factor  $e^{-\tau_\lambda}$  we get

$$\frac{d}{d\tau} (I_\lambda(\hat{k}, \tau_\lambda) e^{-\tau_\lambda}) = S_\lambda(\tau_\lambda) e^{-\tau_\lambda} , \quad (2.19)$$

and by integrating, using the boundary condition  $I(\tau = 0)$ , we get the *formal solution*

$$I_\lambda(\hat{k}, \tau_\lambda) = I_\lambda(\hat{k}, 0)e^{-\tau_\lambda} + \int_0^{\tau_\lambda} S_\lambda(\tau'_\lambda) e^{-\tau'_\lambda} d\tau'_\lambda . \quad (2.20)$$

This formal solution gives us now the possibility to get the value of  $I_\lambda$  on each position  $\tau_\lambda$  along a ray with direction  $\hat{k}$  as long as the value of  $S_\lambda$  is known. However, this is normally not the case. In our case the source function depends on  $J_\lambda$ . Which itself depends on the intensity  $I_\lambda$ . So, already for a quite simple source function this equation cannot be solved analytically. Nevertheless the *formal solution* is very important for numerical radiative transfer especially for iterative methods (see section 3.2.1).

For numerical methods it is more convenient not to use  $\tau_\lambda$  as the integration variable for the formal solution but rather use the the spatial position  $\vec{r}$  for integration. This is especially true if we are interested in more than one wavelength  $\lambda$  and more than one direction  $\hat{k}$ . As  $\tau_\lambda$  depends on  $\lambda$  and  $\hat{k}$  it would give us a different position in the model space for each wavelength and direction, which is cumbersome for a discretised model space. By using the spatial position  $\vec{r}$  as the integration variable we can write the formal solution in the following way (Shu 1991, chap. 3):

$$I_\lambda(\hat{k}, \vec{r}) = I_\lambda(\hat{k}, \vec{r}_0) \exp[-\tau_\lambda(\hat{k}, \vec{r}, \vec{r}_0)] + \int_{s_0}^s S_\lambda(\vec{r}') \exp[-\tau_\lambda(\hat{k}, \vec{r}, \vec{r}')] \chi_\lambda(\vec{r}') ds' , \quad (2.21)$$

where  $\tau_\lambda(\hat{k}, \vec{r}, \vec{r}')$  is the optical depth along a ray with direction  $\hat{k}$  from position  $\vec{r}'$  to  $\vec{r}$ .  $\vec{r}_0$  represents the starting point of the ray.

### 2.1.2 Schwarzschild-Milne Equation

The last part of the radiative transfer introduction is about the *Schwarzschild-Milne* equation. This equation is quite important, as it shows us how to get  $J$  by using the formal solution, and is very helpful to better understand the complexity of numerical radiative transfer. This section follows mainly the description of Rutten (2003, chap. 4).

For simplicity we consider the one-dimensional case (e.g. one-dimensional atmosphere). We then can write the RTE in a slightly different notation. In the one-dimensional case the path derivative in the direction of  $z$  has the form

$$(\hat{k} \cdot \nabla) I_\lambda = \frac{dz}{ds} \frac{dI_\lambda}{dz} = \mu \frac{dI_\lambda}{dz} \quad (2.22)$$

where  $\mu = \cos \theta = (dz/ds) = k_z$ . The RTE then has the form

$$\mu \frac{dI_\lambda}{d\tau_\lambda} = I_\lambda - S_\lambda. \quad (2.23)$$

We now split the formal solution into two parts, one for the outgoing radiation  $I_\lambda^+$  ( $\mu \geq 0$ ) and one for the incoming radiation  $I_\lambda^-$  ( $\mu \leq 0$ ):

$$\begin{aligned} I_\lambda^+(\tau_\lambda, \mu) &= \int_{\tau_\lambda}^{\infty} S(\tau'_\lambda) e^{-(\tau'_\lambda - \tau_\lambda)/\mu} d\tau'_\lambda / \mu \\ I_\lambda^-(\tau_\lambda, \mu) &= \int_0^{\tau_\lambda} S(\tau'_\lambda) e^{-(\tau'_\lambda - \tau_\lambda)/\mu} d\tau'_\lambda / |\mu|. \end{aligned} \quad (2.24)$$

These expressions can now be integrated over solid angle ( $\mu$ ) to get the *Schwarzschild-Milne equation* for  $J$ :

$$\begin{aligned} J_\lambda(\tau_\lambda) &= \frac{1}{2} \int_{-1}^1 I_\lambda(\tau_\lambda, \mu) d\mu \\ &= \frac{1}{2} \int_{\tau_\lambda}^{\infty} S_\lambda(\tau'_\lambda) E_1(\tau'_\lambda - \tau_\lambda) d\tau'_\lambda + \frac{1}{2} \int_0^{\tau_\lambda} S_\lambda(\tau'_\lambda) E_1(\tau_\lambda - \tau'_\lambda) d\tau'_\lambda \\ &= \frac{1}{2} \int_0^{\infty} S_\lambda(\tau'_\lambda) E_1(|\tau'_\lambda - \tau_\lambda|) d\tau'_\lambda \end{aligned} \quad (2.25)$$

where  $E_1$  is the so called *exponential integral*

$$E_1 := \int_0^1 e^{-x/\mu} \frac{d\mu}{\mu}. \quad (2.26)$$

This integral is the result of the integration of  $I$  over  $\mu$  by using equation 2.24. For  $x \gg 1$

the integral can be approximated by

$$E_1(x) \approx \frac{e^{-x}}{x} \quad \text{for } x \gg 1. \quad (2.27)$$

The *Schwarzschild-Milne equation* can also be written in a more formal way by using the so called  $\Lambda$  operator

$$J_\lambda(\tau_\lambda) = \Lambda[S_\lambda(\tau'_\lambda)] := \frac{1}{2} \int_0^\infty S_\lambda(\tau'_\lambda) E_1(|\tau'_\lambda - \tau_\lambda|) d\tau_\lambda. \quad (2.28)$$

The  $\Lambda$ -operator produces  $J_\lambda$  from  $S_\lambda$  through the given integral expression. In a more general way the  $\Lambda$ -operator just stands for any “procedure” that gives us  $J_\lambda$  from  $S_\lambda$ . We will see this in chapter 3.

## 2.2 Radiative Equilibrium

Until now we have assumed that we know the thermal part  $B_\lambda$  of the source function.  $B_\lambda$  depends on the temperature of the matter and as one can imagine the temperature depends on the radiation field, because the medium is heated by absorbed radiation (e.g. radiation from a star). This makes solving the RTE even more complicated. It can even become worse if the opacity also depends on the temperature. But for the dust in protoplanetary disks this dependency can normally be neglected.

However, for radiation dominated protoplanetary disks (irradiated disks) the temperature in the disk is determined by the radiation field. If we assume LTE we know that all absorbed energy has to be re-emitted and therefore the material in the disk has to reach an equilibrium condition for a distinct local temperature - the so called *radiative equilibrium*.

Under the assumptions of LTE for the material and that radiation is the only source of energy, the condition of radiative equilibrium can be formulated the following way:

$$\int_0^\infty \kappa_\lambda^{abs}(\vec{r}) B_\lambda(T(\vec{r})) d\lambda = \int_0^\infty \kappa_\lambda^{abs}(\vec{r}) J_\lambda(\vec{r}) d\lambda. \quad (2.29)$$

This radiative equilibrium equation (REE) just tells us that all energy absorbed from the radiation field (the right hand side) has to be re-emitted by the dust via thermal emission (the left hand side). For the gas in the disk this equation is only valid in the dense regions where the gas and dust temperature remain coupled caused by high collision rates. But for the upper layers of the disk this is not true anymore, there the dust and gas temperature decouple and various heating and cooling processes (e.g. line cooling) have to be considered for the gaseous component (e.g. Woitke et al. 2009). However, for this work we only deal with dust.

## 2.3 The Overall Problem

To solve the REE (2.29) for the temperature  $T$  we have to know the value of  $J_\lambda$ . If we use the formal solution 2.21 and insert it in the definition for  $J_\lambda$  (equation 2.11) we get

$$J_\lambda(\vec{r}) = \frac{1}{4\pi} \oint I_\lambda(\vec{k}, \vec{r}_0) \exp[-\tau_\lambda(\hat{k}, \vec{r}, \vec{r}_0)] d\Omega + \frac{1}{4\pi} \oint d\Omega \int_{s_0}^s S_\lambda(\vec{r}) \exp[-\tau_\lambda(\hat{k}, \vec{r}, \vec{r}')] \chi_\lambda(\vec{r}') ds'. \quad (2.30)$$

By considering the following assumptions

- LTE,
- radiation is the only source of energy, and
- the density distribution and the opacities do not depend on  $T$

the problem of determining the temperature structure of a protoplanetary dust disk is completely defined by the RTE (2.30) and the REE (2.29).

These are quite strict assumptions. To make a more realistic model of a dust disk also the density distribution of the disk has to be taken into account, because the temperature of the dust will influence the hydrostatic structure of the disk. To solve this, an iterative process for determining the density would be needed. Further, these assumptions also imply that the properties of the dust (e.g. opacity), except the temperature, are not changing in time and no dust is destroyed or created (this is important to determine the location of the inner dust rim).

However, with initial conditions for the star (its radiation field), the density distribution and the opacities for the dust, the temperature structure of the disk can be self-consistently calculated by solving the radiative transfer and the radiative equilibrium equation. How this can be done by using PHOENIX/3D is the content of the next chapters.

# Chapter 3

## 3D Radiative Transfer in PHOENIX/3D

PHOENIX/3D belongs to the group of the so-called discrete ordinate codes. These codes solve the radiative transfer equation along distinct rays (characteristics) with different directions by using the formal solution (ray tracing). These methods were especially successful for one dimensional problems like stellar atmospheres and are also mainly used for line transfer. With PHOENIX/3D this method was extended to 3D and can therefore also be used for protoplanetary disks.

The biggest disadvantage, in comparison to Monte Carlo codes, of this method is probably the resolution problem (see Auer 2003b). In an extended model space like a protoplanetary disk with a typical dimension of several 100 AU, the star in the center is very small ( $\approx 10^{-2}$  AU) and therefore it is quite hard to resolve the star. It must be secured that the radiation from the star reaches every point (or cell) in the disk model, therefore a very large number of rays is necessary. In Auer (2003b) this problem is called the “on/off”-problem. But we will discuss this in more detail later (see chapter 4).

An overview of the PHOENIX/3D framework including line transfer, velocity fields, homologous flows, periodic boundary conditions and example applications can be found in a paper series starting with the paper of Hauschildt and Baron (2006).

This chapter is mainly based on the paper of Hauschildt and Baron (2006) and gives an overview of the current method implemented in PHOENIX/3D for continuum radiative transfer in static media.

### 3.1 Solving the Radiative Transfer Equation

As we see from the REE (equation 2.29) we are mainly interested in the mean intensity  $J$ . To get the mean intensity we have to integrate the formal solution (see equation 2.20) over all solid angles (corresponding directions).

As we have seen in section 2.1.2 this kind of integral can be written in operator notation

$$J_\lambda = \Lambda_\lambda[S_\lambda] = \epsilon_\lambda \Lambda_\lambda[B_\lambda] + (1 - \epsilon_\lambda) \Lambda_\lambda[J_\lambda] \quad (3.1)$$

This notation is motivated by the linearity of the equation. The  $\Lambda$ -operator can be seen as a method or procedure which gives us  $J_\lambda$  if it is used on  $S_\lambda$ . But if we want to know  $J_\lambda$  at some distinct position in the model space we have to know  $S_\lambda$  at *all* positions. For the one dimensional case the values of  $S_\lambda$  can be represented by a vector and the  $\Lambda$ -operator by a matrix. The values of  $J_\lambda$  are then obtained by a simple matrix multiplication. The values of the matrix can be calculated with the help of the formal solution (see section 3.2.2).

Now everything looks quite simple but as we know from section 2.1.1 the source function itself depends on  $J_\lambda$  (at least if scattering is considered). To get a self-consistent solution an iterative scheme is needed.

### 3.1.1 Lambda-Iteration

The simplest possible iteration scheme is the so called  $\Lambda$ -iteration:

$$\begin{aligned} J_{new} &= \Lambda[S_{old}] \\ S_{new} &= (1 - \epsilon)J_{new} + \epsilon B . \end{aligned} \quad (3.2)$$

By starting with a more or less arbitrary value for  $S_{old}$  we get a value for  $J_{new}$ , which is then used to get an improved value for  $S$  ( $S_{new}$ ) for the next iteration. But as it is often the case with iterative schemes, they do not always converge. The  $\Lambda$ -Iteration especially fails in the case of high optical depths and strong scattering. Even worse, there is a physical reason for this. Here I will give a short description for the physical reason, the complete description can be found in Mihalas (1978, chap. 6).

We assume as a starting value for the iteration  $S = B$ . In the case of high optical depths (high densities) the medium is thermalized, which means  $J \approx B$ . Imagine now some kind of atmosphere with lower density at the border and higher density in the center. What we want to calculate is the deviation of  $J$  from  $B$ , as not the whole atmosphere will be thermalized. For strong scattering the thermal coupling parameter  $\epsilon \ll 1$ .  $\epsilon$  can be seen as the probability that a photon is destroyed per scattering event (converted into thermal energy). In the worst case the photon is only destroyed (thermalized) after  $n = 1/\epsilon$  iterations. But the photon can only travel a mean free path  $1/\chi$  until the next interaction (scattering). The mean free path is typically  $\Delta\tau \approx 1$ . This implies that information during one iteration can only be transported over  $\Delta\tau \approx 1$ . This is because the exponential integral (equation 2.26) falls off as  $e^{-\Delta\tau}/\Delta\tau$  for  $\Delta\tau \gg 1$ .

The optical depth at which a photon from the border is thermalized is at  $\tau \gtrsim \epsilon^{-0.5}$ . So the optical thickness a photon will pass until destruction is given by  $n^{0.5}\Delta\tau = \Delta\tau\epsilon^{-0.5} \approx \epsilon^{-0.5}$  for  $\Delta\tau \approx 1$ . So, the number of iterations needed until information of a photon reaches the thermalization depth (which can be very high) is  $\epsilon^{-0.5}$ . For  $\epsilon \ll 1$  this is unreasonable. Therefore ordinary  $\Lambda$ -iteration methods do not converge very well, they tend to stabilize at a probably wrong solution.

The convergence of an iteration scheme can also be tested by looking at the eigenvalues of the amplification matrix, in our case the  $\Lambda$ -operator. The largest eigenvalues of this kind of matrix are  $\lambda_{max} = (1 - \epsilon)/(1 - T^{-1})$  (Hauschildt and Baron 2006; Mihalas et al. 1975), where  $T$  is the optical thickness of the medium. For large  $T$  and small  $\epsilon$ ,  $\lambda_{max}$  is

nearly unity, and therefore the changes from one iteration to the next are very small. So the goal is to minimize the eigenvalues of the  $\Lambda$ -Operator. One possibility for this is the so-called operator splitting method.

### 3.1.2 Operator Splitting/Accelerated Lambda-Iteration

In this section I will describe the operator splitting (OS) technique which is used in PHOENIX/3D. Broadly speaking, operator splitting means to replace the  $\Lambda$ -operator with a different or approximate operator which accelerates the convergence by reducing the maximum eigenvalues of the operator. Therefore this method is also called *Accelerated  $\Lambda$ -Iteration* (ALI). The OS method used in PHOENIX/3D follows the method of Hamann (1987). In Hamann (1987) the approximate  $\Lambda$ -operator,  $\Lambda^*$  is defined as

$$J = \Lambda[S] = \Lambda^*[S] + (\Lambda - \Lambda^*)[S] . \quad (3.3)$$

The  $\Lambda$ -iteration scheme 3.2 can now be changed to

$$J_{new} - \Lambda^*[S_{new}] = (\Lambda - \Lambda^*)[S_{old}] . \quad (3.4)$$

The equality in equation 3.4 only holds if  $S_{old} = S_{new}$  which is the case if the iteration scheme converges. The advantage of this approach is that we now have also information about  $S_{new}$  included in the calculation of  $J_{new}$ , although neither  $S_{old}$  or  $S_{new}$  are known at the beginning. To eliminate  $S_{new}$  we use equation 2.15

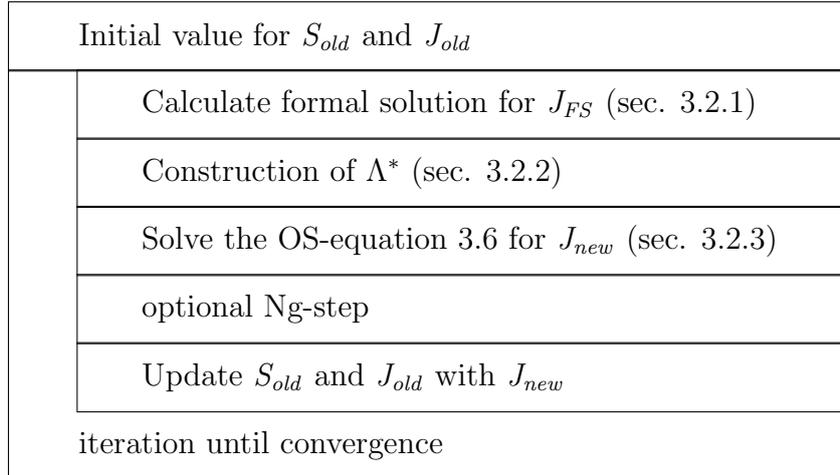
$$J_{new} - \Lambda^*[(1 - \epsilon)J_{new} + \epsilon B] = \Lambda[S_{old}] - \Lambda^*[(1 - \epsilon)J_{old} + \epsilon B] . \quad (3.5)$$

$\Lambda[S_{old}]$  is replaced by  $J_{FS}$  because if we know  $S_{old}$ ,  $J_{FS}$  can be calculated via a formal solution. Further,  $\epsilon B$  cancels out and  $J_{new}$  factors out in the left-hand side of the equation, yielding

$$J_{new} = \{1 - \Lambda^*(1 - \epsilon)\}^{-1} \{J_{FS} - \Lambda^*[(1 - \epsilon)J_{old}]\} . \quad (3.6)$$

With initial values for  $S_{old}$  and  $J_{old}$  (both can have the same value) this equation can be solved to get  $J_{new}$ . We still do not know how this  $\Lambda^*$  looks like, but we already can define some constraints for it. As already mentioned it should minimize the eigenvalues for the  $\Lambda$ -iteration. Further it should be easy to invert the  $\Lambda^*$ -operator. One possibility is to use only the diagonal values of the  $\Lambda$ -operator so the inversion is just a division. On the other hand the best convergence rate is reached by using the full  $\Lambda$ -operator for  $\Lambda^*$ . However, for this inversion of the full  $\Lambda$ -matrix is needed and this is normally not possible, especially for multidimensional problems (see also section 3.2.3).

Therefore a compromise is probably the best solution. In PHOENIX/3D the equivalent of a tridiagonal  $\Lambda$ -matrix is used. This has the advantage that not only local information (diagonal matrix) but also information about neighbouring points is used.



**Figure 3.1** – Nassi–Shneiderman diagram for the main steps of the ALI in PHOENIX/3D.

## 3.2 ALI in PHOENIX/3D

In PHOENIX/3D the model space is represented by a grid of volume cells (voxels). Each voxel holds the information of the relevant physical quantities (e.g. density, opacity,  $J$ ,  $S$ ) where the quantities are constant over the volume of a voxel. It is possible to use a Cartesian, cylindrical or spherical coordinate system for the voxel grid (Hauschildt and Baron 2009). But the main steps of the ALI are not affected by the coordinate system.

