Contrib. Astron. Obs. Skalnaté Pleso **??**, 1 – 53, (2004)

A description of the SHELLSPEC49 code

J. Budaj[1]

Astronomical Institute of the Slovak Academy of Sciences
059 60 Tatranská Lomnica, The Slovak Republic,
(E-mail: budaj@ta3.sk)

Received: May 22, 2010; Accepted: ?

Abstract. Program SHELLSPEC is designed to calculate lightcurves, spectra and images of interacting binaries and extrasolar planets immersed in a moving circumstellar matter (CM). It solves simple radiative transfer along the line of sight in 3D moving media. Roche model can be used as a boundary condition for the radiative tranfer. The scattered light from the two stars can be taken into account assuming that CM is optically thin. The assumptions include LTE and optional known state quantities and velocity field in 3D. These can be taken from the 3D hydrodynamical simulations. Alternatively, optional (non)transparent objects such as: a central star, companion star, spot, stream, ring, disc, envelope, nebula, flow, jet, ufo, shell or an empty space may be defined in 3D and their composite synthetic spectrum calculated. Stars may have either the Roche or spherical geometry, optional velocity, differential rotation and a spot on the surface. They are subject to the gravity darkening, limb darkening, and irradiation effect including the heating, reflection and day-night heat redistribution. They may be ascribed a precalculated spectrum. Synthetic light curves or trailing spectrograms can be produced by changing your view points on the 3D object.

The main applications are probably in the field of interacting binaries, cataclysmic variable stars, Algol-type eclipsing binaries, and extrasolar planets but the code is a flexible tool which can be used to study a large variety of objects and effects.

Key words: Radiative transfer – Accretion, accretion discs – Stars: binaries: close – Stars: binaries: eclipsing – Stars: novae, cataclysmic variables

1. Introduction

There are sophisticated computer codes for calculating and inverting light curves or spectra of binary stars with various shapes or geometry including the Roche model (Lucy 1968; Wilson & Devinney 1971; Wood 1971; Mochnacki & Doughty 1972; Rucinski 1973; Hill 1979; Popper & Etzel 1981; Zhang et al. 1986; Djurasevic 1992; Drechsel et al. 1994; Vinkó et al. 1996; Hadrava 1997; Orosz & Hauschildt 2000; Bradstreet & Steelman 2002; Pribulla 2004, Southworth et al. 2004, Pavlovski et al. 2006, Tamuz et al. 2006, Jackson et al. 2012). The Wilson & Devinney code is most often used and is continuously being improved or modified (Wilson 1990; Kallrath et al. 1998; Prša & Zwitter 2005, Horvat et al.

2019). The main focus of these codes is to deal with the complicated geometry and to determine the properties of stars and their orbit. However, it is often the case that stellar objects are embedded in some moving optically thin environment and/or are accompanied by rings, discs, streams, jets or shells which give rise to various emission spectra.

There are different approaches to model the circumstellar matter (CM). For example a simple volume integration of emissivity, radiative transfer in a moving disc along the line of sight Horne & Marsh (1986), ray tracing code RADLite for modelling protoplanetary disks (Pontoppidan et al. 2009), 2D radiative transfer in axial symetry in the presence of arbitrary velocity fields Korčáková & Kubát (2005) or a Sobolev aproximation. Very sophisticated NLTE models and spectra of accretion discs based on static local atmospheres were developped by Hubeny & Lanz (1995) and Linnell & Hubeny (1996). General 3D radiative transfer codes are being developed (Ibgui et al. 2013). Hydrodynamical simulations can produce a 3D model of the CM (behavior of state quantities and velocity field) (see for example Richards & Ratliff 1998; Bisikalo et al. 1998; Nazarenko & Glazunova 2006).

The aim of this paper is not to compete with the codes and models mentioned above but rather to bridge the gap in these present approaches and provide a tool which would solve in LTE the simple radiative transfer along the line of sight in an optional optically thin 3D moving medium with the possible nontransparent objects embedded in. The code is quite a multi-purpose, independent, and flexible tool which can calculate light curves, spectra, trailing spectrograms, 2D images and can be used to study various objects or effects. It does not solve the inverse problem of finding stellar and orbital parameters of binary stars and the user is referred to other well known codes which deal with 'solid surfaces' and can handle this problem more effectively. It was originally written in Fortran77. Andrew Tkachenko and Klara Sejnova developed F90 versions of Shellspec (version07) and additional procedures which solve some restricted inverse problems (Tkachenko et al. 2010 and Sejnova et al. 2012). Janka Nemravova and Mirek Broz developed another program (Mourard et al. 2018) which also calculates interferometric observables and can solve the inverse problem simultaneously for photometry, spectroscopy, and interferometry.

This is version49. The main modification since the last public release (version 39) are: the whole code with all subroutines was rewritten into F90 format; introduction of nominal conversion units (Prsa et al. 2016); code can handle large optical depths (assuming no scattering); possibility to include shadows; more sophisticated models for objects JET, NEBULA; reddening and extinction according CCM89 (Cardelli et al 1989); and possibility to define two subgrids with different scales and density (in both the body frozen and line of sight grids).

2. Basic Astrophysics

2.1. Radiative Transfer

In the following analysis, the calculations are carried out in the observer's Cartesian frame with z pointing towards the observer. The radiative transfer equation along the line of sight is:

$$dI_\nu = (\epsilon_\nu - \chi_\nu I_\nu)dz \tag{1}$$

where I_ν is the specific monochromatic intensity at the frequency ν, χ_ν is the opacity, ϵ_ν is the emissivity and z is the distance along the beam. It is convenient to split the opacity into two contributions, the true absorption κ_ν and the scattering σ_ν:

$$\chi_\nu = \kappa_\nu + \sigma_\nu. \tag{2}$$

Assuming LTE, the line opacity corrected for a stimulated emission is stated simply as:

$$\chi_\nu^{line} = (1 - e^{-\frac{h\nu}{kT}})N_l B_{lu} h\nu\varphi_{lu}(\nu - \nu_0)(4\pi)^{-1} \tag{3}$$

where h is the Planck constant, $h\nu$ is the energy of the transition from the lower level l to the upper level u, k is the Boltzmann constant, T is the temperature, N_l is the population of the l-th state of the corresponding ion, and B_{lu} is the Einstein coefficient for the whole solid angle. The velocity field enters the equation via the shifted normalized Voigt profile $\varphi_{lu}(\nu - \nu_0)$ where

$$\nu_0 = \nu_{lu}\left(\frac{v_z(z)}{c} + 1\right) \tag{4}$$

where ν_{lu} is the laboratory frequency of the line and $v_z = \boldsymbol{v}.\boldsymbol{n}$ is the radial velocity (positive towards the observer) or projection of the local velocity vector $\boldsymbol{v}$ to the line of sight unit vector $\boldsymbol{n}$. The Einstein coefficient, B_{lu}, is related to the oscillator strength, f_{lu}, by:

$$B_{lu} = \frac{4\pi^2 e^2 f_{lu}}{m_\mathrm{e} c h \nu_{lu}} \tag{5}$$

where e, m_e are the electron charge and mass, respectively and c is the speed of light. The shape of the Voigt profile is determined by the thermal and the microturbulent broadening, v_{trb}, characterized by the Doppler half-width

$$\Delta\nu_D = \frac{\nu}{c}\sqrt{\frac{2kT}{m} + v_{trb}^2} \tag{6}$$

as well as by the damping broadening characterized by the frame damping parameter

$$a = \gamma/(4\pi\Delta\nu_D) \tag{7}$$

where the damping constant

$$\gamma = \gamma_{Nat.} + \gamma_{Stark} + \gamma_{VDW} \tag{8}$$

includes the contribution from the Natural, Stark and Van der Waals broadening. In the case of LTE, all the line opacity is due to the true absorption process i.e.:

$$\chi_\nu^{line} = \kappa_\nu^{line} \tag{9}$$

It is now possible to supply the line opacity, or any kind of extra true absorption opacity (e.g. due to the molecular lines), in the form of an opacity distribution function $odf(\nu, T)$ as a function of frequency and temperature. The line opacity at a particular frequency is then:

$$\kappa_\nu^{odf} = odf(\nu_2, T) \tag{10}$$

where

$$\nu_2 = \nu \left(1 - \frac{v_z}{c}\right). \tag{11}$$

Apart from that the following continuum opacity sources can be turned on-off: the HI bound-free opacity, the HI free-free opacity, the H^- bound-free, the H^- free-free, Thomson scattering on free electrons, Rayleigh scattering on neutral hydrogen, and Mie scattering/absorption on dust. These are considered to vary only slowly with frequency and the effect of the velocity field are not considered for them.
For HI bound-free opacity, based on Gray (1976) and Mihalas (1978), we have:

$$\kappa_\nu^{HIbf} = \frac{2.8154 \times 10^{29}}{\nu^3} n_{HI} \frac{2}{u_{HI}} (1 - e^{-\frac{h\nu}{kT}}) \times \left[\sum_{n=n_0}^{n_0+2} \left(\frac{g_n^{bf}}{n^3} e^{-\frac{-\chi_n}{kT}}\right) + \frac{kT}{2I}\left(e^{-\frac{-\chi_{n_0+3}}{kT}} - e^{-\frac{-I}{kT}}\right)\right] \tag{12}$$

where n_{HI} is the neutral hydrogen number density, u_{HI} is its partition function, n is the main quantum number, n_0 is the value of n for the lowest level of importance (levels with $n < n_0$ have their photoionization edges at higher frequency than ν and thus do not contribute to the opacity at ν), $g_n^{bf}(\nu)$ is the bound-free Gaunt factor of the particular level, χ_n is the excitation potential of the level and I is the ionization potential of the ion. In the formula above, the contribution from the levels: $n_0, n_0 + 1, n_0 + 2$ is taken into account explicitly while the contribution of the higher levels with $n > n_0 + 2$ is integrated.
The HI free-free opacity (Mihalas 1978) is:

$$\kappa_\nu^{HIff} = 3.69 \times 10^8 g^{ff} \frac{n_e n_{HII}}{\nu^3 T^{1/2}} (1 - e^{-\frac{h\nu}{kT}}) \tag{13}$$

where $g^{ff}(T,\nu)$ is the free-free Gaunt factor, n_e is the electron number density and n_{HII} is the proton number density.
Thomson scattering opacity (Mihalas 1978) is:

$$\sigma_\nu^{TS} = n_e\sigma_e = 6.65\times10^{-25}n_e \quad . \tag{14}$$

For Rayleigh scattering on neutral hydrogen we adopted the following expression from Kurucz (1970):

$$\sigma_\nu^{RS} = n_{HI,0}[5.799\,10^{-13} + (1.422\,10^{-6} + 2.784w)w]w^2 \tag{15}$$

where $w = 10^{-16}[min(\nu, 2.463\,10^{15})/c]^2$ and $n_{HI,0}$ is the population of the ground state of neutral hydrogen.
Bound-free, κ^{H^-bf}, and free-free, κ^{H^-ff}, opacity of H^- is also calculated following Kurucz (1970):

$$\kappa^{H^-bf} = n_{H^-}\alpha_\nu(1 - e^{-\frac{h\nu}{kT}}) \tag{16}$$

where n_{H^-} is negative hydrogen number density and cross-section α_ν is:

$$\begin{aligned}\alpha_\nu = 6.801\,10^{-20} + \{5.358\,10^{-3} + [1.481\,10^{13} + \\ (-5.519\,10^{27} + 4.808\,10^{41}/\nu)/\nu]/\nu\}/\nu\end{aligned} \tag{17}$$

for $\nu \geq 2.111\,10^{14}$ Hz, and

$$\alpha_\nu = 3.695\,10^{-16} + (-1.251\,10^{-1} + 1.052\,10^{13}/\nu)/\nu \tag{18}$$

for $2.111\,10^{14} > \nu > 1.834\,10^{14}$ Hz, and $\alpha_\nu = 0$ for $\nu < 1.834\,10^{14}$ which is close to the ionization limit of H^-.

$$\kappa^{H^-ff} = n_{HI,0}n_eF_\nu(T) \tag{19}$$

$$F_\nu(T) = [1.3727\,10^{-25} + (4.3748\,10^{-10} - 2.5993\,10^{-7}/T)/\nu]/\nu. \tag{20}$$

The total true absorption κ_ν is the sum of these opacity sources:

$$\kappa_\nu = \kappa_\nu^{line} + \kappa_\nu^{odf} + \kappa_\nu^{HIbf} + \kappa_\nu^{HIff} + \kappa^{H^-bf} + \kappa^{H^-ff} \tag{21}$$

The total scattering opacity σ_ν is:

$$\sigma_\nu = \sigma_\nu^{TS} + \sigma_\nu^{RS}. \tag{22}$$

The thermal emissivity associated with the true absorption can then be written as

$$\epsilon_\nu^{th} = B_\nu(T(z))\kappa_\nu \tag{23}$$

where B_ν is the Planck function for the local value of the temperature. For scattering emissivity we have

$$\epsilon_\nu^{sc} = \iint \sigma(\nu', \nu, \boldsymbol{n'}, \boldsymbol{n})I(\nu', \boldsymbol{n'})d\nu'\frac{d\omega'}{4\pi} \tag{24}$$

where $\sigma(\nu', \nu, \boldsymbol{n'}, \boldsymbol{n})$ is the scattering coefficient containing the general redistribution function. It is this term which causes the main difficulty, since apart from redistributing the frequencies ($\nu' \rightarrow \nu$), it also couples the radiation in one direction $\boldsymbol{n}$ with the radiation field in all other directions $\boldsymbol{n'}$. However, in many applications (e.g., optically thin shells) this term can either be neglected or governed by the light scattering from a few bright objects. We assume scattering from a blackbody or from spherical stars with precalculated surface intensity $I^\star_\nu$ or flux $F^\star_\nu$. In case of coherent isotropic scattering (as seen from the scattering particle frame) the emissivity reduces to:

$$\epsilon^{sc}_\nu = \sigma_\nu J_\nu \tag{25}$$

where J_ν is the mean intensity. Ignoring limb darkening, J_ν can be approximated by:

$$J_\nu \approx I^\star_{\nu_1} \omega / 4\pi \tag{26}$$

where ω is the solid angle subtended by the star and

$$\omega/4\pi = \frac{1}{2}\left(1 - \sqrt{1 - \frac{R^2_\star}{|\boldsymbol{r} - \boldsymbol{r}_\star|^2}}\right) \tag{27}$$

where $\boldsymbol{r}$ is the position vector of the scattering particles and $R_\star, \boldsymbol{r}_\star$ are the radius and position vector of the star, respectively, and

$$\nu_1 = \nu\left(1 - \frac{v_1}{c}\right) \tag{28}$$

and

$$v_1 = -\frac{(\boldsymbol{r} - \boldsymbol{r}_\star).(\boldsymbol{v} - \boldsymbol{v}_\star)}{|\boldsymbol{r} - \boldsymbol{r}_\star|} + v_z \tag{29}$$

where $\boldsymbol{v}$ is the velocity field vector at the given point specified by the vector $\boldsymbol{r}$ and $\boldsymbol{v}_\star$ is the velocity of the center of mass of the star.
For $R_\star/r << 1$ an approximation including the limb darkening and the non-isotropic phase function $g(\boldsymbol{n'}, \boldsymbol{n})$ is used:

$$\epsilon^{sc}_\nu = g(\boldsymbol{n'}, \boldsymbol{n})\sigma_\nu J_\nu \tag{30}$$

where $g(\boldsymbol{n'}, \boldsymbol{n})$ is the dipol phase function:

$$g(\boldsymbol{n'}, \boldsymbol{n}) = \frac{3}{4}(1 + (\boldsymbol{n'}.\boldsymbol{n})^2) = \frac{3}{4}\left(1 + \frac{(r_z - r_{\star,z})^2}{|\boldsymbol{r} - \boldsymbol{r}_\star|^2}\right). \tag{31}$$

For J_ν one can use

$$J_\nu \approx \frac{F^\star_{\nu_1}}{4\pi}\frac{R^2_\star}{|\boldsymbol{r} - \boldsymbol{r}_\star|^2} \tag{32}$$

where

$$F^\star_{\nu_1} = \pi I^\star_{\nu_1}\left(1 - \frac{u_1}{3} - \frac{u_2}{6}\right) \tag{33}$$

where $u_{1,2}$ are quadratic limb darkening coefficients (see Eq.70) and $I^{\star}_{\nu_1}$ is the intensity normal to the surface. Note that the calculations of the mean intensity and line opacities take into account the mutual velocities of the medium and stars but we ignore the velocity field in the calculations of the continuum and dust opacities.
The total emissivity is then

$$\epsilon_\nu = \epsilon^{th}_\nu + \epsilon^{sc}_\nu \tag{34}$$

and the total source function is:

$$S_\nu = \epsilon_\nu / \chi_\nu. \tag{35}$$

The flux, F_ν, from the object at the Earth is then obtained by the integration of the output intensities I^{out}_ν through the 2D projection surface of the 3D object:

$$F_\nu = \int I^{out}_\nu cos(\theta) d\Omega \approx \int I^{out}_\nu d\Omega \tag{36}$$

where θ is an angle bewteen the ray and normal to the detector. In most astronomical situations objects are so far away that $cos\theta \approx 1$. Ω is the solid angle on the sky subtended by the object and

$$F_\nu = \iint \frac{I^{out}_\nu}{D^2} dxdy \tag{37}$$

where D is the distance to the object from the Earth.