Fig. 3.1 shows an overview of the main steps for the Accelerated Lambda iteration in PHOENIX/3D.

### 3.2.1 Formal Solution

$J$  is the average of the specific intensities integrated over all solid angles (see equation 2.11). One possible way to solve this integral is to use the so-called *discrete ordinate method*. For this method the solid angle space of  $4\pi$  is split into solid angle elements  $\Omega_i$  with a finite area. For further discussions we often use the term *ray* for one of these solid angle elements of the solid angle grid with a distinct direction given by  $(\theta_i, \phi_i)$ , where  $0 \leq \theta_i \leq \pi$  and  $0 \leq \phi_i < 2\pi$ . The discretisation of the solid angle space just means that the integral for  $J$  is approximated by some quadrature formula (Mihalas 1978, chap. 3):

$$J_\lambda = \frac{1}{4\pi} \oint I_\lambda(\Omega) d\Omega \approx \frac{1}{4\pi} \sum_{\Omega_i} a_i I_\lambda(\Omega_i), \quad (3.7)$$

where  $\{\Omega_i\}$  are called *quadrature points*,  $\{a_i\}$  the *quadrature weights* and  $I(\Omega_i)$  the *discrete ordinates*.

To estimate  $J$  at a distinct point or voxel in the model space we need to know the values of  $I_\lambda(\Omega_i)$ .  $I_\lambda(\Omega_i)$  can be evaluated by solving the RTE along the propagation path of the light. For this we follow the way of the light for one direction  $(\theta_{ray}, \phi_{ray})$  from the boundary of the model space to the actually considered point (voxel). For fixed values of the source function the RTE can be solved along these propagation paths, which are also

called *characteristics*, by the formal solution (equation 2.20). After the values of  $I$  for all solid angles are known the integral for  $J$  can be evaluated by some quadrature formula.

In PHOENIX/3D the quadrature formula is replaced by a simple “Monte Carlo – Sum” (Hauschildt and Baron 2006). The MC–sum looks the same way as equation 3.7 but the quadrature weights are set to unity. This method is used because for moving media or in Lagrangian frame radiation transport the angles  $(\theta, \phi)$  can vary along a characteristic, and therefore the weights are different for each voxel. By using the “MC–sum” the calculation and storing of these weights for each voxel and direction can be avoided. Although there also exist different methods in PHOENIX/3D for the integration of  $J$ , the MC–sum is usually also used for static media, because the accuracy of the MC–sum is sufficient as long as a high enough number of rays is used. For our case (static media) the “MC–sum” is calculated in two steps: first, the contribution of each ray is added to the sum, then the sum is normalized (divided by the number of rays/quadrature points for the current voxel). The distribution of the rays in the solid angle space is also important. In  $\phi$  the points  $\phi_i$  are distributed equidistantly, in  $\theta$  the points are distributed equidistantly in  $\mu$  from  $[-1, 1]$  where  $\theta_i = \arccos(\mu_i)$ .

The next thing we want to discuss is how the formal solution for one solid angle point (ray) is actually done in PHOENIX/3D. With definition (2.20) the formal solution along a characteristic can be written in the following way (Hauschildt and Baron 2006, see Olson and Kunasz 1987 for a derivation):

$$I_\lambda(\tau_i) = I_\lambda(\tau_{i-1})e^{\tau_{i-1}-\tau_i} + \int_{\tau_{i-1}}^{\tau_i} S_\lambda(\tau)e^{\tau-\tau_i} d\tau, \quad (3.8)$$

where  $i$  labels the points along the characteristic.  $\tau_i$  is the optical depth at that point, where  $\tau_1 = 0$  and  $\tau_{i-1} \leq \tau_i$ . For better readability the wavelength index  $\lambda$  is suppressed for the wavelength dependent quantities  $\tau, S, I, \chi$ .  $\tau$  is calculated via piecewise linear interpolation of  $\chi$  along the characteristic:

$$\Delta\tau_{i-1} = \frac{1}{2}(\chi_{i-1} + \chi_i)|s_{i-1} - s_i| \quad (3.9)$$

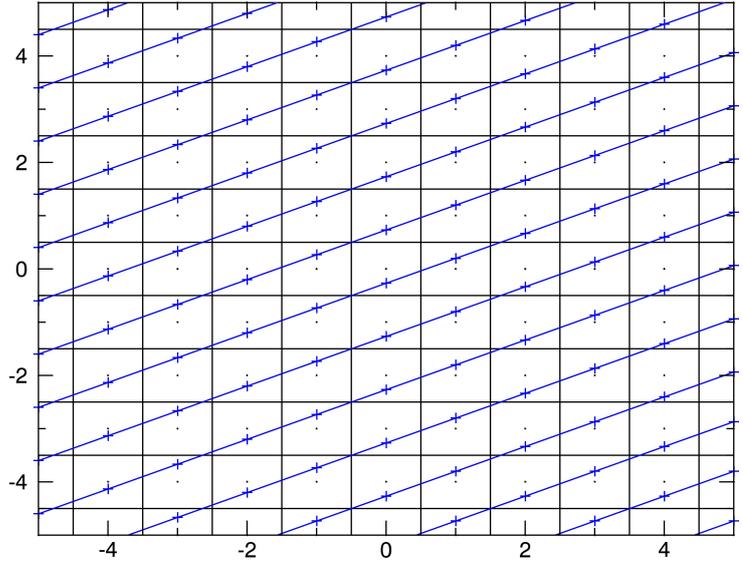
where  $\Delta s = |s_{i-1} - s_i|$  is the geometric distance between point  $i$  and  $i - 1$  along the characteristic (see Fig. 3.2). The integral in equation 3.8 can be solved analytically by replacing  $S(\tau)$  by a linear or parabolic polynomial of the form:

$$\int_{\tau_{i-1}}^{\tau_i} S(\tau)e^{\tau-\tau_i} d\tau = \alpha_i S_{i-1} + \beta_i S_i + \gamma_i S_{i+1}. \quad (3.10)$$

The coefficients for the interpolation can be found in Olson and Kunasz (1987) or Hauschildt and Baron (2006). For the linear interpolation only the values of  $S_{i-1}$  and  $S_i$  are considered. Both interpolation methods can be used in PHOENIX/3D. The parabolic interpolation is in general better in terms of convergence and accuracy.

The linear interpolation has to be used at least for the last integration point (the boundary). It can also be advantageous to use linear interpolation for non–boundary points, especially for regions with low optical depths as this can improve the convergence and stability (Hauschildt and Baron 2006; Auer 2003a). PHOENIX/3D uses automatically

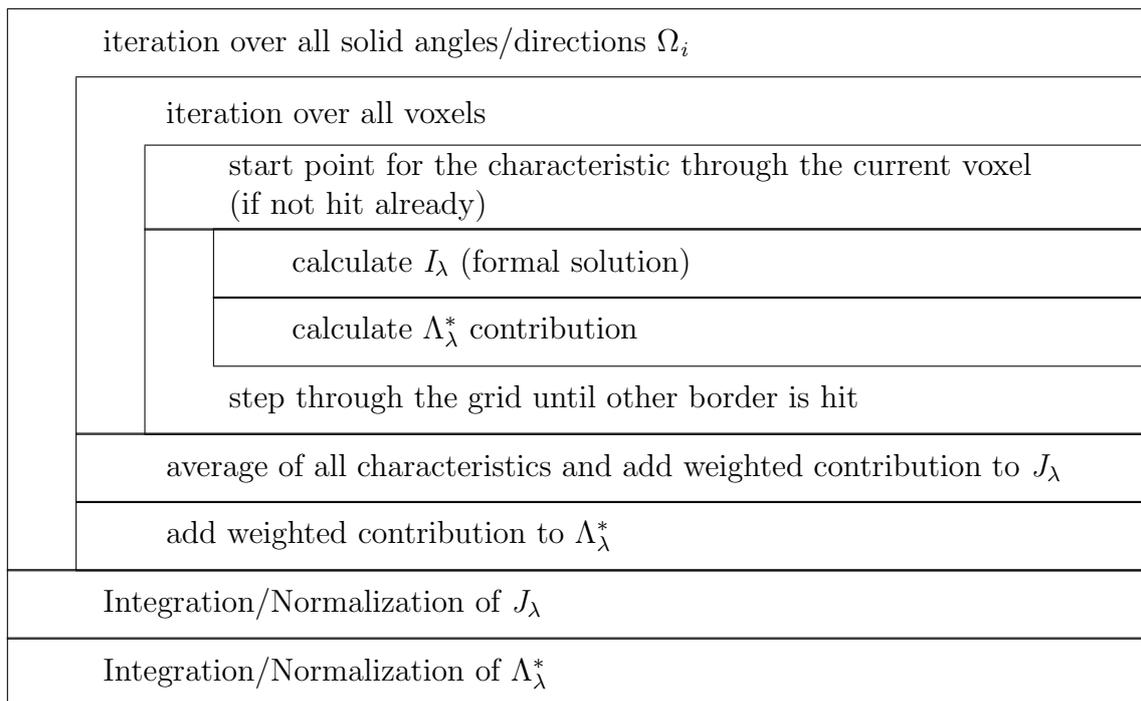
**Figure 3.2** – Sketch for the long characteristics method for a regular 2D grid. The borders of the voxels are given by the black lines. The dots mark their centres. The blue lines show the characteristics for one direction  $(\theta, \phi)$ . The characteristic is started at one border of the grid and followed through the grid until it leaves the grid. The points between which the geometric distance  $\Delta s$  is calculated are marked with “+”. This distance is then used for the calculation of the optical depth  $\Delta\tau_{i-1}$ .



linear interpolation if the optical depth along the characteristic ( $\Delta\tau_{i-1}$ ) drops below a certain value (typically  $10^{-3}$ ). This can be controlled via the parameter *taulin*. Further also drastic changes in the source function along the 3 points for the interpolation step can cause troubles for the piece-wise parabolic interpolation. If the change of the source function along the characteristic is higher than a certain value (typically 100) also linear interpolation is used. This is controlled by the parameter *PPM\_max\_step*.

In PHOENIX/3D the calculation of  $I_\lambda(\Omega_i)$  is not done separately for each voxel, but rather the following way. The starting point for a characteristic at the border of the model space is determined by “shooting” a ray from the center of the voxel in the opposite direction. By this it is ensured that the voxel is hit by the characteristic. Now equation 2.20 is solved by following the characteristic from the starting point through the grid and the target voxel until the characteristic leaves the grid. This method is also called the long or full characteristic method as the characteristic is tracked through the whole model space. The tracking along the characteristic is done by determining the distance to the next voxel along the current direction (the implementation for finding the next voxel depends on the coordinate system). With this distance,  $\Delta\tau_{i-1}$  (equation 3.9) can be calculated. During this tracking the contribution of the current characteristic to  $I(\Omega_i)$  is added to each voxel which is hit by the characteristic. This is repeated until each voxel is hit at least once by a characteristic with direction  $\Omega_i$ . Because of this algorithm it can happen that one voxel is hit more than once by a characteristic with the same direction (but not necessarily at the same position inside the voxel, see also Fig. 4.3). So the value of  $I$  for one direction is not given by a single characteristic but rather by bunch of parallel characteristics. In that case an average of all contributions from the different parallel characteristics is taken. So for one ray with a distinct direction the value of the intensity is given by a certain number of parallel characteristics.

To get  $J_\lambda$  for each voxel this has to be done for all rays (solid angle points)  $\Omega_i$ . For one voxel the sum of the contributions of all rays  $\Omega_i$  is taken. Finally  $J_\lambda$  is evaluated by normalizing this sum as already described above. A schematic overview of this algorithm is shown in Fig. 3.3.



**Figure 3.3** – Nassi-Shneiderman diagram for the long characteristic solver for the formal solution. The solver covers the part for the formal solution and the construction of the  $\Lambda^*$  in the ALI. The construction of the  $\Lambda^*$ -operator is only done in the first iteration of the ALI. The solid angle loop is parallelized (MPI).

### 3.2.2 Construction of $\Lambda^*$

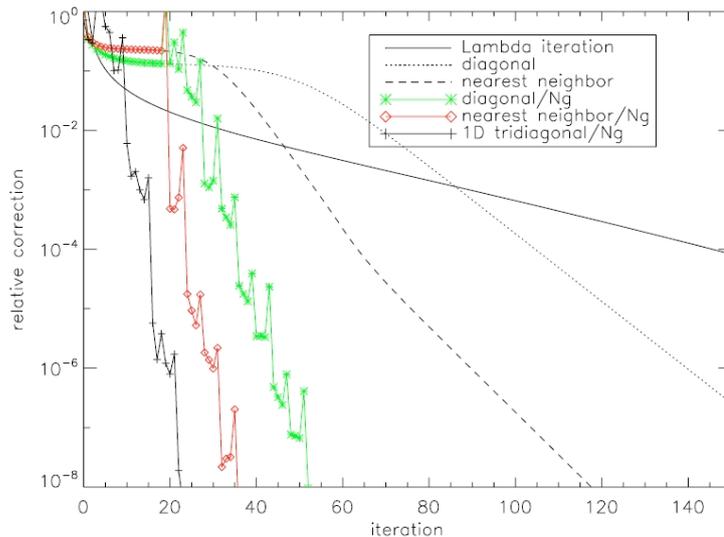
In Olson et al. (1986) and Olson and Kunasz (1987) it is shown that the  $\Lambda$ -operator can be constructed by a formal solution and that the same method used for  $J_{FS}$  can be applied (the same coefficients  $\alpha, \beta$  and  $\gamma$  can be used). As an example let us consider the 1D case. In 1D the  $\Lambda$ -operator is represented by a 2D matrix. To calculate the values of the  $j$ -column of this matrix a test-source function, where the  $j$ -entry equals unity and all other entries are zero, is used (Knop 2007):

$$\begin{pmatrix} \Lambda_{1j} \\ \vdots \\ \Lambda_{jj} \\ \vdots \\ \Lambda_{ij} \end{pmatrix} = \Lambda \cdot \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \quad (3.11)$$

By using the method described in section 3.2.1 the elements of the  $\Lambda$ -operator can be constructed in only one iteration step.

As already mentioned in section 3.1.2 the use of the full  $\Lambda$ -operator is normally not practicable (e.g. matrix inversion). For the 3D-case, where the  $\Lambda$ -operator becomes a six dimensional matrix, memory consumption is also a severe problem. Therefore a diagonal or tridiagonal matrix is used for the  $\Lambda^*$ -operator. But a diagonal matrix considers only the influence of the local source function to the voxel itself.

**Figure 3.4** – Convergence properties for a test case in Cartesian coordinates (a sphere with a grey temperature structure,  $\tau_{max} = 10^4, \epsilon = 10^{-4}$ ). The 3D test runs use  $65^3$  voxels and  $16^2$  solid angle points. The labels indicate the different methods used. Adapted from Hauschildt and Baron (2006).



A non-local operator considers also the interaction of voxel with its neighbours. If only the two nearest neighbours are considered this is equivalent to a tri-diagonal  $\Lambda^*$ -operator (for the 1D case). This approximation especially makes sense for regions with high optical depths where the mean free path of a photon is small. In that case a voxel in the grid is mainly influenced by its direct neighbours and the contribution from distant voxels can be neglected.

In three dimensions the situation is similar except that a voxel has  $3^3 - 1 = 26$  neighbours. This means that for a non-local  $\Lambda^*$  operator 27 (26 neighbours + local voxel) times the total number of voxels have to be stored. For this a data structure containing the effects to its neighbours is attached to each voxel. However, the same method for the construction of  $\Lambda^*$  in 3D as in the 1D case can be used (Hauschildt and Baron 2006). The whole process for the construction of  $\Lambda^*$  in PHOENIX/3D is included in the algorithm for the formal solution. The same interpolation coefficients for the source functions and the same angle integration as for  $J_{FS}$  are used (see also Fig. 3.3). As the  $\Lambda^*$ -operator only contains information about the optical depths the construction has to be done only in the first iteration of the ALI-scheme.

For more details concerning the construction of the  $\Lambda^*$ -operator in PHOENIX/3D see Hauschildt and Baron (2006) and references therein.

### 3.2.3 Solving the Operator Splitting Equation

After the construction of  $\Lambda^*$  and the calculation of  $J_{FS}$  the only thing left is the OS-equation 3.6.

For the non-local operator in 3D the  $\Lambda^*$  matrix is still quite large, which makes a direct inversion of the  $\Lambda^*$ -operator impossible. Therefore iterative solvers for the OS-equation were implemented in PHOENIX/3D. Best results were obtained with the Gauss-Seidel and Jordan methods. For more information concerning the iterative solvers see Hauschildt and Baron (2006) and Knop (2007).

### 3.2.4 Convergence Properties and Performance

The most time consuming part of the ALI in PHOENIX/3D is the calculation of  $J_{FS}$  and the construction of  $\Lambda^*$  (the part which is done in the long characteristic solver, see Fig. 3.3).

The time needed for the construction of the  $\Lambda^*$  is similar to the time needed for one formal solution  $J_{FS}$  (this may change depending on the actual problem to solve). But the memory consumption for storing of the  $\Lambda^*$  can be a problem for large grids (it scales with the *(number of voxels)*<sup>3</sup>). However, as the construction is only done once and the convergence is significantly improved, it is worth the effort.

Nevertheless the limits of a single computer are rapidly exceeded and therefore the code was parallelized. As the formal solution for one solid angle does not depend on the formal solution of the other solid angles the solid angle loop can easily be parallelized (MPI) (see Fig. 3.3). The speedup for the MPI version is nearly optimal (e.g. a factor of 28 for 32 MPI processes). Further the convergence rate can be improved by using Ng-acceleration (Ng 1974).

In Fig. 3.4 the general convergence properties of PHOENIX/3D for different  $\Lambda^*$  operators in comparison to the ordinary  $\Lambda$ -iteration and 1D problems are given. For further tests and examples for performance and convergence see Hauschildt and Baron (2006).



# Chapter 4

## Disk Radiative Transfer with PHOENIX/3D

3D disk radiative transfer with PHOENIX/3D was already done by Hügelmeyer (2009). The disk models therein are limited to the dense inner region of the disk (up to 0.2 AU). Typical protoplanetary disks have a radial extension of several 100 AU. Modelling of such disks causes additional problems for the radiative transfer, especially for discrete ordinate codes like PHOENIX/3D. One main problem is the angular resolution. Small sources like the star or also the inner rim of the disk must be resolved by the given angular resolution which depends on the number of solid angle points used for the integration of  $J$ .

Another problem is the large range in the physical properties. The density at the inner border of the disk is normally several orders of magnitudes larger than in the outer part of the disk. It has to be secured that the resolution of the spatial grid is sufficient to sample the gradients of the physical properties (e.g. opacity). A general discussion of these problems can be found in Auer (2003b).

In this chapter the necessary adaptations to PHOENIX/3D for disk radiative transfer are presented. Section 4.1 describes the disk model and the spatial grid used for this work and section 4.2 shows how we tackle the angular resolution problem.