2.2. Dust

Some astronomical objects have low enough temperature and high enough density that grains of condensates can be formed. Such grains are usually called "dust", although some authors use the more generic term "condensates". With some limitations and understanding, it is also possible to include dust into your SHELLSPEC calculations. Calculation of dust optical properties may be quite complicated and expensive when calculated on fly. That is why we precalculated various tables which can be loaded into SHELLSPEC to save you the effort and speed up the calculations. These tables are described in more detail in Budaj et al. (2015). They assume homogeneous spherical grains and Deirmendjian particle size distribution. They are freely available for download from

`http://www.ta3.sk/~budaj/dust`

A brief introduction is given below.

Dust can absorb the impinging radiation and convert it directly into heating of the grains. This process is called 'absorption' or 'true absorption' to emphasize that the photon is thermalized. It is quantified by the absorption opacity. Dust can also scatter radiation in a process called 'scattering' without being heated.

This process is characterized by the scattering opacity. Furthermore, scattering can be highly asymmetric, a property that is described by means of the phase function, which depends on the scattering angle (the deflection angle from the original direction of the impinging radiation). The most prominent feature is a strong forward scattering for large values of $x = 2\pi a/\lambda$ where a is particle size (radius) and λ is wavelength. Finally, condensation affects the chemical composition of the object. It removes the condensed elements from the gas phase within the dust cloud and, in the atmosphere, also from the region above the clouds due to the rain-out.

2.2.1. Dust opacities

Optical properties of spherical homogeneous grains can be calculated using Mie theory. These are mainly $Q_{\rm a}$, $Q_{\rm s}$, and $Q_{\rm e}$, which are efficiency factors for absorption, scattering, and extinction, respectively, for particles with radius r. They are related to the $C_{\rm a}$ and $C_{\rm s}$, the absorption and scattering cross-sections, via

$$C_{\rm a} = Q_{\rm a}\pi r^2, \quad C_{\rm s} = Q_{\rm s}\pi r^2, \quad Q_{\rm e} = Q_{\rm a} + Q_{\rm s}. \tag{38}$$

The average cross-sections for absorption $\overline{C_{\rm a}}$ and scattering $\overline{C_{\rm s}}$ for an ensemble of particles with an optional size distribution n_r are:

$$\overline{C_{\rm a}} = \frac{1}{n}\int C_{\rm a} n_r dr \qquad \overline{C_{\rm s}} = \frac{1}{n}\int C_{\rm s} n_r dr, \tag{39}$$

where

$$n = \int n_r dr. \tag{40}$$

Subsequently, the absorption and scattering opacities $\kappa_\nu^{\rm dust}, \sigma_\nu^{\rm dust}$ of condensates in units of cm^{-1} are given by

$$\kappa_\nu^{\rm dust} = \overline{C_{\rm a}} n\ , \qquad \sigma_\nu^{\rm dust} = \overline{C_{\rm s}} n. \tag{41}$$

For practical applications it is more convenient to use the opacities per gram of dust material, given by

$$\kappa_{\nu,\rho}^{\rm dust} = \kappa_\nu^{\rm dust}/\rho_{\rm d}\ , \qquad \sigma_{\nu,\rho}^{\rm dust} = \sigma_\nu^{\rm dust}/\rho_{\rm d} \tag{42}$$

where $\rho_{\rm d}$ is the density of dust made of particular species and $\kappa_{\nu,\rho}^{\rm dust}, \sigma_{\nu,\rho}^{\rm dust}$ are absorption and scattering opacities, respectively, per gram of dust material in units of $cm^2\, g^{-1}$. The dust density can be expressed as

$$\rho_{\rm d} = \int M_r n_r dr = \rho^{\rm g}\int V^{\rm g} n_r dr = \rho^{\rm g}\overline{V^{\rm g}} n = \overline{M^{\rm g}} n \tag{43}$$

where M_r is mass of a dust grain of radius r. $\overline{V^{\rm g}}, \overline{M^{\rm g}}$ are the mean volume and mass of a dust grain, respectively, and

$$\overline{M^{\rm g}} = \frac{\rho^{\rm g}}{n}\int V^{\rm g} n_r dr = \frac{4\pi\rho^{\rm g}}{3n}\int r^3 n_r dr. \tag{44}$$

In the above expression, $\rho^{\rm g}$ is the (constant) density of a grain of dust. Substituting equations (43) and (41) into equation (42) one can obtain the following relation between opacities and cross-sections:

$$\kappa_{\nu,\rho}^{\rm dust} = \frac{\overline{C_{\rm a}}}{\overline{M^{\rm g}}} \tag{45}$$

$$\sigma_{\nu,\rho}^{\rm dust} = \frac{\overline{C_{\rm s}}}{\overline{M^{\rm g}}}. \tag{46}$$

Using these opacities, the monochromatic optical depth along the line of sight z is then given by:

$$\tau_\nu = -\int \rho_{\rm d}(z)[\kappa_{\nu,\rho}^{\rm dust}(z) + \sigma_{\nu,\rho}^{\rm dust}(z)]dz. \tag{47}$$

2.2.2. Dust phase functions

Phase function, $p(\alpha, r)$, describes the angular distribution of the scattered light. It is a function of, α, the scattering angle which measures the deflection from the original direction of the photon and particle size. It is normalized such that its integral over the whole solid angle $d\omega = \sin(\alpha)d\alpha d\phi$ is 4π:

$$\int p(\alpha, r)d\omega = \int_0^{2\pi}\int_0^{\pi} p(\alpha, r)\sin(\alpha)d\alpha d\phi = 4\pi. \tag{48}$$

The phase function of the population of particles, $\overline{p}(\alpha)$, with the size distribution n_r, can be calculated in the following way:

$$\overline{p}(\alpha) = \int p(\alpha, r)C_{\rm s}n_r dr/(\overline{C_{\rm s}}n) \tag{49}$$

Since $C_{\rm s}n_r$ does not depend on direction, the averaged phase function $\overline{p}$ has the same normalization.

The mean cosine of the scattering angle g, also known as the asymmetry parameter, is calculated from the normalized poly-dispersed phase function as:

$$g = \int \overline{p}(\alpha)\cos(\alpha)d\omega \;/\; \int \overline{p}(\alpha)d\omega. \tag{50}$$

Once the opacities are known, the thermal emissivity of condensates (energy per unit time, frequency, volume, and solid angle) associated with its true absorption can be calculated as:

$$\epsilon_\nu^{\rm th,dust} = B_\nu(T)\rho_{\rm d}\kappa_{\nu,\rho}^{\rm dust}, \tag{51}$$

where B_ν is the Planck function. The angle-dependent scattering emissivity of condensates into a direction α_0 can be calculated from the scattering opacity and phase function via the following expression:

$$\epsilon_\nu^{\rm sc,dust}(\alpha_0) = \iint p[\alpha(\alpha_0,\alpha_1),r] C_{\rm s} n_r I_\nu[\theta(\alpha_1,\phi)] dr d\omega/(4\pi) \tag{52}$$

or

$$\epsilon_\nu^{\rm sc,dust}(\alpha_0) = \int \overline{p}(\alpha) \rho_{\rm d} \sigma_{\nu,\rho}^{\rm dust} I_\nu d\omega/(4\pi), \tag{53}$$

where $I_\nu[\theta(\alpha_1,\phi)]$ is the specific intensity of radiation coming from the direction (α_1,ϕ) with solid angle $d\omega = \sin(\alpha_1) d\alpha_1 d\phi$, see Fig. 1. In the following we will write the specific intensity simply as I_ν. One can define another averaged phase function, $P_{\rm DA}$ such that:

$$P_{\rm DA}(\alpha_0) = \frac{\int \overline{p}(\alpha) I_\nu d\omega}{\int I_\nu d\omega} = \frac{\int \overline{p}(\alpha) I_\nu d\omega}{4\pi J_\nu}, \tag{54}$$

where, J_ν is the mean intensity given by:

$$J_\nu = \int I_\nu d\omega/(4\pi). \tag{55}$$

This phase function is an average over the source of light. Since intensity depends on the angle, equation (54) does not guarantee that $P_{\rm DA}$ is normalized, contrary to e.g. $p(\alpha, r)$. However, using $P_{\rm DA}$, the scattering emission can be expressed in a very simple way as:

$$\epsilon_\nu^{\rm sc,dust}(\alpha_0) = P_{\rm DA}(\alpha_0) \rho_{\rm d} \sigma_{\nu,\rho}^{\rm dust} J_\nu. \tag{56}$$

If we consider a distinct source of light, such as a star for example, and a medium surrounding the source and our dust grain, which is optically thin, or if we are assuming single scattering events, then $P_{\rm DA}$ is an average of the phase function over the surface of the star and α_0 can be conveniently measured from the ray originating at the centre of the star. If the star is far or if it is small, then $P_{\rm DA}$ equals $\overline{p}$. However, if the star has a non negligible angular dimension on the sky, compared to the characteristic changes in the phase function, then one has to take its angular dimension into account. The phase function of large grains or at short wavelengths usually has a very strong forward-scattering peak, which can be much sharper than the stellar disc as seen e.g. from a typical hot-Jupiter or a close in extrasolar planet. To take this effect into account one has to split the stellar disc into elementary surfaces and integrate the phase function over the disc as in the equation (54). In doing so we assume a quadratic limb darkening of the stellar surface:

$$I_\nu = I_\nu(0)[1 - u_1(1-\cos\theta) - u_2(1-\cos\theta)^2], \tag{57}$$

where $I_\nu(0)$ is intensity perpendicular to the surface of the source and θ is angle between the line of sight and a normal to the surface.

The Fortran90 code which reads our phase function tables and calculates such disc averaged phase functions $P_{\rm DA}$ is provided together with the tables. As mentioned above, this code and precalculated $P_{\rm DA}$ phase functions are mainly useful in the case that a dust grain 'sees' a distinct source of light and/or in the optically thin dust regime.

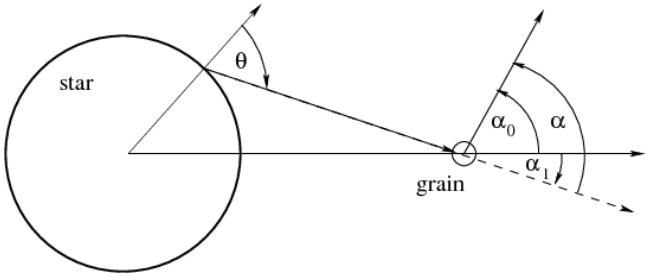

Figure 1. Definition of the geometry and angles.

Note that while J_ν takes into account the mutual velocities of the source of light and dust particle (see Eqs. 26,32) we do not take the velocity field into account when calculating the dust opacities since they are not expected to vary significantly with wavelength.

2.2.3. Interstellar extinction and reddening

Dust is also major contributor to the interstellar extinction. Absorption and scattering by large dust particles (relative to the wavelength) is wavelength independent. However, the scattering by small particles has a very strong, λ^{-4}, dependence (Rayleigh scattering) and the absorption by small particles has λ^{-1} dependence. Blue light is scattered and attenuated more efficiently and for this reason the dust is causing the reddening of light. The extinction at some wavelength or filter in magnitudes is the difference between the observed and intrinsic brightness: $A(V) = V_{obs} - V_{int}$. Reddening (selective extinction/color excess) is usually expressed as a difference between the observed and intrinsic color index:

$$E(B-V) = (B-V)_{obs} - (B-V)_{int} = A(B) - A(V) \tag{58}$$

A relative slope of the wavelength dependence of the extinction can be characterized by a single parameter $1/R(V)$ where R(V) is:

$$R(V) = A(V)/E(B-V) = A(V)/[A(B) - A(V)] \tag{59}$$

R(V) is sensitive to the particle size. The typical value of R(V) for the interstellar dust in our Galaxy is 3.1 ± 0.2. The absolute amount of the extinction as

a function of λ (the extinction curve) can be characterized by two parameters: R(V) and E(B-V). In the code it is possible to correct the model for the extinction and reddening according to the extinction curves of Cardelli, Clayton & Mathis (1989). They depend on R(V) and E(B-V) and cover 0.1-3.3 micron region.

2.3. Roche Geometry

In the SHELLSPEC code the Roche model serves as a boundary condition for the radiative transfer in the circumstellar matter. Both objects, star and companion, may have shapes according to the Roche model for detached or contact systems. Descriptions of the Roche model can be found in Kopal (1959), Plavec & Kratochvil (1964), Mochnacki & Doughty (1972), Hilditch (2001) and many other papers and books. Let us assume a Cartesian coordinate system (x,y,z) centered on one of the stars (labeled as 1) such that the companion (labeled as 2) is at (1,0,0) and revolves around the z axis in the direction of positive y axis. Let the mass ratio, q, always be m_2/m_1 or 'companion/star' and $q < 1$ will indicate the companion is lighter while $q > 1$ means the central star is lighter. Then, the normalized Roche potential, C, is expressed as:

$$C(x,y,z) = \frac{2}{(1+q)r_1} + \frac{2q}{(1+q)r_2} + \left(x - \frac{q}{1+q}\right)^2 + y^2 \tag{60}$$

where $r_1 = \sqrt{x^2+y^2+z^2}$ and $r_2 = \sqrt{(x-1)^2+y^2+z^2}$. The Roche surface of a detached component is defined as an equipotential surface $C_s = C(x_s, y_s, z_s)$ passing through the substellar point (x_s, y_s, z_s) (point on the surface of the star in between the stars, $0 < x_s < 1, y_s = z_s = 0$) which is localized by the 'fill-in' parameter $f_i \leq 1$. We define this by:

$$f_i = x_s/L_{1x}, \qquad f_i = (1-x_s)/(1-L_{1x}) \tag{61}$$

for the primary and the secondary, respectively. L_{1x} is the x coordinate of the L1 point L1$(L_{1x}, 0, 0)$. The Roche equipotential surface C_s of a contact system will be defined by the fill-out parameter $1 < f_o \leq 2$:

$$f_o = \frac{C1 - C_s}{C1 - C2} + 1 \tag{62}$$

where potentials $C1, C2$ correspond to the potentials at the L1 and L2 points, respectively. First, we calculate L1, L2, C_s and x-boundaries of the object using the Newton-Raphson iteration method e.g.,

$$x_{i+1} = x_i - \frac{C(x_i,0,0) - C_s}{C_x(x_i,0,0)} \tag{63}$$

and then the 3D shape of the surface is solved using the Newton-Raphson iteration in y and z coordinates with the precision of about 10^{-7}. Here are the

derivatives necessary for the task:

$$C_x = \frac{\partial C}{\partial x} = -\frac{2x}{(1+q)r_1^3} - \frac{2q(x-1)}{(1+q)r_2^3} + 2x - \frac{2q}{1+q} \tag{64}$$

$$C_y = \frac{\partial C}{\partial y} = -\frac{2y}{(1+q)r_1^3} - \frac{2qy}{(1+q)r_2^3} + 2y \tag{65}$$

$$C_z = \frac{\partial C}{\partial z} = -\frac{2z}{(1+q)r_1^3} - \frac{2qz}{(1+q)r_2^3} \tag{66}$$

$$\frac{\partial^2 C}{\partial x^2} = \frac{6x^2}{(1+q)r_1^5} + \frac{6q(x-1)^2}{(1+q)r_2^5} - \frac{2}{(1+q)r_1^3} - \frac{2q}{(1+q)r_2^3} + 2 \tag{67}$$

Gravity darkening is taken into account by varying the surface temperature according to the von Zeipel's law:

$$T/T_p = (g/g_p)^\beta \tag{68}$$

where g is the normalized surface gravity, β is the gravity darkening coefficient, T_p, g_p are the temperature and gravity at the rotation pole. The corresponding gravity vector is $\boldsymbol{g} = (C_x, C_y, C_z)$ and:

$$g = \sqrt{C_x^2 + C_y^2 + C_z^2}. \tag{69}$$

Notice, that there is an imminent singularity in the calculations in the vicinity of L1, L2 points since gravity falls to zero which drags temperatures (a denominator in many equations) to zero. We avoid the problem by setting the lowest possible value of $g/g_p = 10^{-4}$.