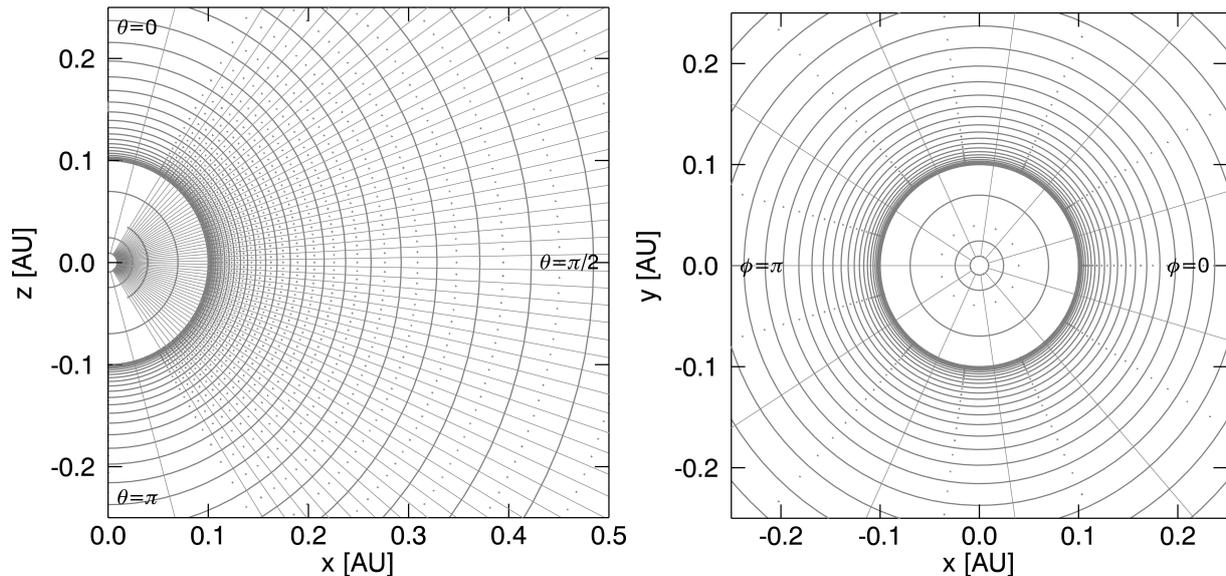
### 4.1 Modelling of the Disk

#### 4.1.1 The Spatial Grid

For disks a cylindrical coordinate system seems to be the more natural choice. In PHOENIX/3D the cylindrical grid does not allow a different spacing or a different number of points for the  $z$ -coordinate depending on the radial position. However, this is needed to cover also large and flaring disks, where the outer regions are much more extended than the inner regions. Therefore the spherical coordinate system  $(r, \theta, \phi)$  was chosen.

In PHOENIX/3D the spherical coordinates are defined as follows:

$(r, \theta = 0, \phi = \textit{degenerate})$  corresponds to the positive  $z$ -axis of a Cartesian coordinate system  $(x, y, z)$ ,  $(r, \theta = \pi, \phi = \textit{degenerate})$  to the negative  $z$ -axis,  $(r, \theta = \pi/2, \phi = 0)$  to



**Figure 4.1** – Example grid for  $n_r=30$ ,  $n_t=30$ ,  $n_p=5$ . The circle in the center represents the border of the star. The inner border of the disk is at 0.1 AU. The center of the voxels are marked by the small dots. The gray lines indicate the border of the voxels. *Left panel:* the grid in the  $x/z$ -plane ( $\phi = 0$ ) with the “big” voxel at the poles. *Right panel:* The grid in the azimuthal plane ( $x/y$ -plane,  $\theta = \pi/2$ ).

the positive x-axis and  $(r, \theta = \pi/2, \phi = \pi/2)$  to the positive y-axis (see also Fig. 4.1). The voxel grid is specified by voxel coordinates ranging from  $(-nr, -nt, -np)$  to  $(nr, nt, np)$ . Where  $-nr$  corresponds to  $r = 0$  and  $nr$  to the border of the model space,  $-nt$  ( $\theta = 0$ ) and  $nt$  ( $\theta = \pi$ ) are the poles, and  $-np$  represents  $\phi = 0$ .

### Radial Grid

The configuration for the radial grid is mainly driven by the density distribution of the disk in radial direction (see section 4.1.3).

For the disk, ranging from  $r_{in}$  to  $r_{out}$ , a logarithmic scaling similar to that of Hügelmeyer (2009) of the form

$$r_i = r_{in} \exp(i^\alpha \Delta d) , \quad (4.1)$$

is used, with

$$\Delta d = \frac{\log(r_{out}) - \log(r_{in})}{N_r^\alpha} . \quad (4.2)$$

Where  $r_{in}$ ,  $r_{out}$  are the inner and outer border of the disk,  $N_r$  is the total numbers of voxels used for the radial dimension of the disk and  $i$  is the voxel index ranging from 1 to  $N_r$ .  $\alpha$  is a parameter to control the coarseness of the grid.  $\alpha = 1$  means a logarithmically equidistant grid,  $\alpha > 1$  means a finer grid near  $r_{in}$  and a coarser grid near  $r_{out}$ .

The innermost radial point is just given by the dimension of the star. The space between the star and the disk is empty and therefore only one radial point is used for this region.

### $\theta$ -Grid

In PHOENIX/3D there is only the possibility for an equidistant  $\theta$ -grid, but for disks it is also beneficial to have a non-equidistant grid in the  $\theta$ -coordinate because of the exponential density distribution in the vertical direction. So a non-equidistant  $\theta$ -grid was implemented in PHOENIX/3D (the implementation is similar to the already existing implementation for the  $r$ -coordinate). With this implementation it is now possible to use for example a logarithmic grid for the  $\theta$ -coordinate. Further it is possible to reduce the number of  $\theta$ -points as the empty space around the poles can now be covered by only one  $\theta$ -point (see Fig. 4.1).

As we will see in section 4.2 it is beneficial to use the same distribution for the spatial  $\theta$ -points as it is used for the solid angle grid (see section 3.2.1). In that case the  $\theta$ -points are given by

$$\begin{aligned}\Delta\mu &= \frac{2}{N_\theta - 1} \\ \theta_i &= \arccos(i \cdot \Delta\mu),\end{aligned}\tag{4.3}$$

where  $N_\theta$  is the total number of points used for the  $\theta$ -grid.

### $\phi$ -Grid

Because of the symmetry of the disk in the  $\phi$ -direction a normal equidistant scaling with a small number of points is used.

An example grid is given in Fig. 4.1.

## 4.1.2 The Star

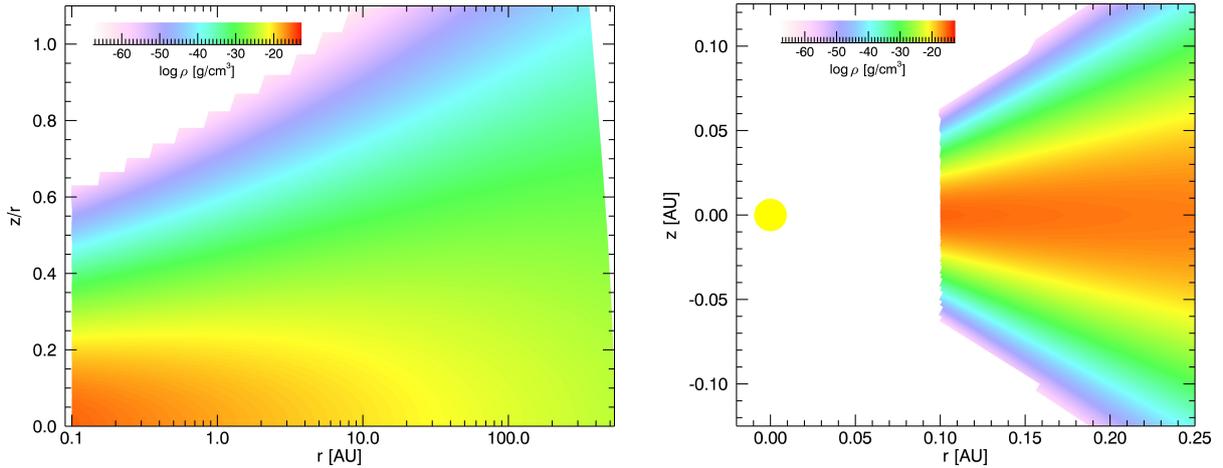
The star is modelled as a sphere with a given radius  $R_*$  in the center of the coordinate system. For the moment the star is just a uniformly radiating blackbody with a distinct temperature  $T_{eff}$ . For the radiative transfer the surface of the star is treated as a boundary condition. For a ray starting at the surface of the star  $I_\lambda(\tau_\lambda = 0) = B_\lambda(T_{eff})$ . The possibility of using a spectrum for the star is not implemented yet. But in principle it is possible without changing the radiative transfer, only the initializing routines have to be adapted.

## 4.1.3 The Disk

The density structure for an azimuthally symmetric disk in hydrostatic equilibrium is given by

$$\rho(r, z) = \frac{\Sigma(r)}{\sqrt{2\pi} \cdot h(r)} \exp\left(-\frac{z^2}{2h(r)^2}\right),\tag{4.4}$$

where  $\Sigma(r)$  is the surface density and  $h(r)$  the scale height ( $r, z$  are cylindrical coordinates). Although the radial profile of the surface density can be quite complex and



**Figure 4.2** – The density structure of the disk in the  $x/z$  - plane as calculated in PHOENIX/3D. The *left panel* shows the density distribution for the whole disk (the ordinate shows  $z/r$ ). The *right panel* shows the inner border of the disk and the star with a radius of  $2R_{\odot}$ . The white area is empty ( $\rho = 0$ ). Here  $r$  and  $z$  are cylindrical coordinates.

the scale height depends on the equilibrium between thermal pressure and stellar gravity (Williams and Cieza 2011), it is common to approximate both by simple power laws.

Therefore, we use the same approach as presented in Pinte et al. (2009). The parameters for the density distribution are: the mass of the disk  $M_{disk}$ , the inner border of the disk  $r_{in}$  and the outer border of the disk  $r_{out}$ . The mass of the disk is distributed between these two borders via equation 4.4.

For the surface density  $\Sigma(r)$  and the scale height  $h(r)$  a power-law distribution of the form

$$\Sigma(r) = \Sigma_0 \cdot \left(\frac{r}{r_0}\right)^{p_{\Sigma}} \quad (4.5)$$

$$h(r) = h_0 \cdot \left(\frac{r}{r_0}\right)^{p_h} \quad (4.6)$$

is used, where  $h_0$  is the scale height at radius  $r_0$ .

With given values for  $M_{disk}$ ,  $r_{in}$ ,  $r_{out}$ ,  $r_0$ ,  $h_0$ ,  $p_h$ , and  $p_{\Sigma}$ ,  $\Sigma_0$  can be evaluated via the integral

$$M_{disk} = 2\pi \int_{r_{in}}^{r_{out}} \Sigma(r) r \, dr \quad (4.7)$$

and is therefore given by

$$\Sigma_0(r) = \frac{M_{disk}}{2\pi \cdot \frac{r_0^{p_{\Sigma}}}{p_{\Sigma}+2} \cdot \left(r_{out}^{(p_{\Sigma}+2)} - r_{in}^{(p_{\Sigma}+2)}\right)}. \quad (4.8)$$

For the spatial grid a spherical coordinate system is used (see section 4.1.1). To calculate  $\rho(r, \theta, \phi)$  for each voxel in the grid, the spherical coordinates are converted to cylindrical coordinates and the density is calculated via equation 4.4. As the disk is symmetric in  $\phi$ , the values are just copied.

One problem that arises is the sharp inner wall of the disk. In spherical coordinates it happens that at the sharp inner wall only a fraction of a voxel is inside the disk but not its center. Therefore we get a jagged wall. To weaken this effect the density for one voxel is not only calculated for its center, but also for a large number of uniformly distributed points inside the voxel. The according density for the voxel is then given by the average of all points. By this it is secured that the density is greater than zero, also for voxels which are only partly inside the disk. Although this method reduces the jags, still a fine radial and  $\theta$ -grid is needed for a good sampling of the inner wall. However, the typically high optical depths in this region require a fine grid anyway.

A similar mechanism would be necessary if also the outer wall of the disk were sharp. This is not done as additional radial points would be required. It is just secured that the highest point of the disk in  $z$ -direction at the given outer radius  $r_{out}$  is still inside the model sphere. Therefore the extension of the disk in radial direction is larger than the given outer radius  $r_{out}$ .

It is also possible to define a cut-off for the density structure in  $z$  direction. The grid is only filled up to a given multiple of the scale-height  $h(r)$  (typically  $10h(r)$ ). To test the density structure, the total mass of the disk is calculated by a integration of the density  $\rho(r, \theta, \phi)$  in spherical coordinates and compared to the given mass  $M_{disk}$ . The deviation is usually below 1%. Further an IDL-script for testing the optical depths, especially the expected  $\Delta\tau$  value between the voxels (radial, and vertical), was implemented. This is helpful for finding the best grid configuration (e.g. the best  $\alpha$  parameter in equation 4.1) for the actual problem.

An example for the density structure is given in Fig. 4.2.

## Dust Properties

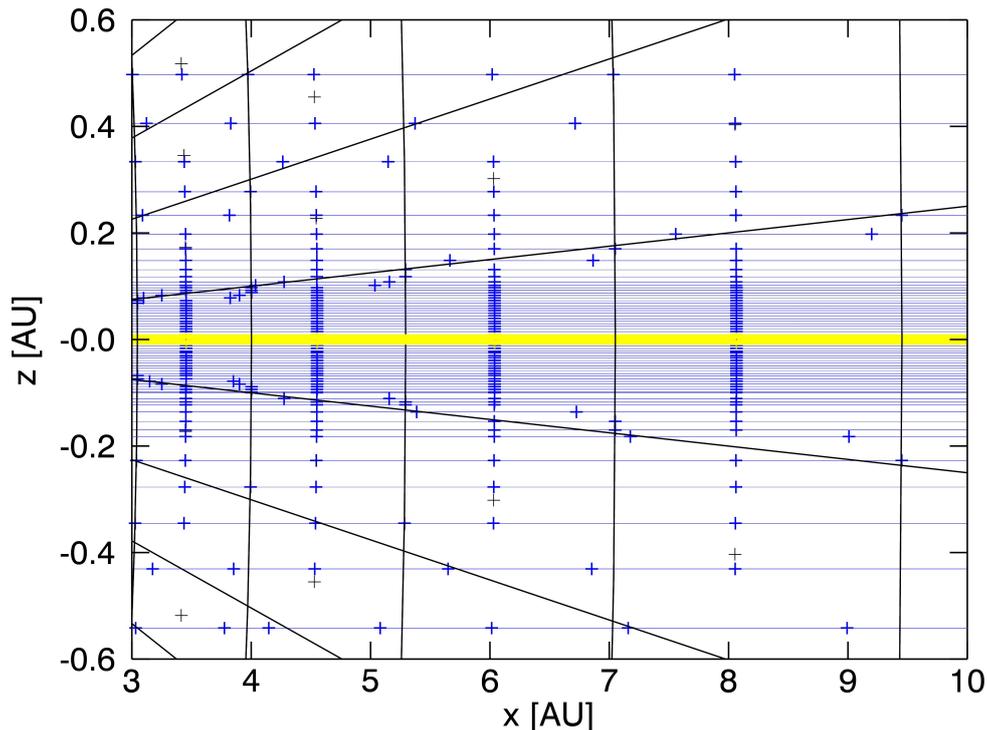
For the radiative transfer the absorption and scattering coefficients  $\kappa_{\lambda}^{abs}$  and  $\kappa_{\lambda}^{sca}$  for the dust are needed. Although the absorption and scattering coefficients could be pre-calculated in PHOENIX/3D, this was not done for this work. For the tests in chapter 6 pre-tabulated values from Pinte et al. (2009) were used.

## 4.2 The Resolution Problem

As already mentioned discrete ordinate codes may have problems with resolving small objects in the model space. As one can imagine this is definitely true for a star surrounded by a rather huge disk. Also the inner region of the disk (the *inner rim*) is quite small compared to the rest of the disk (see Fig. 4.2 for an example). Let's discuss this problem first on the basis of the star.

### 4.2.1 Resolving the Star

For the radiative transfer it is crucial to correctly treat the light from the star. It has to be secured that radiation from the star reaches each voxel in the grid. For this at



**Figure 4.3** – This plot shows the characteristics for one ray with direction ( $\theta = \pi/2, \phi = 0$ ) penetrating through the x-z plane of the spherical model space. The yellow solid lines indicate the characteristics starting at the inner boundary condition - the star, the blue solid lines the characteristics which start at the outer border of the model grid. The blue “+” symbols indicate the points between which the geometric distance  $\Delta s$  is calculated (at which point the voxel is hit). The solid black lines are the border of the voxels and the black “+” symbols their center. Although this example is the result of a 3D-calculation, for more clarity, only the characteristics in the x-z plane are plotted. We can see in this plot how the characteristics for one ray are placed in a spherical grid. The denser distribution at  $z = 0$  is caused by the smaller voxels at the inner region of the model space as this smaller voxels also have to be hit by at least one characteristic.

least one of the rays tracked through a voxel must hit the star. As the size of the star is very small compared to the typical dimension of the disk this is usually not the case for discrete ordinate methods (the “on/off” problem; Auer 2003b).

There exist two possibilities to solve this problem. In spherical coordinates it is possible to use the same spatial grid in  $(\theta_{vox}, \phi_{vox})$  as is used for the solid angle grid  $(\theta_{ray}, \phi_{ray})$  (the coordinates  $(\theta_{ray}, \phi_{ray})$  are defined the same way as the spherical spatial coordinates, see section 4.1.1). By this it is secured that for all voxels at least one ray (the ray where  $\theta_{vox} = \theta_{ray}$  and  $\phi_{vox} = \phi_{ray}$ ) goes through the center of the spherical grid and therefore also hits the star. This method has the drawback that we lose flexibility for the choice of the spatial grid distribution.

Therefore also another method was implemented. For this, additional rays with directions given by the spatial coordinates  $\theta_{vox}, \phi_{vox}$  are added to the normal angular grid. To secure that each voxel is hit by the star, each one of these rays has to be tracked only through the voxels where the spatial coordinates are equal to the angular coordinates of the actual ray ( $\theta_{vox} = \theta_{ray}$  and  $\phi_{vox} = \phi_{ray}$ ). So, only one additional ray is added to the angular grid for each voxel. Although now the angular grid points are not the same for each voxel, this is not a problem because the integration of  $J$  is done by the MC-sum (see section

3.2.1). It is also possible to track this artificially added rays through all voxels, but this costs additional computational time and the result is the same (at least for the correct treatment of the star).

Now we have secured that each voxel gets light from the star, but there is still another problem. It is caused by the limited number of rays (solid angle points) which can be calculated (limited by computational time), therefore the star is not completely resolved. For example, with  $10^2$  solid angle points, one solid angle element for the integration of  $J$  is roughly  $10^{-1}$  ( $4\pi/100$ ). A star with a size of  $2 R_{\odot}$  at a distance of 1 AU would have a solid angle of  $\approx 3 \cdot 10^{-4}$  (see equation 4.9). If we now assume that the star sits in a vacuum sphere all rays except the rays which hit the star have intensity zero. For the integration of  $J$  all solid angle elements have the same size of  $10^{-1}$ . This means that for the integration of  $J$  the star seems to have the solid angle of  $10^{-1}$  and not  $\approx 3 \cdot 10^{-4}$ , and so the contribution from the ray, hitting the star to  $J$  is much too large. This effect becomes larger with distance from the star. To completely resolve the star at a distance of 400 AU more than  $10^9$  solid angle points would be necessary, which is not practical.

To overcome this, the weights for the rays which hit the star have to be corrected. For this we first need the real solid angle of the star at a given distance  $r$ . From geometrical considerations the following equation for the solid angle of the star  $SA_*$  can be derived

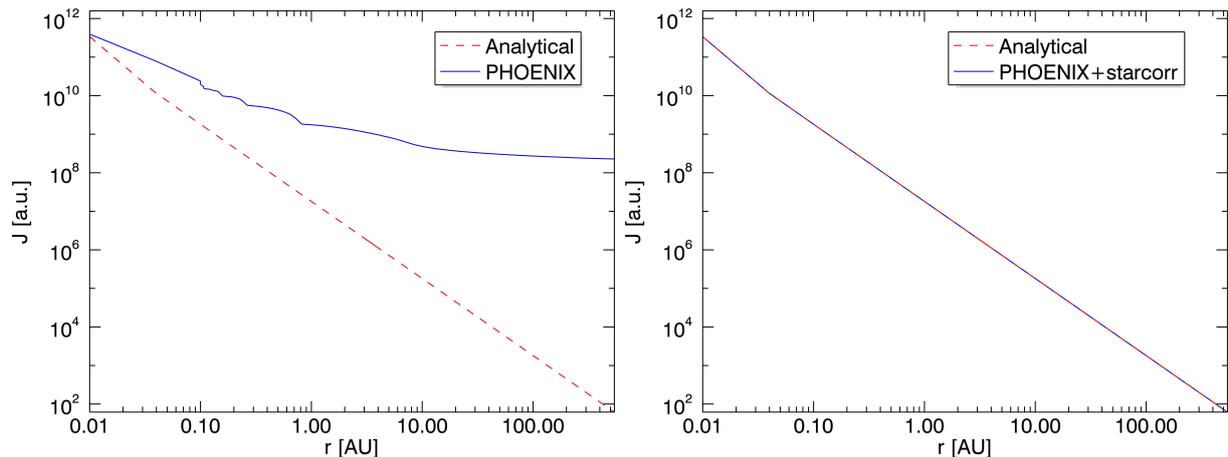
$$SA_*(r) = 2\pi \left( 1 - \sqrt{1 - (R_*/r)^2} \right), \quad (4.9)$$

where  $R_*$  is the radius of the star.

By knowing the real solid angle of the star we can correct the weight for the  $J$  integration. In PHOENIX/3D all solid angle points have the same weight (for our case unity). The solid angle element for one ray is given by  $SA_{ray} = 4\pi/N_{rays}$ , where  $N_{rays}$  is the total number of solid angle points (rays). Further we have to consider that for one direction a bunch of parallel characteristics can hit the voxel, where some of them hit the star and some not (see also Fig. 4.3). The corrected weight  $W_*$  for a *star ray* is then given by

$$W_* = \frac{SA_*}{\frac{N_{c*}}{N_c} SA_{ray}}, \quad (4.10)$$

where  $N_{c*}$  is the number of parallel characteristics starting at the star and  $N_c$  is the total number of characteristics going through the voxel.  $W_*$  would be unity if the star is correctly resolved. We have also to consider that it is possible that there is more than one ray for a voxel which hits the star. This is important for the adjustment of the weights for the non-star rays. To do the integration for  $J$  correctly the difference of  $W_*$  to unity is added in equal parts to all the non-star rays. The weight for the non-star rays is then  $> 1$ , but the sum of all weights, needed for the normalization of  $J$  remains the same as for the case without the correction. For this we have to know in advance how many star rays and non-star rays we have for each voxel. Therefore a pre-RT step is needed, for which also the long characteristics solver is used but only the tracking is done, while no intensities are calculated. We end up with two different weights, one for the star rays and one for the non-star rays, although the values are different for each voxel (as the weights depend on the distance to the star and the value of  $\frac{N_{c*}}{N_c}$ ). To test this method a simple test case was created. In this test case the star is the only source of radiation and is



**Figure 4.4** – Results for the test problem used to test the correct treatment of the star. The red dashed line shows the analytical solution for  $J(r)$  (equation 4.11), the blue solid line the solution of PHOENIX/3D. The *left panel* shows the results without correction for the star (with  $80^2$  solid angle points), the *right panel* with correction. With the correction, the results of PHOENIX/3D are identical with the analytical solution.

surrounded by vacuum. For such a configuration an analytical solution for the radiative transfer exists.  $J(r)$  is then given by

$$J(r) = I_* \frac{SA_*(r)}{4\pi}, \quad (4.11)$$

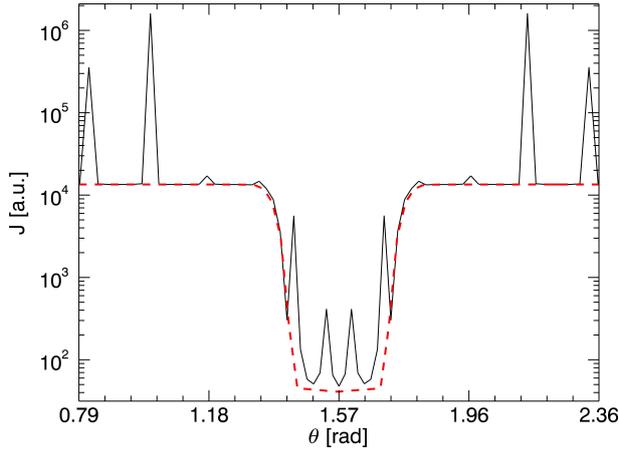
where  $I_*$  is the intensity of the star (the boundary condition) and  $SA_*$  is given by equation 4.9. The PHOENIX/3D results for this test case, with and without correction, in comparison to the analytical solution are shown in Fig. 4.4.

The only thing which cannot be tested with this test case is the weighting for the non–star rays as for all non–star rays the intensity is zero, which is not the case for real models. But for a large number of rays the weights for the non–star rays only deviates slightly from unity. Typically the deviation is smaller than  $10^{-2}$  and therefore no significant inaccuracy is expected.

### 4.2.2 Resolving the Inner Rim

Unfortunately the star is not the only small radiation source in our model space. Also the inner rim of the disk is rather small in comparison to the rest of the disk and it is also a more complex structure (nonspherical). So it is quite hard to estimate a typical size for the rim structure. If we look at the example for the density structure in Fig. 4.2 we can say that a typical size for the rim is  $\approx 10^{-1}$  AU (in comparison a star with  $2R_\odot$  has a size of  $\approx 10^{-2}$  AU). Therefore the problem is similar to the problems already discussed for the star. The resolution problem for the inner rim is mainly influenced by two things:

- The distribution of the solid angle points (rays) (see section 3.2.1). Around  $\theta_{ray} = \pi/2$  the solid angle grid is finer due to the shape of the *arccos*-function. So for voxels near the midplane of the disk more angular points are placed toward the center (the star and the rim), but for voxels in the upper regions this is not the case as the solid angle grid is the same for all voxels.



**Figure 4.5** – Comparison of the results for  $J$  for a logarithmic spatial grid in  $\theta$  (*black solid line*) with the results where the spatial grid and angular grid are aligned (*red dashed line*).  $J$  is plotted along the  $\theta$  coordinate at some distance  $r$ . For both the same number of rays and the same number of voxels is used. It is clearly seen that  $J$  can deviate up to two orders of magnitude from the expected results, if the grids are not aligned.

- Voxels are not only hit by one characteristic but rather by a bunch of parallel characteristics and the intensity is given by the average of the intensities of all characteristics. Although this should improve the accuracy it may also be a problem. A voxel in the outer part of the disk is rather large,  $\gtrsim 10$  AU in each direction. Therefore it is hit by many parallel characteristics. The distribution of these characteristics is mainly driven by the spatial grid. For a ray which hits the inner region of the disk more characteristics are coming directly from the inner rim than from the area around the rim (see also Fig. 4.3) and therefore the average value for the intensity is rather inaccurate, as the characteristics coming directly from the inner rim are overweighted.

However, even with a large number of rays (e.g.  $256^2$ ) there still remain effects caused by bad resolution. The consequences of the too low resolution for the inner rim manifests in a jagged pattern for  $J$  in the vertical direction of the disk (see Fig. 4.5).

Therefore the same method as used for the star was applied. In general this is working well, but it was not possible to remove all effects because it is hard to define the size of the region for which the correction has to be done (it is not known how many rays hit the inner rim). Further it was tried to add additional equally distributed parallel characteristics to the outer voxels to get a more accurate intensity for one ray. But it turned out that again a very large number of additional characteristics is needed, and that it is more efficient to add more angular points.

The inner rim problem could only be solved by using the same distribution for the spatial coordinate  $\theta_{vox}$  as it is used for the solid angle grid (see equation 4.3).

As already mentioned a voxel can be hit by more than one characteristic for a ray. This is especially true for the outermost voxels as they are quite large. If we now consider one outermost voxel and the ray where  $\theta_{ray} = \theta_{vox}$  and  $\phi_{ray} = \phi_{vox}$ , the important dimension of the voxel is the area normal to the ray direction. For the 2D example in Fig. 4.3 this would be the vertical dimension of the voxel. This area of the voxel also defines the region from where the characteristics are coming. Typically the dimension of the outermost voxels are much larger than the typical dimension of the structure in the central region of the model space (the star and the inner rim). For this voxel the central region is covered by many parallel characteristics of the considered ray. For a neighbouring ray that does not go directly through the center of the model space the size of the region from where the

characteristics are coming is the same as for the central ray, because the size is defined by the size of the voxel surface. If we now have, for example, more solid angle points in  $\theta_{ray}$  than for the spatial  $\theta_{vox}$ -coordinate, the areas for the two neighbouring rays are overlapping (imagine a direction which is just slightly off center for the voxel). The intensities for each of these two rays is given by the average of all these characteristics. As some characteristics of these two rays are coming from the same area, information about the same region (e.g. the inner rim) is included for both intensities. This would not happen if we only had one characteristic for one ray. If we now use the same number of  $\theta$ -points for the solid angle and the spatial grid and also the same distribution this overlapping is weakened, especially for the central region. This is probably the reason why the jags vanish if the two grids are aligned. Actually the aligned grids are only necessary in  $\theta$  but not in  $\phi$ , probably because the disk is azimuthally symmetric.

Although the aligning of the angular and spatial grid solves the resolution problem for the inner region, it also has its disadvantages. If more spatial points are needed to better sample the physical structure of the disk we automatically need more solid angle points, which increases computational time. Further the spatial grid in  $\theta$ -direction has to follow the distribution of the solid angle points which is given by the integration method used for  $J$ , therefore it is not possible to adapt the spatial grid to the given physical structure.

Besides these drawbacks the method of aligning the grids works quite well and is therefore used for the tests presented in chapter 6. However, it is still possible to use non-aligned grids in combination with the star correction algorithm. This is especially useful for code development as then for the testing a small number of rays can be used. Further, the resolution problem does not occur in the inner regions of the disk as there the angular resolution is good enough, so if one is only interested in the inner region of the disk, resolving the inner rim is not a problem at all.

# Chapter 5

## Temperature Correction

Until now there was no method in PHOENIX/3D for solving the radiative equilibrium equation (REE) implemented. In this section a new temperature correction scheme for PHOENIX/3D based on the  $\Lambda^*$ -operator is presented. The main idea for this scheme is from Voitke (2011, priv. comm.).

### 5.1 Solving the Radiative Equilibrium Equation

To get the temperature distribution for the disk we have to solve the implicit radiative equilibrium equation (REE) 2.29 for the temperature.

For the wavelength space we assume for now some discrete wavelength grid and replace the integrals of the REE by a simple sum:

$$\sum_l \kappa_{ijk}^l B^l(T_{ijk}) \Delta\lambda_l = \sum_l \kappa_{ijk}^l J_{ijk}^l \Delta\lambda_l . \quad (5.1)$$

Where  $ijk$  are the spatial indices for the 3D grid and  $l$  is the index for the wavelength grid.  $\Delta\lambda_l$  represents the integration weight for the wavelength point  $l$ .  $B^l(T_{ijk})$  is the value of the Planck function for wavelength  $l$  and temperature  $T$  at position  $ijk$  in the grid.

Following the idea of Voitke (2011) we try to solve the REE by a Newton-Raphson iteration of the form

$$x_{new} = x_{old} - \frac{f(x_{old})}{f'(x_{old})} . \quad (5.2)$$

For this we nullify equation 5.1

$$F_{ijk} := \sum_l (J_{ijk}^l - B^l(T_{ijk})) \kappa_{ijk}^l \Delta\lambda_l = 0 . \quad (5.3)$$

$F_{ijk}$  is now our function  $f(x)$  for the Newton-Raphson scheme 5.2. As we want to solve  $F_{ijk}$  for  $T$  we further need the partial derivative of  $\frac{\partial F_{ijk}}{\partial T_{mno}}$ . Under the assumption of

temperature independent opacities the derivative is given by

$$F'_{ijk}{}^{mno} := \frac{\partial F_{ijk}}{\partial T_{mno}} = \sum_l \left( \frac{\partial J_{ijk}^l}{\partial T_{mno}} - \frac{\partial B_{ijk}^l}{\partial T_{mno}} \right) \kappa_{ijk}^l \Delta \lambda_l \quad (5.4)$$

where  $mno$  are again the spatial indices and the abbreviation  $B_{ijk}^l := B^l(T_{ijk})$  is used. Although we now have our two main ingredients  $F_{ijk}$  and the derivative  $F'_{ijk}$ , we do not know how to evaluate the derivative  $\frac{\partial J_{ijk}^l}{\partial T_{mno}}$ . As we have seen in section (3.1),  $J$  can be calculated by using the  $\Lambda$ -operator on  $S$ . By using Einstein's summation convention this can be formally written in index notation as

$$J_{ijk} = \Lambda_{ijk}{}^{mno} S_{mno} . \quad (5.5)$$

Further we know that the source function depends on the temperature via the Planck function  $B$  (equation 2.15). Therefore we can write the derivative  $\frac{\partial J_{ijk}^l}{\partial T_{mno}}$  as

$$\frac{\partial J_{ijk}^l}{\partial T_{mno}} = \frac{\partial J_{ijk}^l}{\partial S_{mno}} \frac{\partial S_{mno}^l}{\partial T_{mno}} = \Lambda_{ijk}{}^{mno} \frac{\partial S_{mno}^l}{\partial T_{mno}} \quad (5.6)$$

where we have used

$$\frac{\partial J_{ijk}}{\partial S_{mno}} = \Lambda_{ijk}{}^{mno} . \quad (5.7)$$

As we know from section (3.2) usually the full  $\Lambda$ -operator is replaced by the approximate operator  $\Lambda^*$ , therefore we do this also for our derivative

$$\frac{\partial J_{ijk}^l}{\partial T_{mno}} \approx \Lambda_{ijk}^{*lmno} \frac{\partial S_{mno}^l}{\partial T_{mno}} . \quad (5.8)$$

For the  $\Lambda^*$ -operator used in PHOENIX/3D this means that we only consider the derivatives of the neighbouring voxels and the indices  $mno$  can only have values  $ijk = mno$  or  $m = i \pm 1, n = j \pm 1, o = k \pm 1$ .

With equation 5.8 we can rewrite equation 5.4 and get

$$F'_{ijk}{}^{mno} = \sum_l \left( \Lambda_{ijk}^{*lmno} \frac{\partial S_{mno}^l}{\partial T_{mno}} - \delta_{ijk}^{mno} \frac{\partial B_{mno}^l}{\partial T_{mno}} \right) \kappa_{ijk}^l \Delta \lambda_l , \quad (5.9)$$

where  $\delta_{ijk}^{mno}$  is the Kronecker delta (all elements are zero except for  $ijk = mno$  where they are equal to unity). The Kronecker delta is used because all the derivatives  $\frac{\partial B_{ijk}^l}{\partial T_{mno}}$  are zero anyway except for  $ijk = mno$ .

The next thing we want to do is to eliminate the derivative of  $S_{mno}^l$  to avoid an otherwise required numerical evaluation. We use equation 2.15 and do the derivative with respect to  $T$  explicitly

$$\frac{\partial S_{mno}^l}{\partial T_{mno}} = \epsilon^l \frac{\partial B_{mno}^l}{\partial T_{mno}} + (1 - \epsilon^l) \frac{\partial J_{mno}^l}{\partial T_{mno}} . \quad (5.10)$$

By using equation 5.8 and by bringing the derivative of  $S$  to the left-hand side we find

$$\begin{aligned} \frac{\partial S_{mno}^l}{\partial T_{mno}} &= \epsilon^l \frac{\partial B_{mno}^l}{\partial T_{mno}} + (1 - \epsilon^l) \Lambda_{mno}^{*lmno} \frac{\partial S_{mno}^l}{\partial T_{mno}} \\ &= \frac{\epsilon^l}{1 - ((1 - \epsilon^l) \Lambda_{mno}^{*lmno})} \frac{\partial B_{mno}^l}{\partial T_{mno}} . \end{aligned} \quad (5.11)$$

and using equation 5.11 in equation 5.9 gives

$$\begin{aligned} F'_{ijk}{}^{mno} &= \sum_l \left( \frac{\Lambda_{ijk}^{*lmno} \epsilon^l}{1 - ((1 - \epsilon^l) \Lambda_{mno}^{*lmno})} \frac{\partial B_{mno}^l}{\partial T_{mno}} - \delta_{ijk}^{mno} \frac{\partial B_{mno}^l}{\partial T_{mno}} \right) \kappa_{ijk}^l \Delta \lambda_l \\ &= \sum_l \left( \frac{\Lambda_{ijk}^{*lmno} \epsilon^l}{1 - ((1 - \epsilon^l) \Lambda_{mno}^{*lmno})} - \delta_{ijk}^{mno} \right) \frac{\partial B_{mno}^l}{\partial T_{mno}} \kappa_{ijk}^l \Delta \lambda_l . \end{aligned} \quad (5.12)$$