Limb darkening is taken into account using the quadratic limb darkening law:

$$I(\theta) = I(0) f_{LD} \tag{70}$$

$$f_{LD} = 1 - u_1(1 - \cos\theta) - u_2(1 - \cos\theta)^2 \tag{71}$$

and by calculating the cosine of the angle θ between the line of sight unit vector $\mathbf{n} = (n_x, n_y, n_z)$ and a normal to the surface:

$$\cos\theta = -\mathbf{n}.\mathbf{g}/g = -\frac{n_x C_x + n_y C_y + n_z C_z}{\sqrt{C_x^2 + C_y^2 + C_z^2}}. \tag{72}$$

2.4. Irradiation and heat redistribution

In this section we will describe our treatment of the mutual irradiation of the objects. It was developed and tested in more detail in Budaj (2011). It can be applied to both objects but we will neglect multiple reflections between the two objects since this is not essential if one of them is much less luminous.

We will distinguish between three separate processes: **reflection** of the light off the object (or scattering which does not produce any heating of the irradiated surface); **heating** of the irradiated surface (day side) by the absorbed light; and subsequent **heat redistribution** over the entire surface of the object. Let's assume that the day side of a planet is irradiated by the star then the impinging flux at a location $\mathbf{r}$ from star at $\mathbf{r}_*$ is:

$$F_{ir} = \cos\delta \frac{R_\star^2}{(\mathbf{r}-\mathbf{r}_*)^2}\sigma T_\star^4 \tag{73}$$

where $R_\star, T_\star$ are radius and effective temperature of the star, δ is an irradiating angle which is the zenith distance of the center of the star as seen from the surface of the planet:

$$\cos\delta = \frac{(\mathbf{r}-\mathbf{r}_*).\mathbf{g}}{|\mathbf{r}-\mathbf{r}_*|g} = \frac{(r_x-r_{x*})C_x + (r_y-r_{y*})C_y + (r_z-r_{z*})C_z}{|\mathbf{r}-\mathbf{r}_*|\sqrt{C_x^2+C_y^2+C_z^2}}. \tag{74}$$

Now, let's define two local parameters $A_B(\alpha,\beta)$, $P_r(\alpha,\beta)$ on the day side of the planet (irradiated side of an object) where α,β are the longitude and latitude, respectively, measured from the sub-stellar point. P_r will be the local heat redistribution parameter and A_B will be the local Bond albedo of the surface. Consequently $A_B F_{ir}$ is flux immediately reflected off the surface, $(1-A_B)F_{ir}$ will be a fraction of the irradiating energy which is converted into the heat, $P_r(1-A_B)F_{ir}$ will be a part of the latter which is redistributed over the day and night side of the object, while the remaining part, $(1-P_r)(1-A_B)F_{ir}$, will heat the local area. [1] Then the energy conservation for the day-night heat transport can be written as:

$$\iint P_r(1-A_B)F_{ir}dS_{day} = \iint \sigma T_{dn}^4 dS_{day+night} \tag{75}$$

Let's assume, of simplicity (or in the absence of a better approximation), that $A_B(\alpha,\beta), P_r(\alpha,\beta)$ are constant and the heat is homogeneously redistributed over the day and night sides so that the surface has a constant temperature T_0. Then,

$$P_r(1-A_B)\iint F_{ir}dS_{day} / \iint dS_{day+night} = \sigma T_0^4. \tag{76}$$

[1] Note that our local P_r should not be confused with the global P_n parameter Burrows et al. (2006). P_n is a fraction of the impinging stellar radiation which is transferred to and reradiated from the night side, $P_n \approx P_r(1-A_B)/2$. P_n is from the interval 0-0.5 while P_r runs from 0 to 1. P_r is a direct indicator of the heat redistribution while P_n parameter only reflects a combination of the heat redistribution and Bond albedo.

In case the planet has a spherical shape and is far from the star this reduces to:

$$T_0^4 = \frac{1}{4} P_r (1 - A_B) \frac{R_\star^2}{d^2} T_\star^4. \tag{77}$$

Let's explore another case, namely that the horizontal circulation on the planet along the lines of constant latitude is so strong that it dominates the day-night heat transport and the equilibrium surface temperature, T_1, will be a function of latitude. In this case and the assumptions above (spherical planet far from the star), one can consider an energy conservation for a fixed latitude:

$$P_r (1 - A_B) \int F_{ir} \cos(\beta) d\alpha = \sigma T_1^4(\beta) \int \cos(\beta) d\alpha \tag{78}$$

where

$$\cos(\delta) = \cos(\alpha) \cos(\beta). \tag{79}$$

The solution is that the temperature depends on the fourth root of $\cos(\beta)$:

$$T_1^4(\beta) = \frac{1}{\pi} P_r (1 - A_B) \frac{R_\star^2}{d^2} T_\star^4 \cos(\beta) = \frac{4}{\pi} T_0^4 \cos(\beta). \tag{80}$$

The study of these two extreme cases lead us to suggest a heat redistribution model in which the day-night heat transport is a linear combination of the two cases mentioned above. Namely, we will express the surface temperature in the following way:

$$T_{dn}^4(\beta) = T_0^4 [P_a + P_b \cos(\beta)] \tag{81}$$

where P_a, P_b are the 'zonal temperature redistribution parameters'. $P_a =< 0, 1 >$ and P_b is to be determined from Eq.75 so that the total energy budget is conserved. From Eq. 75,76 and 81 we obtain:

$$P_b = (1 - P_a) \frac{\iint dS_{day+night}}{\iint \cos\beta dS_{day+night}}. \tag{82}$$

It can be shown that in case of a spherical planet far from the star

$$P_b = \frac{4}{\pi} (1 - P_a) \tag{83}$$

and $P_b =< 0, 4/\pi >$. Notice, that P_a is a measure of the effectiveness of the homogeneous temperature distribution over the surface versus the zonal distribution. It is intimately linked with the effectiveness of the heat flows along the meridians versus parallels.

Finally, the temperature distribution on the surface of the irradiated planet will be:

$$T^4 = T_{ir}^4 + T_{dn}^4 + T_{old}^4 \tag{84}$$

where $T_{ir}^4 = (1-P_r)(1-A_B)F_{ir}/\sigma$ on the day side, $T_{ir}^4 = 0$ on the night side, and T_{old} is the prior temperature distribution over the surface in the absence of the irradiation (including the gravity darkening etc.). It should be noted that imposing the external irradiation on one side of the object can alter the original temperature distribution. Budaj et al. (2012) argue that the core cooling rates (T_{old}) from the day and night side of a strongly irradiated planet may not be the same and that the difference depends on several important parameters, such as the effectiveness and the depth where the day-night heat transport occurs, the stellar irradiation flux, and vertical redistribution of the opacities, atmospheric abundances, and/or presence of the stratospheres.