Equations 5.3 and 5.12 now can be used for a Newton-Raphson step

$$T_{ijk}^{new} = T_{ijk}^{old} - \frac{F_{ijk}}{F'_{ijk}} . \quad (5.13)$$

To avoid matrix inversion the following form of the equation 5.13 is more convenient

$$F'_{ijk}{}^{mno} \delta T = -F_{ijk} . \quad (5.14)$$

This equation has to be solved for  $\delta T$ , and then we get our new temperature for the next step. But we can also use  $\delta T$  to update the mean intensity  $J$ , by using once again equation 5.8, but now only for the change  $\delta J$

$$\delta J_{ijk}^l = \Lambda_{ijk}^{*lmno} \frac{\partial S_{mno}^l}{\partial T_{mno}} \delta T_{mno} , \quad (5.15)$$

and with equation 5.11 we find

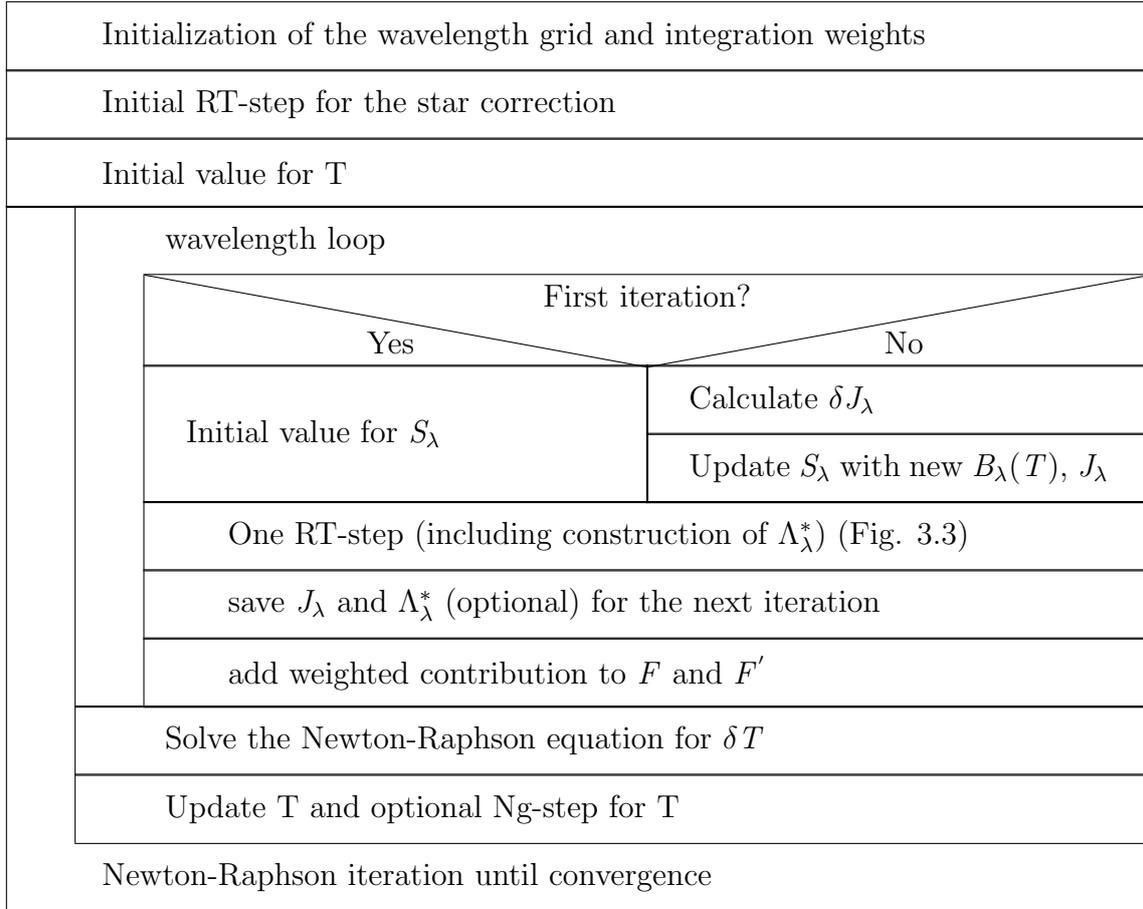
$$\delta J_{ijk}^l = \frac{\Lambda_{ijk}^{*lmno} \epsilon^l}{1 - ((1 - \epsilon^l) \Lambda_{mno}^{*lmno})} \frac{\partial B_{mno}^l}{\partial T_{mno}} \delta T_{mno} . \quad (5.16)$$

$\delta J_{ijk}^l$  can now be used to update the source function prior to the next radiative transfer step. With equation 5.14 we can now solve the REE equation and with equation 5.16 we get the impact of the temperature change on the mean intensity and the source function.

## 5.2 Implementation

We have already seen in section (3.2) how the RTE can be solved with the help of the  $\Lambda^*$ -operator. How this  $\Lambda^*$ -operator can be used to solve the REE was described in section (5.1). Fig. 5.1 shows a graphical overview of the implementation for the new temperature correction scheme by combining this two methods.

Let us start from inside out. The main things happen in the wavelength loop. At first



**Figure 5.1** – Nassi–Shneiderman diagram for the temperature correction scheme.

one radiative transfer step (see Fig. 3.3) for the actual wavelength is one. It is really only one iteration step of the iteration scheme described in section 3.2. In this step the  $\Lambda_\lambda^*$ -operator is constructed (if required) and we also get our first value for  $J_\lambda$ . Both values are saved for the next Newton-Raphson iteration. By storing the  $\Lambda_\lambda^*$ -operator for each wavelength the construction of the operator is only required in the first Newton-Raphson iteration. However, in the case of large grids or a large number of wavelength points memory requirements may be a problem. Therefore the storing of the  $\Lambda^*$ -operator is optional, and it is also possible to newly create the  $\Lambda^*$ -operator for each iteration step.

The next step is the construction of  $F_{ijk}$  and  $F'_{ijk}{}^{mno}$ . The construction of  $F_{ijk}$  is more or less given by equation 5.3. The value for  $J_{ijk}^l$  is given by the radiative transfer and  $B_{ijk}^l(T_{ijk})$  is given by the actual or initial temperature. The integration over wavelength is simply done with a sum over all wavelength points  $l$  with the corresponding weights  $\Delta\lambda_l$ . The wavelength grid and weights are initialised at the beginning of the Newton-Raphson-iteration.

As we can see from equation 5.9,  $F'_{ijk}{}^{mno}$  has the same form as the  $\Lambda^*$ -operator. Therefore the same data structure as for the  $\Lambda^*$  operator is used. In the case of the nearest neighbour method the indices  $(m, n, o)$  have the range  $[-1, 1]$ . Similar to  $\Lambda^*$ ,  $F'_{ijk}{}^{lmno}$  holds the data of all surrounding neighbouring voxels  $mno$  of voxel  $ijk$ . For the construction of  $F'_{ijk}{}^{mno}$  the  $\Lambda^*$ -operator and the derivative of the Planck function (which is done analytically) are needed.

After the wavelength loop and the integration of  $F_{ijk}$ ,  $F'_{ijk}{}^{mno}$  we have to solve the Newton-Raphson equation 5.14. It has a similar form as the OS-equation 3.6. For the same reasons as described in section 3.2.3 an iterative method for solving equation 5.14 is used. For the implementation the Gauss-Seidel method was chosen. As for the OS-equation the iterative method is very fast and the time needed for solving the Newton-Raphson equation is negligible in comparison to the RT-step.

In the next iteration step we now can use  $\delta T$  to improve  $J_\lambda$  (equation 5.16) and calculate  $B_\lambda$  with the new temperature. These values are used to update the source function  $S_\lambda$  prior to the next RT-step. This is done until some convergence criterion for  $T$  is reached.

To avoid negative temperature caused by too large values of  $\delta T$  during the first iterations a simple damping mechanism was implemented. If the absolute value of  $\delta T$  is larger than the current value of  $T$  the change is limited to a fraction of  $\delta T$ . The value for the fraction can be set via a parameter (typically a value of 0.8 is used). To improve the convergence of the Newton-Raphson iteration an optional Ng-acceleration for the temperature was implemented. For this the already available routine for the Ng-step from the radiative transfer iteration is used (see chapter 6 for convergence properties).

At the moment only the RT-step is parallelized (see Fig. 3.3) for the temperature correction, although the current implementation would fit into the hierarchical parallelization scheme used for the line transfer in PHOENIX/3D (Baron and Hauschildt 2007). In this scheme also the wavelength loop can be parallelized, only a collective MPI operation (sum) for  $F$  and  $F'$  after the wavelength loop is necessary.

The temperature correction scheme presented here is designed for protoplanetary disks, but it may also be applied to other suitable physical problems. However, the scheme is based on the assumptions of LTE and temperature independent opacities which may be inappropriate for some physical problems.

Until now the scheme was only tested for protoplanetary dust disks. The tests and results are described in chapter 6.



# Chapter 6

## Tests and Results

To test the disk radiative transfer and our new temperature correction scheme we have chosen the benchmark test of Pinte et al. (2009). This benchmark test is especially designed for dust radiative transfer in protoplanetary disks. The different test cases include very high optical depths and anisotropic scattering. The results for several Monte Carlo and discrete ordinate codes are freely available and include the temperature structure in the disk, SEDs, scattered light images and polarisation maps.

Here we only use the results for the temperature structure and limit ourselves to the case of *isotropic scattering* as anisotropic scattering is not yet implemented in PHOENIX/3D. For comparison of our results only the results of one Monte Carlo code, MCFOST (Pinte et al. 2006), are used as the differences between the various MC-codes are quite small.

### 6.1 Benchmark Problem

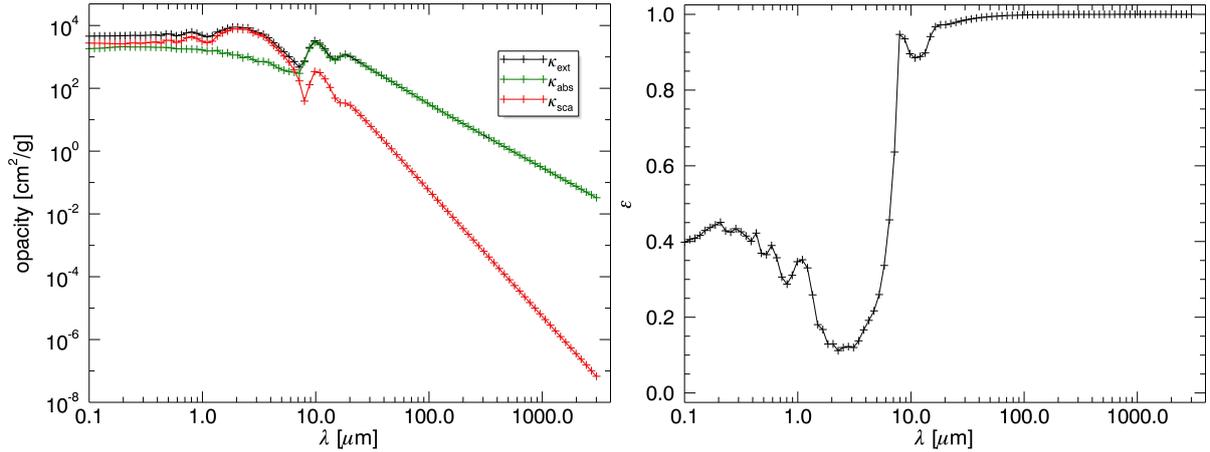
The density structure of the disk is given by the power laws already presented in section 4.1.3. The inner border of the disk is at  $r = 0.1$  AU and the outer border at  $r = 400$  AU. The borders are sharp and the space between the disk and the star and outside of the disk is empty. The exponents for the power laws are  $p_\Sigma = -1.5$  for the surface density ( $r_0 = 100$  AU) and  $p_h = 1.125$  for the scale height ( $h_0 = 10$  AU).

The benchmark test includes four different disk masses,  $3 \cdot 10^{-8}$ ,  $3 \cdot 10^{-7}$ ,  $3 \cdot 10^{-6}$  and  $3 \cdot 10^{-5} M_\odot$ . These masses result in an optical depth in the midplane of the disk (radial from the star to the outer border of the disk) of  $1.22 \cdot 10^3$ ,  $1.22 \cdot 10^4$ ,  $1.22 \cdot 10^5$  and  $1.22 \cdot 10^6$  for the  $0.81 \mu\text{m}$  wavelength.

The star is a spherical black body with a temperature of 4000 K and a radius of  $2 R_\odot$ .

For the moment the opacities are not calculated in PHOENIX/3D; instead, the available tabulated values from the benchmark test are used. The given opacities are for homogeneous, spherical dust grains composed of astronomical silicates (Weingartner and Draine 2001) with a single size of  $1 \mu\text{m}$  and a fixed mass density of  $3.5 \text{ g/cm}^3$ .

The given wavelength grid ranges from  $0.1 \mu\text{m}$  to  $3000 \mu\text{m}$  and the values for  $\kappa_{ext}$  and  $\kappa_{sca}$  are given for 100 logarithmically spaced wavelength points in this range. The optical properties of the dust are presented in Fig. 6.1.



**Figure 6.1** – The optical properties of the dust for the Pinte et al. (2009) benchmark. *Left panel:* The opacity of the dust. Black is the total opacity ( $\kappa_{ext} = \kappa_{abs} + \kappa_{sca}$ ), green is the absorption coefficient ( $\kappa_{abs}$ ) and red is the scattering coefficient ( $\kappa_{sca}$ ). *Right panel:* The thermal coupling parameter  $\epsilon$ . The “+” symbols indicate the points for the wavelength grid.

## 6.2 Results

### 6.2.1 Test Configuration

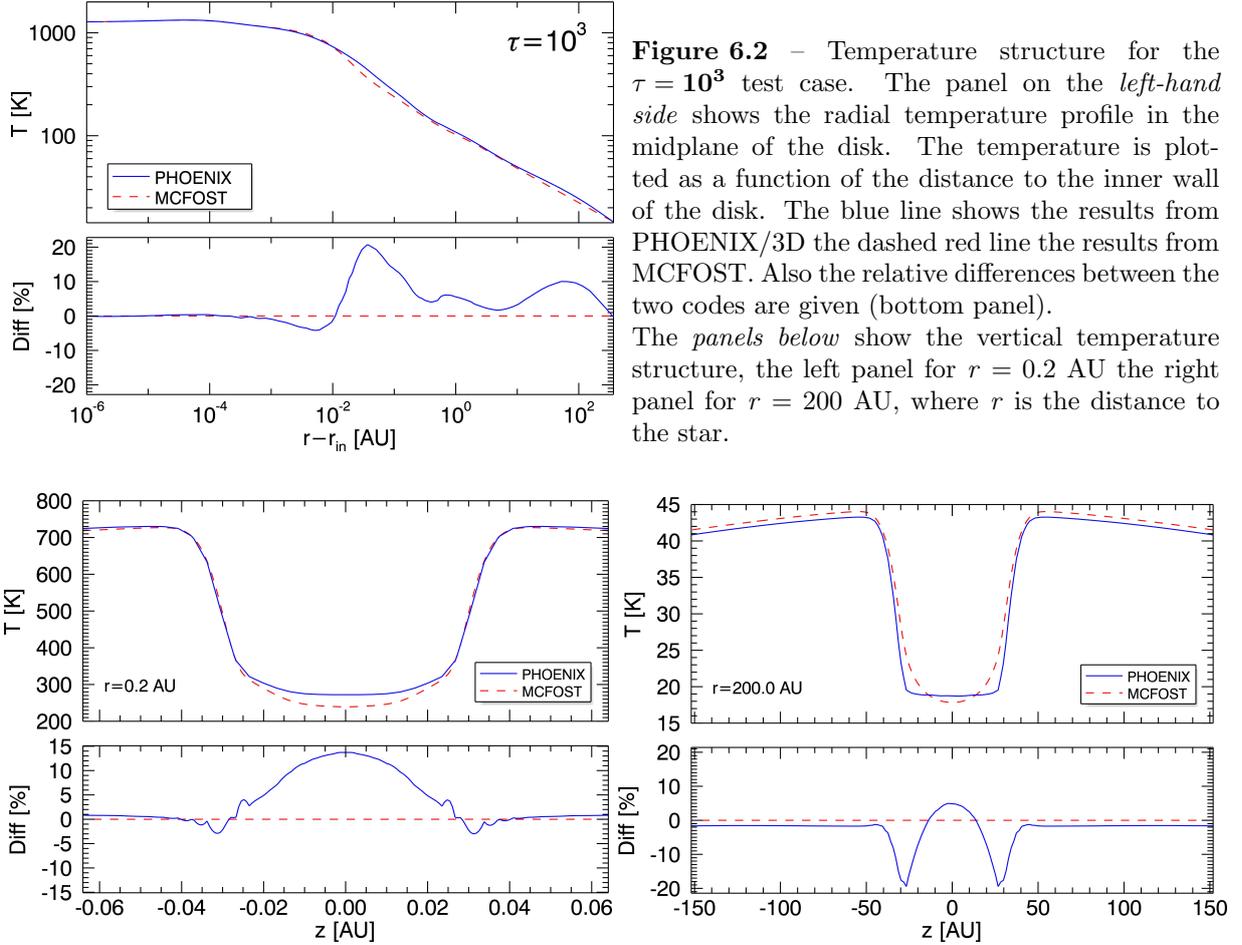
For the test runs a spatial grid with  $nr = 50$ ,  $nt = 60$  and  $np = 4$  and a solid angle grid with  $ntheta = 121$  and  $nphi = 90$  (in total 10,980 rays) is used. The grids are aligned as described in section 4.2. For the spatial  $\theta$ -coordinate the voxels around the poles are replaced by one big voxel resulting in an actual number of  $nt = 50$ . The voxel grid then has a total number of  $101 * 101 * 9 = 91,809$  voxels. Different from the given benchmark problem the outer border of the disk (radial) is not sharp (see section 4.1). The limit for the vertical extension of the disk is set to  $12h(r)$  (scale height), everything else has density zero. For the temperature correction 50 wavelength points are used (every second from the predefined wavelength grid). Tests with 100 wavelength points have shown that the usage of more wavelength points does not significantly affect the results (the deviations are  $< 1\%$ ).

To test the convergence properties a rather strict convergence criterion of a maximum relative change in  $T$  of  $10^{-5}$  is used.

### 6.2.2 Discussion

For simplicity the four different test cases are named  $\tau = 10^3$  to  $\tau = 10^6$ . We use the term low optical depth test case for  $\tau = 10^3$  and  $\tau = 10^4$  and high optical depth test cases for  $\tau = 10^5$  and  $\tau = 10^6$ , respectively. As in the benchmark paper the results are presented for the radial temperature profile in the midplane of the disk, and for the vertical temperature structure at  $r = 0.2$  AU and  $r = 200$  AU, where  $r$  is the distance to the star (see Fig. 6.2, 6.4, 6.5 and 6.6). For the radial temperature structure the temperature is plotted versus the distance to the inner rim, which makes deviations between the codes better visible.

For the discussion of the results we separate the disk into three different regions: the *inner*



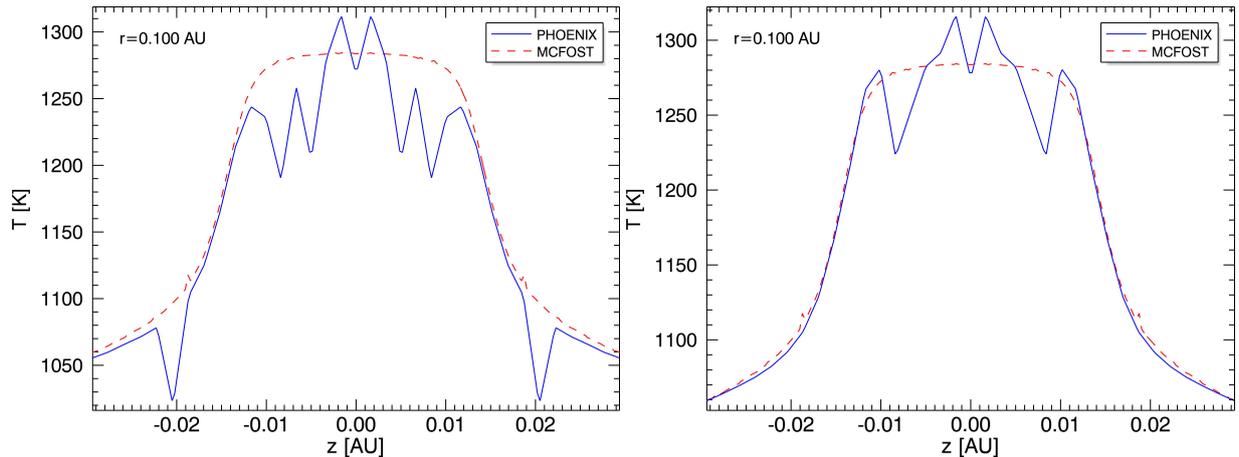
**Figure 6.2** – Temperature structure for the  $\tau = 10^3$  test case. The panel on the *left-hand side* shows the radial temperature profile in the midplane of the disk. The temperature is plotted as a function of the distance to the inner wall of the disk. The blue line shows the results from PHOENIX/3D the dashed red line the results from MCFOST. Also the relative differences between the two codes are given (bottom panel). The *panels below* show the vertical temperature structure, the left panel for  $r = 0.2$  AU the right panel for  $r = 200$  AU, where  $r$  is the distance to the star.

region ranging from  $r_{in}$  to  $r_{in} + 10^{-3}$  AU, the *middle region* (also the most problematic region) ranging from  $r_{in} + 10^{-3}$  AU to  $r_{in} + 10$  AU and the rest we call the *outer region*.

### The Inner Region

For the low optical depth test cases PHOENIX/3D and MCFOST agree well. The relative difference in temperature for the  $\tau = 10^3$  test case are below 1%. However, for increasing optical depth the models agree less well. For the  $\tau = 10^6$  test case PHOENIX/3D yield a temperature that is about 20% lower than that found by MCFOST. This problem is probably caused by the high optical depths in this region for the short wavelength regime. This was verified by a test where only the radiative transfer for one wavelength ( $0.81\mu\text{m}$ ) for a disk with a fixed temperature structure (constant  $B_\lambda$ ) is done. Although the radiative transfer iteration appears to converge, the final values of  $J_\lambda$  are nearly equal to  $B_\lambda$  in the midplane of the inner region. This should not be the case as the inner rim is directly heated by stellar radiation. It seems that the stellar radiation cannot penetrate into this inner region, not even into the first radial voxel (at least for the high optical depth test cases). The reason for this is a too low spatial resolution.

In Fig. 6.3 the vertical temperature structure at  $r = 0.1$  for the  $\tau = 10^4$  test case with  $nr = 50$  (left panel) and  $nr = 60$  (right panel) is shown. Although the result is improved the problem does not completely vanish. So, for the even higher optical depths much more radial points are needed. Probably also a higher number of  $\theta_{vox}$ -points is required



**Figure 6.3** – The inner wall for the  $\tau = 10^4$  test case. The *left panel* shows the results for 101, the *right panel* for 121 radial points. The MCFOST data used for this plot was provided by Woitke (2011, priv. comm.).

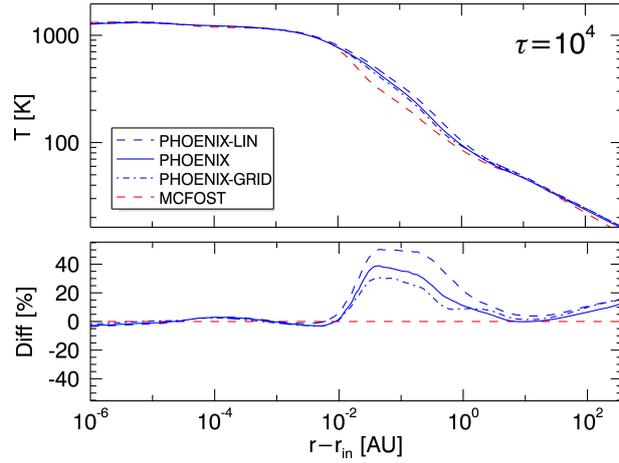
as stellar radiation is also scattered into the midplane from the upper layers of the disk. Some negative effects for the inner region may also be caused by the not completely sharp inner wall (see section 4.1.1). But these effects do not depend on optical depth and can also occur at larger heights above the midplane. The jag in the temperature structure at  $z = \pm 0.02$  AU, visible in Fig. 6.3, also occurs for the  $\tau = 10^3$  test case. Although these jags also depend on the number of radial points, it is more the combination of the radial and  $\theta_{vox}$ -points witch causes this problem. As a rule of thumb a higher number of  $\theta_{vox}$ -points also requires a finer radial grid near the inner rim.

So, the important thing for the inner region is the spatial resolution as the resulting temperature is dominated by the direct irradiation of the inner wall by the star. Especially important is the fineness of the radial grid which can also be controlled via the  $\alpha$ -parameter in equation 4.1. This parameter has to be used with care, as a too high  $\alpha$ -value has a negative impact on the temperature structure of the middle region, as the grid there becomes coarser. For the test cases presented here values of  $\alpha = 3.5$  for  $\tau = 10^3$  and  $\tau = 10^4$  and  $\alpha = 4$  for the higher optical depths are used.

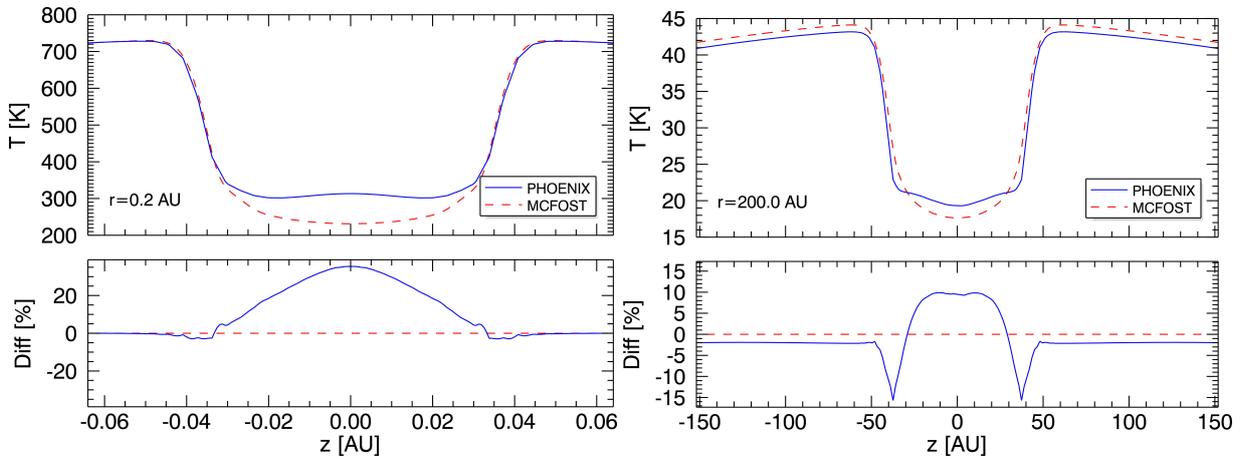
### The Middle Region

In this region the temperature in the midplane drops rapidly as the dust is not heated anymore by the direct radiation of the star. The optical depths in radial direction (from the star to the middle region) but also in vertical direction (from the border of the disk to the midplane) are so high that radiation from the star, even scattered radiation, cannot reach the midplane. The only heating mechanism left for this deep inner region are the dust re-emission from the upper layers of the disk (Pinte et al. 2009).

Monte Carlo RT-codes usually use the diffusion approximation method for this deep region as the temperature structure is defined by the local emission only (e.g. Pinte et al. 2009; Min et al. 2009). This is done mainly for performance reasons, as a full Monte Carlo treatment of this deep region would require a very high number of photon packages. All MC-codes presented in the benchmark paper are using this diffusion approximation. According to Min et al. (2009) the diffusion approximation tends to overestimate the



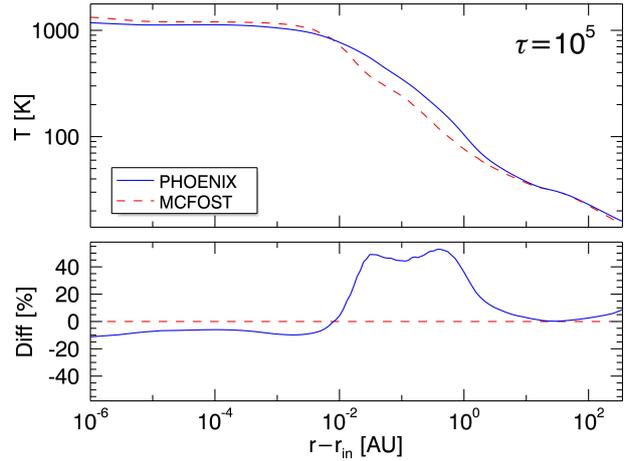
**Figure 6.4** – The same as Fig. 6.2 but for the  $\tau = 10^4$  testcase. For the radial profile also the results of a test run where only linear interpolation was used (*blue dashed line*) and a test run with a larger grid (*blue dash-dotted line*) are included. The *blue solid line* represents the results with parabolic interpolation (the default configuration).



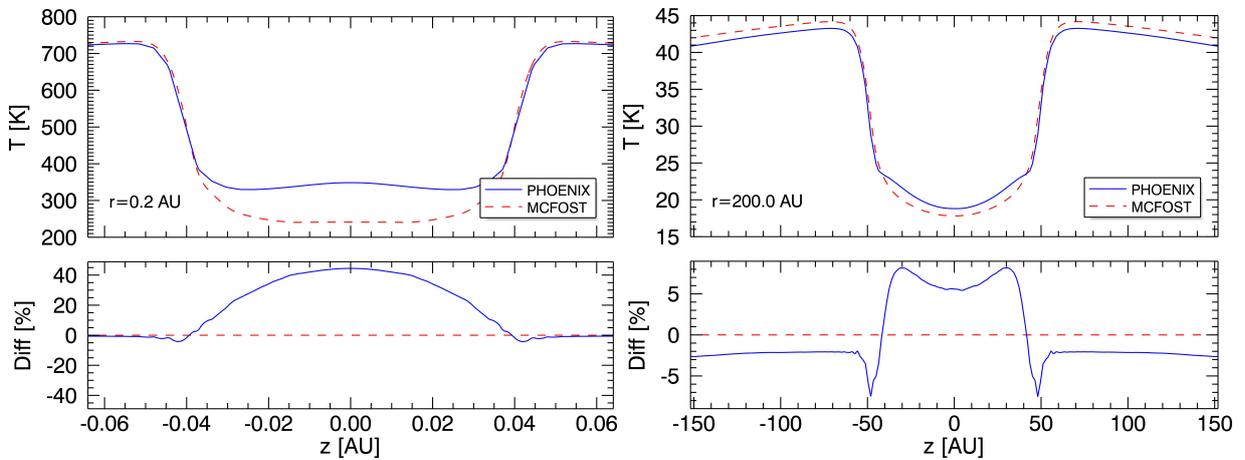
temperature as it prevents cooling by long wavelength radiation which still would be able to exit the deep region.

However, to calculate the correct temperature for this region is not so important at all, at least if one wants to produce observables (e.g. SED) (Woitke 2011, priv. comm.). Emission from this region cannot escape because of the high optical depths. Therefore it does not influence the radiation field which actually is observed. The correct temperature structure becomes important if one wants to calculate the hydrostatic structure of the disk as it influences the scale height of the disk (Min et al. 2009).

As seen from the radial temperature profiles, PHOENIX/3D always overestimates the temperature in the midplane. Compared to MCFOST (with diffusion approximation) the maximum differences are ranging from +20% for  $\tau = 10^3$  up to +65% for  $\tau = 10^6$ . For the lower optical depth test cases of  $\tau = 10^3$  and  $\tau = 10^4$  the maximum error is located between  $r - r_{in} = 10^{-2}$  and  $r - r_{in} = 10^{-1}$  AU. The maximum error for the higher optical depths is located further out at  $r - r_{in} \approx 10^0$  AU. Further out the error drops quite fast as the disk becomes optically thin in the vertical direction, and the dust in the midplane is again heated by scattered light of the star (Pinte et al. 2009). From the vertical profiles for  $r = 0.2$  AU ( $r - r_{in} = 10^{-1}$  AU) it is visible that the error remains in the deep midplane of the disk. In the upper layers where the disk becomes optically thin the results of both codes are nearly identical for all test cases. This indicates that the too high temperature in the midplane is not caused by a too high temperature in the upper layers which are the only remaining heating source for this deep region.



**Figure 6.5** – The same as Fig. 6.2 but for the  $\tau = 10^5$  testcase.



We found that for this region, the result depends on the interpolation method used for the source function during the evaluation of the formal solution (see equation 3.10). The radial profile in Fig. 6.4 shows the results for a test run where only linear interpolation is used in comparison to the test run where both parabolic and linear interpolation is used. This figure shows that parabolic interpolation significantly improves the accuracy of the results, especially in the deep inner regions of the disk. However, it is not possible to use the parabolic interpolation everywhere in the model grid. The parabolic interpolation can cause convergence problems especially in regions with drastic changes of the source function along a characteristic. In radial direction the most problematic region is the transition from the middle region to the outer region. There the disk becomes optically thin in vertical direction and scattered light can reach the midplane again. Another problematic area is the jump of the source function in vertical direction caused by the exponential density profile. This jump becomes also visible in the vertical temperature profiles. In these regions the parabolic interpolation results in negative  $I_\lambda$  and consequently negative values for  $J_\lambda$  and  $S_\lambda$  which is cumbersome for the convergence ratio but may also prevent convergence at all. The negative values for  $I_\lambda$  are a side effect of parabolic interpolation, as the parabolic interpolation can cause local extrema where no extrema should be, and these extrema can even result in negative  $I_\lambda$  (see Auer 2003a). This convergence problem especially applies to the short wavelength range and becomes more severe for the high optical depth test cases ( $\tau = 10^5, 10^6$ ).

As described in section 3.2.1 the interpolation method is automatically chosen depending

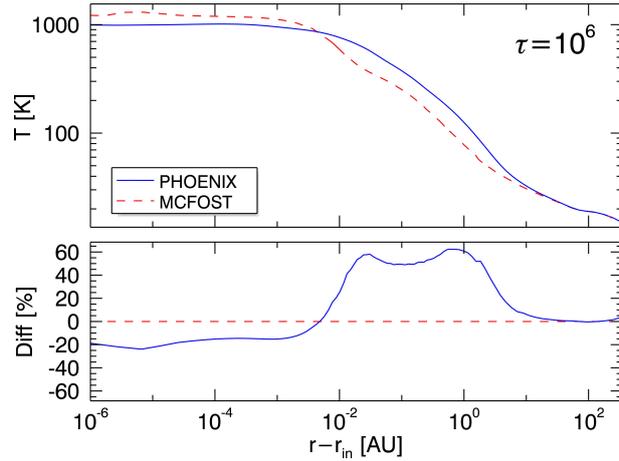
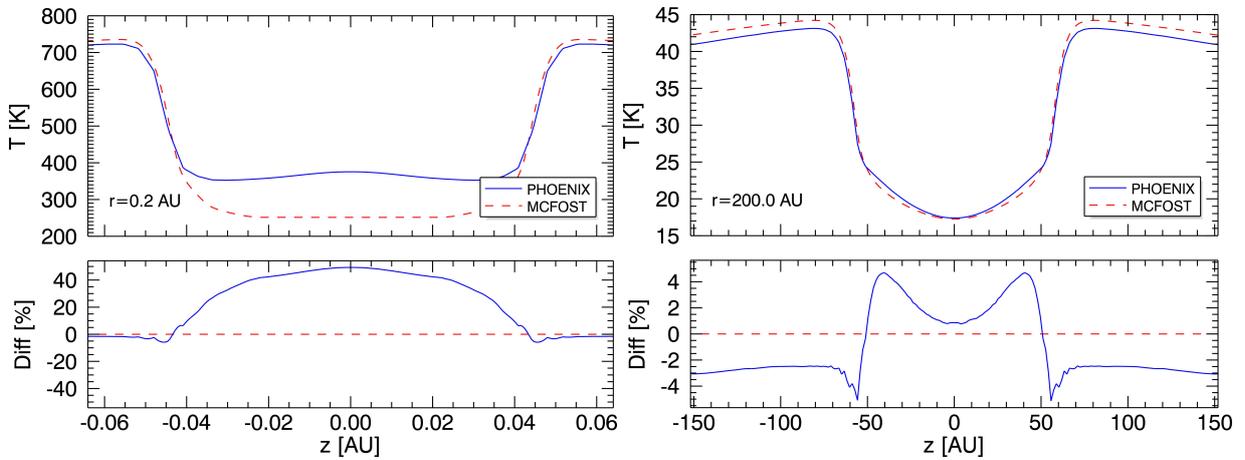


Figure 6.6 – The same as Fig. 6.2 but for the  $\tau = 10^6$  testcase.



on the optical depth for one step ( $\Delta\tau$ ) and the change of the source function along the characteristic via the parameters *taulin* and *PPM\_max\_step*, respectively.

It turned out that the usage of the parameter *PPM\_max\_step* is problematic. As the source function changes between two iterations it can happen that the interpolation method for the same characteristic and same voxel also changes. This just screws up the convergence of the temperature correction scheme. This is caused by the fact that the  $\Lambda^*$ -operator is only calculated in the first iteration and therefore in subsequent iterations it may be the case that for the same voxel and characteristic a different interpolation method for the formal solution ( $J_{FS}$ ) is used as it was used for the  $\Lambda^*$ -operator in the first iteration. However, deactivating *PPM\_max\_step* does not resolve all convergence problems.

As the *PPM\_max\_step* parameter cannot be used, the only selection criterion for the interpolation method is the optical depth between two voxels ( $\Delta\tau$ ). For the high optical depths test cases, still linear interpolation has to be used (which is achieved by a very high value for *taulin*). Also for lower optical depth test cases ( $\tau = 10^3, 10^4$ ) a quite high threshold of  $\approx 10^{-1}$  for the *taulin* parameter is necessary to secure convergence. The convergence problems are mainly caused by the short wavelength range, but in PHOENIX/3D it is not possible to set the *taulin* parameter depending on the wavelength. To make this selection criterion for the interpolation method more flexible, the following adaptations were implemented:

- The wavelength range can be separated into a short and a long wavelength range. The boundary between these two ranges can be set via an additional parameter (typical values are between 1  $\mu\text{m}$  and 2  $\mu\text{m}$  for the benchmark test). By this the *taulin*-parameter for the long wavelength range can be set to a lower value and therefore the parabolic interpolation can be used more often.
- An additional *taulin* parameter can be set for steps where only the radial voxel index changes. For these radial steps a lower *taulin*-value can typically be used which increases the number of parabolic interpolations for the radial direction. This is possible as usually a relatively fine radial grid is used. This parameter does not depend on wavelength.
- Further it turned out that for the high optical depth test cases the parabolic interpolation causes problems (negative  $J_\lambda$ ) if the interpolation is done between voxels with different  $\phi$ -coordinate indices. This especially happens in the inner region of the disk near the midplane. This may be caused by the large dimensions of the voxels in  $\phi$ -direction. To avoid this, linear interpolation is always used if voxels with different  $\phi$ -coordinate indices are involved in the interpolation.

With these adaptations it is now possible to use parabolic interpolation also for the higher optical depth test cases. The best values for the various *taulin*-parameters vary between test cases and were just found by playing around with different parameter values. Usually the higher optical depth test cases require higher values for the *taulin* parameters. However, these adaptations secure that for the inner and middle region near the midplane mostly parabolic interpolation is used (for the longest wavelengths this may not be always the case as the opacities for them are very low). For the results presented here, the adaptations described above were already used, and it seems that even the parabolic interpolation method is too inaccurate for this deep inner region of the disk.

The still too high temperatures are probably a result of numerical diffusion. In radial direction the values for  $\Delta\tau$  (equation 3.9) become very large in the midplane of the middle region (e.g. up to  $\Delta\tau_{0.81\mu\text{m}} \approx 10^4$  for the  $\tau = 10^6$ ). For these large values of  $\Delta\tau$  the interpolation for the source function integral (equation 3.10) becomes inaccurate. The influence of the neighbouring source functions is too high as the sampling of the source function in “ $\tau$ -space” is too bad.