Once we know the temperature distribution over the surface, one can approximate the monochromatic flux from the surface as being composed of two parts:

$$F_\nu = F_\nu^{reflect} + F_\nu^{thermal} \tag{85}$$

Reflection depends on the surface albedo A_ν and has to take into account the mutual velocities:

$$F_\nu^{reflect} = A_{\nu_2} F_{\nu_1 ir} \tag{86}$$

$$F_{\nu_1 ir} = \cos\delta \frac{R_\star^2}{(\mathbf{r}-\mathbf{r}_*)^2} F_{\nu_1}^\star \tag{87}$$

where $F_{\nu_1}^\star$ is properly shifted flux emerging from the surface of the irradiating star. The Doppler shifts are the following

$$\nu_2 = \nu\left(1 - \frac{v_z}{c}\right) \tag{88}$$

$$\nu_1 = \nu\left(1 - \frac{v_z + v_2}{c}\right) \tag{89}$$

$$v_2 = -\frac{(\mathbf{r}-\mathbf{r}_\star).(\mathbf{v}-\mathbf{v}_\star)}{|\mathbf{r}-\mathbf{r}_\star|} \tag{90}$$

where $\mathbf{v}$ is the velocity field vector at the given point on the irradiated surface specified by the vector $\mathbf{r}$, $\mathbf{v}_\star$ is the velocity of the center of mass of the irradiating star, and z coordinate points to the observer. [2] To calculate the reflected intensity we assume that the reflection is isotropic in which case:

$$I_\nu^{reflect} = F_\nu^{reflect}/\pi. \tag{91}$$

Finally, $F_\nu^{thermal}$ can be approximated by the flux emerging from the non-irradiated model atmosphere with the effective temperature equal to the surface temperature of the irradiated object given by Eq.84.

$$F_\nu^{thermal} = F_{\nu_2}(T_{eff} = T) \tag{92}$$

[2]Note that generally Doppler shifts in the scattered and thermal radiation are not the same and may not be trivial. More detailed treatement would require high resolution radiative transfer in the irradiated atmospheres for a set of radial planet-star velocities.

and the associated intensity is given by:

$$I_\nu^{thermal} = I_\nu(0)^{thermal} f_{LD} \tag{93}$$

$$I_\nu(0)^{thermal} = \frac{F_\nu^{thermal}}{\pi(1 - u_1/3 - u_2/6)} \tag{94}$$

In this way we fully include into account the mutual velocities of the objects and observer, the rotation of the reflecting object, its limb darkening but neglect the rotation of the irradiating object. In some cases the later can be easily taken into account by feeding the code with the precalculated rotationally broadened spectrum $F_\nu^\star$.

Local Bond albedo used here is a weighted monochromatic albedo A_ν:

$$A_B = \frac{\int A_\nu F_\nu^\star d\nu}{\int F_\nu^\star d\nu}. \tag{95}$$

One has to keep in mind that our albedo refers to the reflected light only (not to the absorbed and re-radiated light). In case the irradiated object is very cold compared to the irradiating object there is a clear distinction between its thermal radiation and reflected radiation. However, if the two objects have comparable temperatures it is almost impossible to distinguish whether a particular photon was scattered or absorbed and re-radiated. In this case it is still possible to use our formalism and e.g. approximate the albedo by the single scattering albedo to cope with the problem.

It might be convenient to define the mean temperature of the whole distorted object:

$$T_{mean}^4 \equiv \frac{\int T^4 dS}{\int dS}. \tag{96}$$

Note that it does not takes into account the reflected light and should not be confused with the effective temperature or brightness temperature.

3. Numerical Design

3.1. Solution of the Radiative Transfer Equation

A number of optional objects (transparent, nontransparent, empty space, ...) can be defined within the model and the line of sight may cross more of them within a few grid points. A simple and stable method is needed to cope with such velocity, density, temperature fields which are optional and are allowed to be noncontinuous. The formal solution of the radiative transfer along the beam at the frequency ν is the intensity $I_\nu(0)$ at the surface emerging from the object towards the observer:

$$I_\nu(0) = \int_0^{\tau_\nu} S_\nu e^{-t_\nu} dt_\nu + I_\nu^b e^{-\tau_\nu} \tag{97}$$

where S_ν is the source function, τ_ν is the total optical depth (t_ν), I^b_ν is the boundary condition (radiation impinging from behind the object). The definition of S_ν, t_ν is:

$$S_\nu = \frac{\epsilon_\nu}{\chi_\nu}, \quad t_\nu(z) = -\int_{z_2}^{z} \chi_\nu dz \tag{98}$$

where ϵ_ν, χ_ν are emission and absorption coeficients, respectively, z is the geometrical distance along the ray. Object is confind between two boundaries $z_1 < z < z_2$. Optical depth $t_\nu(z_2) = 0$ at the surface facing the observer and increases along the line of sight, which is opposite to the geometrical depth which increases along the ray, and $\tau_\nu = t_\nu(z_1)$. Integration of the Eq.97 is carried out along the optical depth t_ν i.e. along the line of sight from the observer to the far end of the object. Given the optional (potentially discontinuous) behaviour of the quantities we implement simple trapesoidal integration of the Eq.97:

$$I_{i+1} = I_i + (S_{i+1}e^{-t_{i+1}} + S_i e^{-t_i})(t_{i+1} - t_i)/2 \tag{99}$$

where $I_1 = 0$.

If the line of sight happens to hit a nontransparent object the integration is stopped, the bondary condition I^b_ν is calculated and the second term of Eq.97 added.

$$I^b_\nu = I^\star(\nu_2) f_{LD} \tag{100}$$

where

$$\nu_2 = \nu\left(1 - \frac{v^\star_z}{c}\right) \tag{101}$$

and $v^\star_z$ is the radial velocity of the surface of the nontransparent object where it intersects the line of sight. Here $I^\star$ is the surface intensity of the nontransparent object perpendicular to the surface in the comoving (frozen to the surface) frame. If $I^\star$ is not defined in the input it is calculated from the surface flux using Eq.33. In case of the black body approximation with the limb darkening one can use

$$I^\star(\nu_2) = B_{\nu_2}(T_{eff})/(1 - \frac{u_1}{3} - \frac{u_2}{6}). \tag{102}$$

This follows from the definitions of the Stefan-Boltzmann constant and the effective temperature, Eq.33, and an assumption of frequency independent limb darkening:

$$\int \pi I^\star_\nu (1 - \frac{u_1}{3} - \frac{u_2}{6}) d\nu = \sigma T^4_{eff} = \int \pi B_\nu d\nu. \tag{103}$$

Rotation of the nontransparent objects is fully taken into account by including it into the calculations of $v^\star_z$. f_{LD} is a limb darkening factor:

$$f_{LD} = 1 - u_1(1 - \cos\theta) - u_2(1 - \cos\theta)^2 \tag{104}$$

where $u_{1,2}$ are the limb darkening coefficient and θ is the angle between the normal to the surface of the nontransparent object and the line of sight. $T_{eff}, I^\star$

here are the local surface quantities which include effects of gravity darkening, irradiation and heat redistribution.

If the line of sight happens to pass through an empty space this region is also skipped and the integration continues with $I_{i+1} = I_i$. If the step in the optical depth were larger than 0.3 it would be broken into a geometrical sequence of smaller steps.

3.2. Calculation of grid points

The code uses two different grids. The so-called 'body frozen' grid in which objects are defined and the so-called 'line of sight' grid in which the spectra and images are calculated. Each of these grids can be composed of two subgrids with different scale and density which are then merged together. Both subgrids may ovelap in which case a denser subgrid has priority. One does not need to use both subgrids. Two subgrids may be convenient in some cases e.g. when one encounters objects with vastly different spatial scales or a binary object with empty space between the components etc.. Code can set up a homogeneous subgrid or a subgrid based on geometric series. Homogeneous subgrids are obvious. Geometric sequence of subgrid points within interval $< x_b, x_e >$ is calculated in the following way:

$$\begin{aligned} x_{n+1} &= (x_b + x_e)/2 \\ X_m &= (x_e - x_b)/2 \\ x_{n+1+i} &= X_m \tfrac{1-q^i}{1-q^n} + x_{n+1} \\ x_{n+1-i} &= -X_m \tfrac{1-q^i}{1-q^n} + x_{n+1} \end{aligned} \tag{105}$$

where X_m is the half-width of the interval, q is the common ratio of the geometric sequence, and i runs from 1 to n. The total number of points is $2n + 1$.