In Fig. 6.4 also the radial temperature profile for a test run with a larger grid with 131 radial and 125  $\theta$ -points is shown. The result for the deep middle region is improved as the  $\Delta\tau$  values are lower for a larger grid, but it becomes clear that probably an unreasonably large grid would be necessary to solve this problem completely, at least for the high optical depth test cases. Also, increasing the number of spatial  $\phi$ -points does not change the overall picture. Although the results are slightly improved because more characteristics are tracked through the grid, it is in general more efficient to use more radial and/or  $\theta$ -points in terms of accuracy and the needed computational time.

Another possibility would be to adapt the grid to the given optical depths between voxels. Brief tests were done for an adaptive radial grid. The radial dimension of the voxels are adapted to a given maximum value of  $\Delta\tau$ . This secures that the  $\Delta\tau$  value for a step along the radial coordinate between two voxels cannot exceed this maximum value. By this the maximum error in the middle region is reduced without the need for more voxels.

However, this method has its limitations. The given maximum value for  $\Delta\tau$  cannot be reduced at will as the radial voxel dimension in this region would become extremely low which results in unreasonably large voxel sizes for the outer region. This is especially true for the high optical depth test cases. To allow a better comparison between the different test cases the adaptive radial grid was not used for the results presented here.

In Auer (2003a) a method for avoiding local extrema caused by the parabolic interpolation is proposed, which would probably allow to use the parabolic interpolation throughout the disk. Also different interpolation methods like Hermite interpolation are presented. However, the methods presented in Auer (2003a) are for the short characteristic method and cannot be directly applied to the long characteristic method used in PHOENIX/3D. It also has to be considered that a change in the interpolation method influences the construction of the  $\Lambda^*$ -operator. However, improving the interpolation method seems to be necessary, but further investigations on this topic are needed.

### The Outer Region

In this region the disk becomes optically thin in vertical direction even for the shorter wavelengths. The temperature near the midplane is mainly determined by the stellar radiation which is scattered into the midplane. The results for the outer region are best compared via the vertical temperature profiles at  $r = 200$  AU. Differently to the already discussed regions the results are now better for the high optical depth test cases. For  $\tau = 10^6$  (Fig. 6.6) the differences to MCFOST, for the vertical profile at  $r = 200$  AU, are below 5%, which is comparable to the deviations between the different Monte Carlo codes presented in the benchmark paper.

For the low optical depth test cases the differences are larger (up to 20%) and also a somewhat unnaturally looking boxy vertical temperature profile is visible. The reason for this is probably again the interpolation method. For the high optical depths test cases the  $\Delta\tau$ -values in radial direction near the midplane are still typically  $\gtrsim 1$  for the short wavelength range and therefore parabolic interpolation is used. For the low optical depth test cases mainly linear interpolation is used as the  $\Delta\tau$ -values are typically below the threshold for the *taulin*-parameter. By using lower values for the *taulin*-parameter the boxy shape of the vertical temperature profile can be weakened somewhat. But we are limited in the range for *taulin* as a too low value causes convergence problems especially in the regions already discussed above.

Another cause for these problem may be the angular resolution, but as already a quite high number of rays (10,890) is used this is probably not the case. Testing of the dependency of the results on the angular resolution is not really possible as the the angular and spatial grid are aligned, and an increase of the angular points always requires a larger grid at least for the  $\theta$ -coordinate. What can be tested is a higher number of solid angle points for  $\phi$ , but no significant improvement was found.

Similar to the *middle region* an improvement of the interpolation seems to be necessary, at least for the low optical depth test cases. Differently to the middle region the results in the outer region are not significantly improved by using a somewhat larger grid (see Fig. 6.4). Another method to increase the accuracy is to use more characteristics. Tests with additional characteristics which are launched around the center of the model space were quite

successful concerning the boxy temperature profile. Roughly 100 additional equidistantly distributed characteristics for each ray (direction) are launched in an area with a radius of 0.1 AU around the center. In principal it is possible with this method to correct the boxy temperature profile, but its application is quite tricky. The best choice for the number of additional characteristics and also their distribution depends on the size of the grid. Further, this method also causes some artefacts in the temperature profile in the inner region of the disk, probably caused by the arbitrary choice for the size of the “launching area” (a much larger area becomes quite computational expensive). Therefore this method was not used for the test runs presented here. However, it may be worth to further investigate this approach as it can solve the problems for the optically thin outer region.

### 6.2.3 Symmetry Test

As for the benchmark test the disk is azimuthally symmetric and axisymmetric in  $z$  (in spherical coordinates in  $\theta$ ), it has to be tested if this is still true for the final temperature structure.

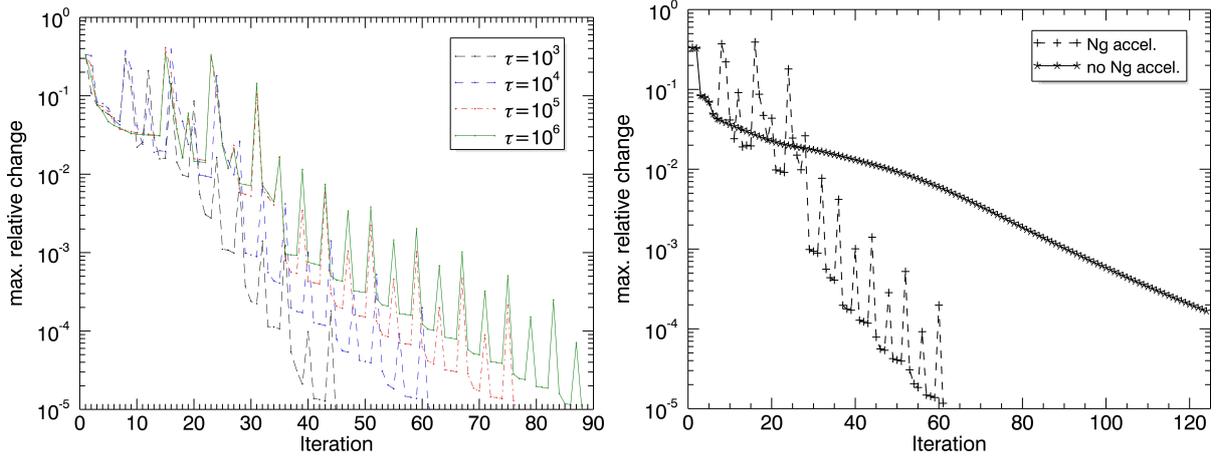
Actually the resulting asymmetries depends on the algorithm which secures that each voxel is hit at least once for each angular point (ray). This is done by going through the three dimensional voxel grid (array), check if the voxel is already hit, and if not, a new characteristic is launched. The innermost loop for this stepping is for the radial coordinates, the outermost for the  $\phi$ -coordinate, and the starting voxel is  $(-nr, -nt, -np)$ . But this algorithm can cause some troubles for the symmetry, at least for small grids or a low number of rays. The symmetry of the results was improved by choosing the starting voxel in dependence of the actual direction of the ray. For the  $\theta_{vox}$ -coordinate the starting voxel is  $-nt$  ( $\theta_{vox} = 0$ ) if  $\theta_{ray} \leq \pi/2$  and  $nt$  ( $\theta_{vox} = \pi$ ) if  $\theta_{ray} > \pi/2$ . For the  $\phi_{vox}$ -coordinate the starting index is chosen by searching for the  $\phi_{vox}$ -coordinate which is the closest to  $\phi_{ray}$ . This just improves the distribution of the launched characteristics for one ray.

For the tests presented here the deviation between symmetric voxels are always below 0.5% for the  $\theta_{vox}$ -coordinate and 0.06% for the  $\phi_{vox}$ -coordinate. The deviations for the default algorithm are only slightly worse for the  $\theta_{vox}$ -coordinate but roughly two times higher for the  $\phi_{vox}$ -coordinate. For the convergence rate and the stability no significant differences between the two algorithms were found.

## 6.3 Convergence and Performance

### 6.3.1 Convergence

In Fig. 6.7 the convergence properties of the temperature correction scheme are shown. The used convergence criterion is: *max relative change*  $\frac{\Delta T}{T} < 10^{-5}$ . The convergence criterion is reached for all test cases. The  $\tau = 10^3$  test case needed 45 the  $\tau = 10^6$  test case 88 iterations. The Ng-acceleration works properly for all test cases but it is beneficial to start the Ng-acceleration not too early as this can adversely influence the stability, resulting in negative values of  $J_\lambda$  and  $S_\lambda$ . The right panel of Fig. 6.7 shows the



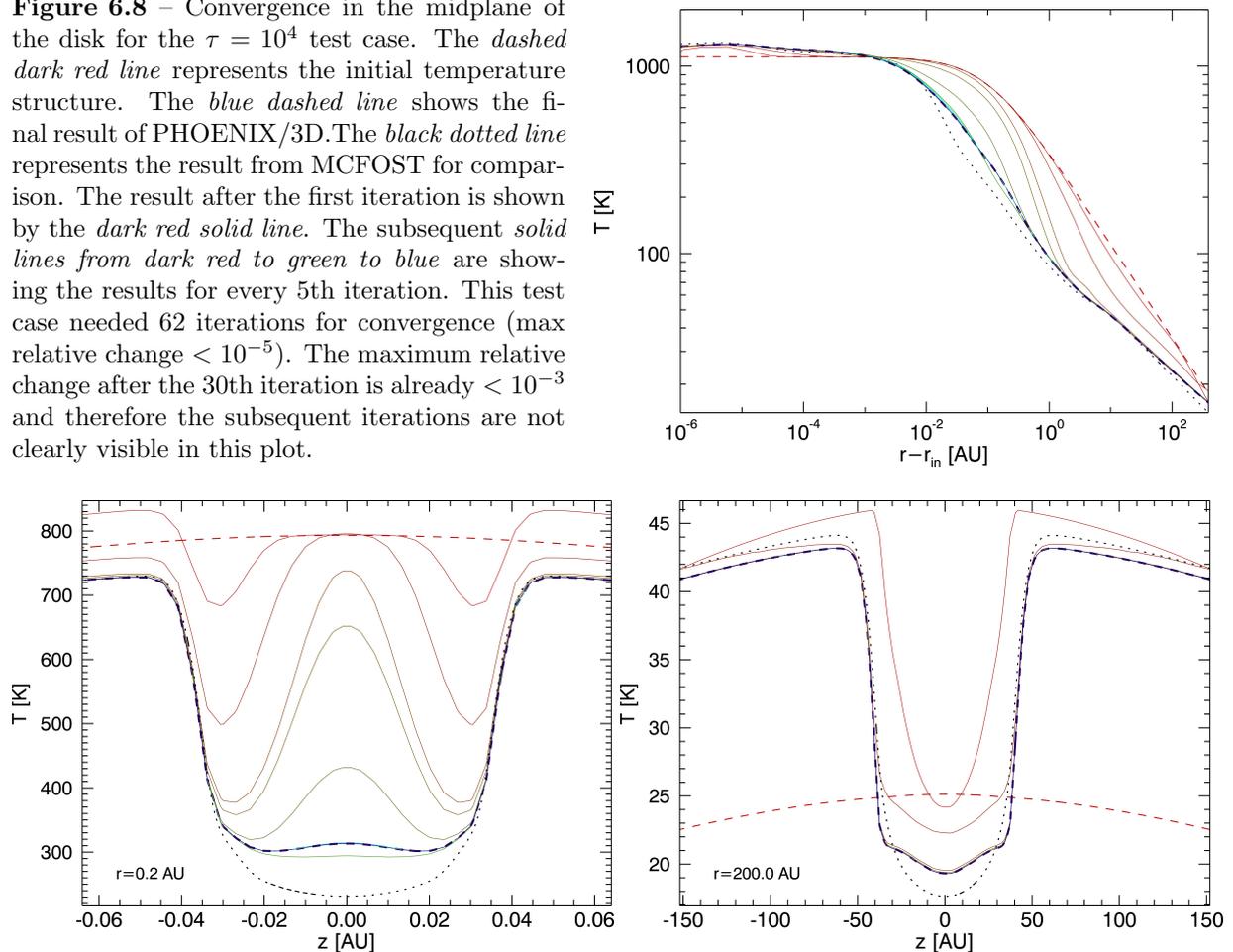
**Figure 6.7** – Convergence properties of the temperature correction scheme. The x-axis shows the number of iterations, the y-axis the maximum relative change in temperature. The *left panel* shows the convergence rates for the different test cases presented in section 6.2. For the test cases  $\tau = 10^5$  and  $\tau = 10^6$  the Ng-acceleration was only started after the 15th iteration as this improves the stability. For the lower optical depth test cases Ng-acceleration was already started after the 8th iteration. The *right panel* shows the convergence rate for the  $\tau = 10^4$  test case with (*dashed line*) and without (*solid line*) Ng-acceleration.

convergence rate for the  $\tau = 10^4$  test case with the use of Ng-acceleration and without. The excellent efficiency of the Ng-acceleration is clearly visible. The test run without Ng-acceleration was already stopped after the 124 iteration where the maximum relative change still is  $> 10^{-4}$ .

Fig. 6.8 and Fig. 6.9 display the history of the convergence for the midplane and in vertical direction, respectively. These figures allow to study the convergence rates in different regions of the disk. The middle region shows clearly the slowest convergence rate, whereas for the inner and outer region convergence is already achieved after  $\approx 15$  iterations. After the 30th iteration the maximum relative change is already  $< 10^{-3}$  and the subsequent iterations do not significantly improve the results anymore. So, a less strict convergence criterion may be used. This reduces the number of iterations and consequently the computational time by a factor of  $\approx 2$ . This shows also that the inaccuracy of our results is not caused by bad convergence. However, it must be considered that the inaccuracy of the results may influence the convergence rate. The too high temperature in the midplane of the middle region may positively influence the convergence rate.

Also a dependency between the convergence rate and the resolution of the grid was found. For a larger grid more iterations are needed. This dependency is similar to that presented in Hauschildt and Baron (2006) for pure radiative transfer tests. This is caused by the lower optical depths between voxels (lower  $\Delta\tau$ ) for a larger grid. For lower  $\Delta\tau$ -values the coupling between voxels is stronger which increases the number of iterations (Hauschildt and Baron 2006). So we have to consider that a finer grid, which improves the results, slows down the convergence. On the other hand a grid with higher spatial resolution allows a more extended use of the parabolic interpolation which in general improves the convergence rate. For the  $\tau = 10^4$  test case (see Fig. 6.4) 10 more iterations are needed for the larger grid. However, for the larger grid also lower values for the *taulin*-parameter were used so the test cases with a smaller and larger grid are not directly comparable.

**Figure 6.8** – Convergence in the midplane of the disk for the  $\tau = 10^4$  test case. The *dashed dark red line* represents the initial temperature structure. The *blue dashed line* shows the final result of PHOENIX/3D. The *black dotted line* represents the result from MCFOST for comparison. The result after the first iteration is shown by the *dark red solid line*. The subsequent *solid lines from dark red to green to blue* are showing the results for every 5th iteration. This test case needed 62 iterations for convergence (max relative change  $< 10^{-5}$ ). The maximum relative change after the 30th iteration is already  $< 10^{-3}$  and therefore the subsequent iterations are not clearly visible in this plot.



**Figure 6.9** – The same as Fig 6.8 but for the vertical direction at  $r = 0.2$  AU (*left panel*) and  $r = 200$  AU (*right panel*). For  $r = 0.2$  AU no significant changes after the 30th iteration are visible. It is also visible that the convergence rate in the midplane ( $z = 0$ ) is the slowest. For  $r = 200$  AU the “final” result is already reached after  $\approx 10$  iterations.

## Initial Conditions

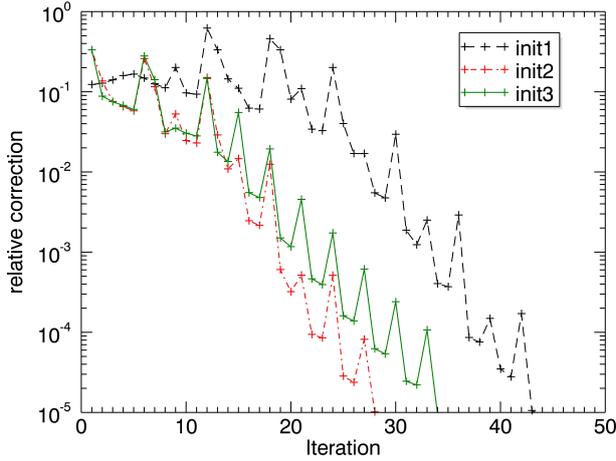
To test the dependency of the convergence on the initial conditions and to check that the results do not depend on them, three different initial conditions for the dust temperature were tested.

The first one (*init1*) is simply a constant temperature  $T = 1500$  K throughout the disk. The two others are taken from Dullemond and Monnier (2010) and are given by equation 6.1 (*init2*) and equation 6.2 (*init3*). These two equations represent approximations to accurate radiative transfer results.

$$T_d = T_* \frac{1}{\epsilon^{0.25}} \sqrt{\frac{R_*}{2r}} \quad (6.1)$$

$$T_d(\tau) = T_* \sqrt{\frac{R_*}{2r}} \left[ \sin(\theta)(2 + 3(\sin \theta)\epsilon) + \left( \frac{1}{\epsilon} - 3\epsilon \sin(\theta)^2 \right) e^{-\frac{\tau}{\sin(\theta)\epsilon}} \right] \quad (6.2)$$

where  $T_d$  is the dust temperature,  $T_*$  the temperature of the star,  $R_*$  the radius of the star,  $r$  the distance to the star,  $\epsilon$  is an efficiency parameter for the cooling of the dust,  $\theta$



**Figure 6.10** – Convergence for three different initial conditions. *init1* (black dashed line) represents the convergence for a constant temperature  $T = 1500$  K, *init2* (red dash-dotted line) for the initial conditions according to equation 6.1, *init3* (green solid line) for the initial conditions according to equation 6.2.

is the angle under which the radiation from the star hits the corresponding voxel and  $\tau$  is the radial optical depth from the star to the voxel.  $\epsilon$  is a correction factor for the cooling of the dust and depends on the size of the dust grains. For the test case presented here with a dust size of  $1 \mu\text{m}$  a value of 0.35 is given in Dullemond and Monnier (2010). The optical depth is calculated for the  $1.8 \mu\text{m}$  wavelength. This is the wavelength at which the Planck function peaks for a value of  $T = 1500$  K.

The convergence rate for a test case with  $\tau = 10^3$  are presented in Fig. 6.10. The test case is converging for all three initial conditions, but surprisingly the fastest convergence is reached for initial condition *init2*. One would expect a better convergence rate for initial condition *init3* as there also the vertical structure is considered. However, for the test runs presented in the previous section initial condition *init2* was used. The final results for the temperature structure are identical for all three initial conditions.

### 6.3.2 Performance

For the four test runs presented above  $64 \times 8 = 512$  CPUs (Intel Xeon Gainestown X5570 or Intel Xeon Harpertown E5472) on the super cluster of the *Höchstleistungsrechenzentrum Nord (HLRN)* were used. The  $\tau = 10^3$  test case needed 4.2 h the  $\tau = 10^6$  test case 7.8 h. One temperature correction iteration took  $\approx 5.2$  m, where one wavelength point needed  $\approx 6$  s. The time for the single iteration steps are the same for all test cases.

The most time expensive part for one wavelength point is the RT-step, only at maximum 5% of the computational time is needed for the additional temperature correction code. The solution for the Newton Raphson-Equation (5.13) only took  $\approx 1$  s. So  $\approx 95\%$  of the computational time needed for one temperature correction iteration is caused by the RT-steps.

The memory consumption for the above test runs is  $\approx 10\text{GB}$  for one node with 8 CPUs. Most of the memory is needed for storing the  $\Lambda^*$ -operator for each wavelength for subsequent iterations (to gain performance). If memory consumption becomes a problem it is also possible to newly construct the  $\Lambda^*$ -operator for each iteration and wavelength, so only one  $\Lambda^*$ -operator is stored at the same time. Although this reduces the memory consumption the computational time increases by a factor of  $\approx 1.5$ . Further we want to note that for the disk configuration used here nearly 50% of the voxels in the spherical

model space are in vacuum and for these voxels no  $\Lambda^*$ -operator is created (this was already implemented in PHOENIX/3D).

These vacuum voxels can also be ignored for the temperature correction as they have no temperature at all. So these voxels are not considered in the algorithm which secures that each voxel is hit at least once for each ray. The performance gain of this method is roughly 30%. However, this also has an influence on the results as there are fewer characteristics tracked through the grid. Tests have shown that it is more efficient, in terms of performance and accuracy of the results, to use a larger grid (e.g. more radial points) than to track more characteristics through the grid by including the vacuum voxels.

One another interesting observation was that the additional RT-step for the star correction (see section 4.2) is only 10% faster than a normal RT-step. This is a bit surprising as for the star correction RT-step no intensities are calculated at all. This implies that most of the time need for one RT-step is caused by just tracking the characteristic through the grid. The reason for this is probably that the step-sizes to reach the next voxels have to be adapted to the different voxel volumes which are inherent to a spherical grid (Hauschildt and Baron 2009). It may be that the highly irregular grid in the radial and  $\theta$ -coordinate, used for the disk configuration, additionally reduces the performance of the adaptive step-size control. No further analysis of this phenomenon was done, but an adjustment of the step-size control algorithm to the disk configuration may be a possibility to improve the overall performance. Further it may be worth to think about a method to speed up the tracking through the vacuum part of the model space. As already mentioned nearly 50% of the voxels are empty and therefore a lot of unnecessary stepping through empty space is done. However, to find an efficient method for this further investigations and discussions with the main developers of PHOENIX/3D are required.

# Chapter 7

## Conclusions and Outlook

As somewhat expected it was not possible to directly apply PHOENIX/3D to the configuration of a protoplanetary disk. The first problem which had to be solved was the angular resolution problem. The problem arises from the small-scale structures like the star and the inner rim (small in comparison to the typical dimension of a protoplanetary disk). For the star the problem could be solved by correcting the weights for the integration of  $J$ . This is not possible for the inner rim, because the actual spatial dimensions of the inner rim are not as well defined as for the star. The problem caused by the inner rim are drastic peaks in both the vertical profile of  $J_\lambda$  and consequently the temperature. The solution for this is to align the angular and the spatial grid for the  $\theta$ -coordinate, so that the discrete points for the angular and the spatial grid have the same distribution. The drawback of this is that we lose the flexibility to adapt the spatial  $\theta$ -grid to the given physical conditions. However, with this two adaptations in the radiative transfer part of PHOENIX/3D it is now possible to solve the radiative transfer equation for the configuration of a typical protoplanetary disk.

To estimate the temperature structure of a protoplanetary disk a new method for solving the radiative equilibrium equation in combination with the radiative transfer equation was implemented. This method is based on the approximate  $\Lambda$ -operator technique already used in PHOENIX/3D. The approximate  $\Lambda$ -operator is used for solving the radiative equilibrium equation via a Newton-Raphson iteration. To test the new temperature correction scheme the benchmark test of Pinte et al. (2009) was used. The method works well and shows quite good convergence properties (it converges for all 4 test cases of the benchmark). The results are satisfactory for larger areas of the disk, but there are still open problems concerning the accuracy of the results, especially in the deep inner region of the disk where no stellar radiation can penetrate. The maximum deviations in this region, compared to the results of the benchmark paper, are  $\approx 20\%$  for the  $\tau = 10^3$  test case and  $\approx 65\%$  for the highest optical depth  $\tau = 10^6$ . This is probably caused by the inaccuracy of the interpolation method used in PHOENIX/3D for the radiative transfer. At the moment one can only overcome this problem by using large spatial grids. However, this becomes unreasonable for very massive disks with high optical depths. So, further investigations concerning this topic are needed.

The computational requirements for the temperature correction scheme are quite high. For the test cases presented in this work 512 CPUs were used. The typical computational

---

time needed for the test runs ranges from 4 to 6h (depending on the optical depth of the disk). The most expensive part in terms of computational time is the pure radiative transfer part as usually large spatial grids and also a large number of rays are necessary, but further performance improvements may be possible. At the moment the symmetry of a protoplanetary disk is not considered. All calculations are still done in full 3D. For typical applications a full 3D treatment is not necessary and performance can be gained by taking advantage of the symmetric structure of the disk (e.g. only consider a “quarter of a disk”). Further performance improvements may be possible by adapting the tracking algorithm for the characteristics to the actual requirements of disk radiative transfer (e.g. the usage of a spherical coordinate system for the disk model results in a lot of empty space which actually could be ignored completely). However, these possible performance improvements require further (extensive) programming efforts.

Besides the above-mentioned necessary improvements, further work is needed to use the new PHOENIX/3D dust disk model for “real world” applications. First of all, the construction of a spectral energy distribution (SED) and scattering images still has to be implemented. The ray tracing capability needed for this is already available in PHOENIX/3D, but for the actual construction of the SED and scattering images some further mostly technical work is necessary. Further, at the moment the dust opacities are not estimated in PHOENIX/3D, the predefined opacities of the benchmark test were used instead. Some already available routines in PHOENIX/3D for e.g. the Mie-theory probably can be used for this.

To test the reliability of the PHOENIX/3D dust radiative transfer and the new temperature correction scheme further tests are necessary. For this other benchmark problems for dust radiative transfer like presented in Pascucci et al. (2004) and Ivezić et al. (1997) or examples for disk models from the literature can be used. Further the full 3D capability of the PHOENIX/3D disk model is still untested. Possible applications for the full 3D disk model are azimuthally asymmetric disk structures like density waves caused by a forming planet (Ogilvie and Lubow 2002; Williams and Cieza 2011).

# References

- André, P., Ward-Thompson, D., and Barsony, M. (1993). Submillimeter continuum observations of Rho Ophiuchi A - The candidate protostar VLA 1623 and prestellar clumps. *The Astrophysical Journal*, 406:122–141.
- Armitage, P. J. (2011). Dynamics of Protoplanetary Disks. *Annual Reviews Astronomy and Astrophysics*, 49:195–236.
- Auer, L. (2003a). Formal Solution: EXPLICIT Answers. In I. Hubeny, D. Mihalas, & K. Werner, editor, *Stellar Atmosphere Modeling*, volume 288 of *Astronomical Society of the Pacific Conference Series*, page 3.
- Auer, L. (2003b). Insight into multi-dimensional transfer. In Ivan Hubeny, Dimitri Mihalas, a. K., editor, *Stellar Atmosphere Modeling, ASP Conference Proceedings*, volume 288.
- Baron, E. and Hauschildt, P. H. (2007). A 3D radiative transfer framework. II. Line transfer problems. *Astronomy & Astrophysics*, 468:255–261.
- Bruderer, S., van Dishoeck, E. F., Doty, S. D., and Herczeg, G. J. (2012). The warm gas atmosphere of the HD 100546 disk seen by Herschel (Evidence of a gas-rich, carbon-poor atmosphere?). *ArXiv e-prints*.
- Dullemond, C. P. and Dominik, C. (2004). Flaring vs. self-shadowed disks: The SEDs of Herbig Ae/Be stars. *Astronomy & Astrophysics*, 417:159–168.
- Dullemond, C. P., Hollenbach, D., Kamp, I., and D’Alessio, P. (2007). Models of the Structure and Evolution of Protoplanetary Disks. *Protostars and Planets V*, pages 555–572.
- Dullemond, C. P. and Monnier, J. D. (2010). The Inner Regions of Protoplanetary Disks. *Annual Reviews Astronomy and Astrophysics*, 48:205–239.
- Dullemond, C. P.; Turolla, R. (2000). An efficient algorithm for two-dimensional radiative transfer in axisymmetric circumstellar envelopes and disks. *Astronomy & Astrophysics*, 360:1187–1202.
- Glassgold, A. E., Najita, J., and Igea, J. (2004). Heating Protoplanetary Disk Atmospheres. *The Astrophysical Journal*, 615:972–990.
- Gorti, U. and Hollenbach, D. (2008). Line Emission from Gas in Optically Thick Dust Disks around Young Stars. *The Astrophysical Journal*, 683:287–303.
- Greene, T. P., Wilking, B. A., Andre, P., Young, E. T., and Lada, C. J. (1994). Further mid-infrared study of the rho Ophiuchi cloud young stellar population: Luminosities and masses of pre-main-sequence stars. *The Astrophysical Journal*, 434:614–626.
- Hamann, W.-R. (1987). *Numerical Radiative Transfer*, chapter 2, pages 35–67. Cambridge University Press.

- Hauschildt, P. H. and Baron, E. (2006). A 3D radiative transfer framework. I. Non-local operator splitting and continuum scattering problems. *Astronomy & Astrophysics* , 451:273–284.
- Hauschildt, P. H. and Baron, E. (2009). A 3D radiative transfer framework. IV. Spherical and cylindrical coordinate systems. *Astronomy & Astrophysics* , 498:981–985.
- Hogerheijde, M. R. (1998). *The molecular environment of low-mass protostars*. PhD thesis, Leiden University.
- Hügelmeier, S. D. (2009). *Multi-dimensional radiative transfer in circumstellar disks*. PhD thesis, Georg-August-Universität zu Göttingen.
- Ivezic, Z., Groenewegen, M. A. T., Men'shchikov, A., and Szczerba, R. (1997). Benchmark problems for dust radiative transfer. *Monthly Notices of the Royal Astronomical Society* , 291:121–124.
- Knop, S. (2007). *General Relativistic Radiative Transfer*. PhD thesis, University of Hamburg.
- Lada, C. J. (1987). Star formation - From OB associations to protostars. In M. Peimbert & J. Jugaku, editor, *Star Forming Regions*, volume 115 of *IAU Symposium*, pages 1–17.
- Mihalas, D. (1978). *Stellar Atmospheres*. W.H. Freeman and Company, second edition.
- Mihalas, D., Kunasz, P. B., and Hummer, D. G. (1975). Solution of the comoving frame equation of transfer in spherically symmetric flows. I - Computational method for equivalent-two-level-atom source functions. *The Astrophysical Journal* , 202:465–489.
- Min, M., Dullemond, C. P., Dominik, C., de Koter, A., and Hovenier, J. W. (2009). Radiative transfer in very optically thick circumstellar disks. *Astronomy & Astrophysics* , 497:155–166.
- Ng, K.-C. (1974). Hypernetted chain solutions for the classical one-component plasma up to Gamma equals 7000. *Journal of Chemical Physics* , 61:2680–2689.
- Ogilvie, G. I. and Lubow, S. H. (2002). On the wake generated by a planet in a disc. *Monthly Notices of the Royal Astronomical Society* , 330:950–954.
- Olson, G. L., Auer, L. H., and Buchler, J. R. (1986). A rapidly convergent iterative solution of the non-LTE line radiation transfer problem. *Journal of Quantitative Spectroscopy and Radiative Transfer* , 35:431–442.
- Olson, G. L. and Kunasz, P. B. (1987). Short characteristic solution of the non-LTE transfer problem by operator perturbation. I. The one-dimensional planar slab. *Journal of Quantitative Spectroscopy and Radiative Transfer* , 38:325–336.
- Pascucci, I., Wolf, S., Steinacker, J., Dullemond, C. P., Henning, T., Niccolini, G., Woitke, P., and Lopez, B. (2004). The 2D continuum radiative transfer problem. Benchmark results for disk configurations. *Astronomy & Astrophysics* , 417:793–805.
- Pinte, C., Harries, T. J., Min, M., Watson, A. M., Dullemond, C. P., Woitke, P., Ménard, F., and Durán-Rojas, M. C. (2009). Benchmark problems for continuum radiative transfer. high optical depths, anisotropic scattering, and polarisation. *Astronomy and Astrophysics*, 498:967–980.
- Pinte, C., Ménard, F., Duchêne, G., and Bastien, P. (2006). Monte Carlo radiative transfer in protoplanetary disks. *Astronomy & Astrophysics* , 459:797–804.
- Pringle, J. E. (1981). Accretion discs in astrophysics. *Annual Reviews Astronomy and Astrophysics*, 19:137–162.

- Rutten, R. J. (2003). *Radiative Transfer in Stellar Atmospheres*. Utrecht University lecture notes, 8th edition.
- Shu, F. H. (1991). *The Physics of Astrophysics - Radiation*, volume one. University Science Books.
- Shu, F. H., Adams, F. C., and Lizano, S. (1987). Star formation in molecular clouds - Observation and theory. *Annual Reviews Astronomy and Astrophysics*, 25:23–81.
- Watson, A. M., Stapelfeldt, K. R., Wood, K., and Ménard, F. (2007). Multiwavelength Imaging of Young Stellar Object Disks: Toward an Understanding of Disk Structure and Dust Evolution. *Protostars and Planets V*, pages 523–538.
- Weingartner, J. C. and Draine, B. T. (2001). Dust Grain-Size Distributions and Extinction in the Milky Way, Large Magellanic Cloud, and Small Magellanic Cloud. *The Astrophysical Journal*, 548:296–309.
- Whitney, B. A., Wood, K., Bjorkman, J. E., and Cohen, M. (2003). Two-dimensional Radiative Transfer in Protostellar Envelopes. II. An Evolutionary Sequence. *The Astrophysical Journal*, 598:1079–1099.
- Williams, J. P. and Cieza, L. A. (2011). Protoplanetary Disks and Their Evolution. *Annual Reviews Astronomy and Astrophysics*, 49:67–117.
- Woitke, P. (2011). Determining the dust temperature with the approximate lambda operator. private communication.
- Woitke, P., Kamp, I., and Thi, W. (2009). Radiation thermo-chemical models of protoplanetary disks. I. Hydrostatic disk structure and inner rim. *Astronomy & Astrophysics*, 501:383–406.
- Wolf, S. (2003). MC3D-3D continuum radiative transfer, Version 2. *Computer Physics Communications*, 150:99–115.



# List of Figures

1.1	Low mass star formation. Credit/Source: Hogerheijde (1998) . . . . .	2
1.2	Models for YSOs according to their observational classes. Credit/Source: Whitney et al. (2003) . . . . .	3
1.3	Sketch of a typical protoplanetary disk. Credit/Source: Dullemond and Monnier (2010) . . . . .	4
1.4	Dust and gas structure of flared protoplanetary disks. Credit/Source: Dullemond et al. (2007) . . . . .	5
2.1	Specific intensity . . . . .	7
3.1	Nassi–Shneiderman diagram for the main steps of the ALI in PHOENIX/3D. . .	18
3.2	Sketch for the long characteristics method. . . . .	20
3.3	Nassi–Shneiderman diagram for the long characteristic solver. . . . .	21
3.4	Convergence properties of PHOENIX/3D. Credit/Source: Hauschildt and Baron (2006) . . . . .	22
4.1	The spatial grid. . . . .	26
4.2	Density structure of the disk . . . . .	28
4.3	Example for the distribution of the characteristics in spherical coordinates. . . .	30
4.4	Test case for the star correction. . . . .	32
4.5	The peaks in $J$ . . . . .	33
5.1	Nassi–Shneiderman diagram for the temperature correction scheme. . . . .	38
6.1	Optical properties of the dust. . . . .	42
6.2	Temperature structure for the $\tau = 10^3$ test case. . . . .	43
6.3	The inner wall for $\tau = 10^4$ testcase. . . . .	44
6.4	Temperature structure for the $\tau = 10^4$ testcase. . . . .	45
6.5	Temperature structure for the $\tau = 10^5$ testcase. . . . .	46
6.6	Temperature structure for the $\tau = 10^6$ testcase. . . . .	47
6.7	Convergence properties of the temperature correction scheme. . . . .	51
6.8	Convergence in the midplane for the $\tau = 10^4$ test case. . . . .	52
6.9	Convergence in vertical direction for the $\tau = 10^4$ testcase. . . . .	52
6.10	Convergence for different initial conditions. . . . .	53



# Acknowledgements

I would like to thank Prof. Manuel Güdel for providing me with this interesting and also quite challenging topic for my thesis, for all his support during my work and his warm welcome to his group. I am looking forward to a possible continuation of my work in your group.

Further I want to thank Prof. Peter Hauschildt for giving me insight into PHOENIX/3D and for providing me with access and computational time for the Höchstleistungsrechenzentrum Nord (HLRN). Without this the extensive testing would not be possible.

For his generous help concerning radiative transfer in protoplanetary disks I would like to thank Dr. Peter Woitke. His experience with modelling of protoplanetary disks was very helpful and enlightening.

I would also like to thank Dr. Ines Brott for proofreading the first versions of my script, and all my other office colleagues for their friendly welcome.

Special thanks goes to my girlfriend Katrin and my daughter Anika, for their endless patience and for cheering me up when I was worrying about my work progress. Ich hab euch einfach nur so lieb (Rab, A. 2012, priv. comm.)!

Meiner Mutter, meinem Vater und dem Rest meiner Familie möchte ich danken für deren Unterstützung auf meinem Weg, auch wenn meine Entscheidungen für sie nicht immer nachvollziehbar waren.



# Curriculum Vitae

## I. Personal Data

Date of birth: April 04, 1978  
Place of birth: Amstetten, Austria  
Homepage: <http://homepage.univie.ac.at/christian.rab>  
Citizenship: Austria



## II. Education & Research

since Sept. 2009 Master studies of Astronomy at the University of Vienna

### 12. Dec - 16. Dec. 2011

26<sup>th</sup> Concurrent Engineering Study of the German Aerospace Center (DLR).  
Continuation of the mission design which was started at the Summer school Alpbach.

### 19. Jul. - 28. Jul 2011

ESA/FFG Summer school Alpbach "Star Formation across the Universe"  
Design of a space mission for the observation of star formation in our Universe.

Mar. 2006 - Aug. 2009 Bachelor studies of Astronomy at the University of Vienna

24. Aug. 2009: Bachelor degree in Astronomy, passed with distinction.  
Topic of my thesis: *Direct Evidence for Dark Matter in Clusters of Galaxies*

Feb. 1999 - Jan. 2006 Software engineer - Dr. E. Hackhofer GesmbH, Vienna

Feb. 1998 - Jan. 1999 Alternative civilian service at a nursing home for the elderly

Oct. 1997 - Jan. 1998 Software engineer - Dr. E. Hackhofer GesmbH, Vienna

Sept. 1992 - Jun. 1992 HTL (higher technical school), St. Pölten,

Department: EDP and Organisation/Computer Science  
6. Jun. 1997: Graduated (Matura) - passed with merit

Sept. 1987 - Jun. 1992 secondary general school (Hauptschule), Blindenmarkt

Sept. 1984 - Jun. 1987 primary school, Ferschnitz

## III. Teaching experiences

### Mar. 2010 - Jul. 2010

Tutor at the University of Vienna - Introduction to Astronomy II

### Oct. 2010 - Feb. 2010

Tutor at the University of Vienna - Introduction to Astronomy III

### Mar. 2011 - Jul. 2011

Tutor at the University of Vienna - Introduction to Astronomy II