3.3. Rotation of the Observer's Frame

The SHELLSPEC code enables the user to look on the object from different points of view and to calculate the corresponding spectra. Beware that this may not always represent the natural rotation or revolution of objects in a more complicated sytem and it is the user's responsibility to provide proper input files for the task. The input model of the shell is defined in its 'body frozen' Cartesian coordinates (x'', y'', z'') with the z'' axis corresponding to the intrinsic rotation axis of the model. This grid is not moving and is in rest with the observer. Imagine the system frozen at one instance of time and overlay it with the convenient x'', y'', z'' grid. The spectrum is always calculated in the observer's 'line of sight' Cartesian frame (x, y, z) with z pointing to the observer and which has the same center of coordinates (see Figure 2.). In this frame, (x, y) specify 'field of view of your detector' and you want to make sure that it is pointing where you want and covers what you want. This grid is also in rest with the observer and only represents different point of view than the 'body frozen' grid. We first calculate

the body frozen coordinates corresponding to the grid points of the line of sight mesh by rotating the latter along the x axis by an angle i (the inclination of the intrinsic rotation axis of the model to the line of sight) to get new subordinate prime coordinates (x', y', z') and then by rotating the prime coordinates by a sequence of angles α around the $z'' = z'$ axis:

$$\begin{aligned} x' &= x & \qquad z'' &= z' \\ y' &= y\cos i - z\sin i & \qquad x'' &= x'\cos\alpha + y'\sin\alpha \\ z' &= z\cos i + y\sin i & \qquad y'' &= y'\cos\alpha - x'\sin\alpha \end{aligned} \tag{106}$$

Then we interpolate all the scalar and vector quantities from the body frozen coordinates to the grid points of the rotated observer's frame and, finally, make a back-transformation of the vector quantities (velocity field) as listed below:

$$\begin{aligned} f(x'', y'', z'') = (1-t)(1-u)(1-v)f_{i,j,k} + t(1-u)(1-v)f_{i+1,j,k} \\ +tu(1-v)f_{i+1,j+1,k} + (1-t)u(1-v)f_{i,j+1,k} \\ +(1-t)(1-u)vf_{i,j,k+1} + t(1-u)vf_{i+1,j,k+1} \\ +tuvf_{i+1,j+1,k+1} + (1-t)uvf_{i,j+1,k+1} \end{aligned} \tag{107}$$

where

$$\begin{aligned} t &= (x'' - x''_i)/(x''_{i+1} - x''_i) \\ u &= (y'' - y''_j)/(y''_{j+1} - y''_j) \\ v &= (z'' - z''_k)/(z''_{k+1} - z''_k) \end{aligned} \tag{108}$$

Back-transform of vector quantities:

$$\begin{aligned} v'_z &= v''_z & \qquad v_x &= v'_x \\ v'_x &= v''_x\cos\alpha - v''_y\sin\alpha & \qquad v_y &= v'_y\cos i + v'_z\sin i \\ v'_y &= v''_y\cos\alpha + v''_x\sin\alpha & \qquad v_z &= v'_z\cos i - v'_y\sin i \end{aligned} \tag{109}$$

3.4. State Quantities, Level Populations and Miscellaneous Notes

The code assumes the known abundances and behavior of state quantities: temperature T and density ρ. Electron number density n_e is optional. The atomic number density of all atoms is calculated as:

$$n_a = \frac{\rho}{wm} = \rho\frac{\sum a_i}{\sum a_i m_i} \tag{110}$$

where wm is the mean molecular weight and a_i, m_i are the element abundances and atomic masses respectively. Abundances are defined relative to hydrogen:

$$a_i = n_i/n_H \tag{111}$$

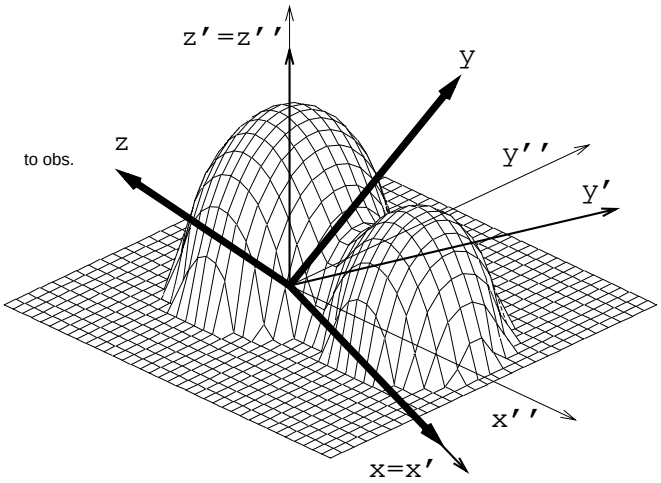

Figure 2. Definition of the 'body frozen' frame (x'', y'', z'') with the z'' axis corresponding to the intrinsic rotation axis of the model and the 'line of sight' frame (x, y, z) with z pointing to the observer.

where n_i, n_H are the element and hydrogen number densities. The hydrogen abundance relative to the total atomic number density is $a'_H = 1/\sum a_i$. Once n_a is known hydrogen number density is calculated as $n_H = a'_H n_a$ and element number densities then follow from their abundances. Solar abundances are assumed (Grevesse & Sauval 1998) but the user is allowed to change all the element abundances. Note that hydrogen number density n_H refers to all hydrogen atoms which can be in the form of n_{HII}, n_{HI}, n_{H-}, or n_{H2}. They satisfy the equation:

$$n_{HII} + n_{HI} + n_{H-} + 2n_{H2} = n_H. \tag{112}$$

All terms can be expressed as a function of n_{HI}:

$$n_{H2} = n_{HI}^2 f_1(T) \tag{113}$$

$$n_{H-} = n_{HI} n_e f_2(T) \tag{114}$$

$$n_{HII} = \frac{n_{HI}}{n_e} f_3(T) \tag{115}$$

where, f_1, f_2, f_3 are mainly functions of temperature, although they contain partition functions $u(HII) = 1, u(H-) = 1, u(HI), u(H2)$ which may slightly

depend on the density. Eq. 112 turns into a quadratic equation in terms of n_{HI}:

$$n_{HI}^2 + \frac{1}{2f_1}(1 + f_3/n_e + n_e/f_2)n_{HI} - \frac{n_H}{2f_1} = 0 \tag{116}$$

which has a form $x^2 + bx + c = 0$ and can be solved easily.

$$n_{HI} = (-b + b\sqrt{1 - 4c/b^2})/2 \tag{117}$$

Potential problem may arrise if $-4c/b^2 << 1$ in which case one has to avoid substracting similar numbers $(-b + b)$ and the solution is $n_{HI} = -c/b$. Other forms of hydrogen are obtained from Eqs.113-115. Function f_3 is taken from Kurucz (1970).

If the electron number density n_e is not known it can be calculated from the known temperature, density, and abundances. The following equation is then solved using the Newton-Raphson iteration:

$$n_e = \sum_i \sum_{j=2}^{J} (j-1)\frac{N_{ji}}{N_i} N_i \equiv f(n_e) \tag{118}$$

where the summation runs through all the chemical elements i and ions j. First J ions of an element are considered. Neutral ion which corresponds to $j = 1$ does not contribute to the sum. N_{ji}, N_i are corresponding ion and element populations. In the text below we will use $ne \equiv n_e$. To speed up calculations you may choose not to consider all the elements (e.g. assuming solar chemical composition, three elements: H, K, Fe, might give an order of magnitude estimate for n_e in a broad range of temperatures and densities). The iteration scheme is the following:

$$ne_{m+1} = ne_m - \delta ne_m = ne_m - \frac{f(ne_m) - ne_m}{g(ne_m) - 1} \tag{119}$$

where

$$g(ne) = \frac{\partial f(ne)}{\partial ne} = \sum_i \sum_{j=2}^{J} (j-1) N_i \frac{\partial N_{ji}/N_i}{\partial ne}. \tag{120}$$

Omitting the i index, ionization fractions of one element can be expressed:

$$\frac{N_j}{N} = \frac{ne^{J-j+1} RT1_j}{A} \tag{121}$$

where

$$A = ne^J + \sum_{j=2}^{J} ne^{J-j+1} RT1_j \tag{122}$$

$$RT1_j = \prod_{k=2}^{j} rt_{k,k-1} \tag{123}$$

$$rt_{k,k-1} \equiv rt_{k,k-1}(T) \equiv \frac{N_k}{N_{k-1}} ne \tag{124}$$

$$rt_{k,k-1}(T) = 4.8294 \times 10^{15} T^{3/2} \frac{u_k(T)}{u_{k-1}(T)} e^{-\frac{I_{k-1}}{kT}} \tag{125}$$

where I_{k-1} is the ionization potential and u_k, u_{k-1} are partition functions. It remains to calculate

$$\frac{\partial N_j/N}{\partial ne} = \frac{(J-j+1) ne^{J-j} RT1_j}{A} - \frac{N_j}{N}\frac{B}{A} \tag{126}$$

where

$$B = \frac{\partial A}{\partial ne} = Jne^{J-1} + \sum_{l=2}^{J} (J-l+1) ne^{J-l} RT1_l. \tag{127}$$

Iterations are stopped when $ne_{m+1} - ne_m < 10^{-3} ne_{m+1}$. In most cases, 3 iterations suffice to achieve the goal. As a starting point of the iteration we choose $ne_0 = n_a$ and iterations are dumped if $ne_m - \delta ne_m < 10^{-3} ne_m$ in which case $ne_{m+1} = 10^{-3} ne_m$. This damping prevents huge downward leaps $\delta ne_m \approx ne_m$ and rounding errors which could occure at low temperatures. Hydrogen contribution to $f(ne), g(ne)$ is included via a separate routine which takes into account formation of negative hydrogen ion (H^-) and $H2$ molecule:

$$f_H(ne) = (f_3/n_e - n_e/f_2) n_{HI} \tag{128}$$

$$g_H(ne) = (-f_3/n_e^2 - 1/f_2) n_{HI} + (f_3/n_e - n_e/f_2) \frac{\partial n_{HI}}{\partial ne} \tag{129}$$

$$\frac{\partial n_{HI}}{\partial ne} = \frac{\partial n_{HI}}{\partial b}\frac{\partial b}{\partial ne} \tag{130}$$

$$\frac{\partial n_{HI}}{\partial b} = \frac{1}{2}(-1 + b/\sqrt{b^2 - 4c}) \tag{131}$$

$$\frac{\partial b}{\partial ne} = \frac{1}{2f_1}(1/f_2 - f_3/n_e^2) \tag{132}$$

The level populations are obtained from the Boltzmann and Saha equations. Partition functions were taken from the UCLSYN code (Smith & Dworetsky 1988, Smith 1992). A FORTRAN77 code containing the partition function routines and tables is also available in Budaj, Dworetsky & Smalley (2002). We have rewritten these routines to F90 and included them in the code. The Gaunt factors are calculated with the subroutines taken from the SYNSPEC code (Hubeny et al. 1994). Damping constants can be found from the VALD atomic line database (Kupka et al. 1999) or in Kurucz (1993a). If the damping constants are not known they are estimated in the code in the way analogous to the SYNSPEC code:

$$\begin{aligned} \gamma_{Nat.} &= 2.4734\, 10^{-22} \nu_{lu}^2 \\ \gamma_{Stark} &= 10^{-8} n_e n_{eff}^5 \\ \gamma_{VDW} &= 4.5\, 10^{-9} X^{0.4} (n_{HI} + 0.42 n_{HeI}) \left(\tfrac{T}{10^4}\right)^{0.3} \end{aligned} \tag{133}$$

where n_{HeI} is the neutral helium number density and the reader is referred to the SYNSPEC source code for the details on X, n_{eff}.

The user needs to ensure that the model is consistent with the abundances and is realistic, e.g., that it satisfies the continuity equation and other conditions where necessary. The user can easily modify the subroutine SMOD1 and insert his/her own prescription for a model there or load a precalculated model from a file.

3.5. Adopted Routines and Units

Several routines used in this code were adopted from other sources. These are:
pfdwor (from UCLSYN, Smith & Dworetsky 1988);
voigt0, state0, gaunt, gfree (from SYNSPEC, Hubeny et al. 1994); and
locate, hunt (from Numerical Recipes, Press et al. 1992).
We also used a few sections from our previous original codes for calculations of radiative accelerations in stellar atmospheres of hot stars (Budaj & Dworetsky 2002). Apart from the above, the code was written from scratch and provides a quite independent tool to study a large variety of objects and effects. Although a substantial effort was devoted to check the calculations there is no guarantee that the code is error free and we would appreciate any bug/error reports.

CGS units are used within the code and the manuscript if not specified otherwise. Nominal values of the solar radius and mass are exploited as conversion factors for input/output in the form (Prsa et al. 2016):

$$\begin{aligned} &1R_{\odot}^{N} = 6.957\,10^{10} cm \\ &1(GM)_{\odot}^{N} = 1.3271244\,10^{26} cm^3 s^{-2} \\ &M_{\odot} = (GM)_{\odot}^{N}/G, \end{aligned} \tag{134}$$

where G is gravity constant. Its exact value is not important since $(GM)_{\odot}^{N}$ is known much more precisely.

4. Compilation

The code is distributed in the form of several files:

shellspecxx.f90	the source code
pfdwor_inc.f90	an include file with the partition functions subroutine
param.inc	an include file with array dimensions
shellspec.in	an example of the main input file
line.dat	an example of the line data input file
shellspec.mod	an example of the input 3D model of the shell
abundances	an example of the abundances input file
phases	an example input file with phases
lambda	an example input file with lambdas

starspec1	an example input spectrum of the primary star
starspec2	an example input spectrum of the secondary star
starspec3	an example input spectrum of the 'third body'
albedo1	an example input of albedo of the primary star
albedo2	an example input of albedo of the secondary star
dust_opac	an example input of dust opacities
mie_phase	an example input of dust phase function
gas_opac	an example input of molecular cross-sections
chem_eq_tab	an example input of molecular populations.

The code is written in standard FORTRAN90. To compile and link the code under LINUX use the following command:

gfortran shellspecxx.f90 -o shellspec
or
gfortran -O2 -mcmodel=medium shellspecxx.f90 -o shellspec
(this last option may allow you to use more memory and run it faster)

where 'xx' stands for the current version of the code. To run the code write:

./shellspec

You may need to adjust the array dimensions according to the available memory. As needed, modify the array dimensions in the file param.inc. The most memory consuming are 3D fields with state quantities and velocities defined in the body frozen grid with the xyz dimensions: $ndimf1 \times ndimf2 \times ndimf3$. Also a $ndim1 \times ndim2 \times mfreq$ field in the line of sight grid which stores output 2D intensities (images) as a function of frequency can occupy a lot of memory.

5. Input

The list of all possible input files with their associated unit numbers follows. These input files are described in more detail in the subsections below:

shellspec.in -(9) main input (geometry, objects...)
line.dat - (8) atomic data for the spectral line (optional if iline=1)
shellspec.mod - (10) input 3D model of the shell (optional if $imodel = 2$)
abundances - (7) abundances (optional if $ichemc = 1$ or $ielnd = 1$)
phases - (15) orbital phases to calculate (optional if $nphase = 0$)
lambda - (15) list of lambdas to calculate (optional if loglam=2)
starspec1 - (12) spectrum of the primary star (optional if $lunt1 > 0$)
starspec2 - (13) spectrum of the secondary (optional if $lunt2 > 0$)
starspec3 - (14) spectrum of the 'third body' (optional if $lunt3 > 0$)

albedo1 - (12) albedo of the primary star (optional if $ialbst = 1$ and $irrst = 1$)
albedo2 - (13) albedo of the secondary (optional if $ialbcp = 1$ and $irrcp = 1$)
dust_opac - (12) dust opacities (optional if $imie > 0$)
mie_phase - (13) dust phase functions (optional if $imiepf = 1$)
gas_opac - (16) molecular cross-section table (optional if iopac=1)
chem_eq_tab - (16) molecular population table (optional if iopac=1)

5.1. shellspec.in

This is the main input file where you can describe the geometry, dynamics and optical properties of the moving medium which we often refer to as a shell or a model. However, it can consist of many objects of various types (unfortunately, one of them is also called a shell but the difference should be obvious from the context). Various numerical and technical details are also specified in this file. Imagine your model as being frozen in one instant of time, overlay it with the Cartesian coordinates (x'', y'', z'') as seen in Figure 2 so that you have a convenient object, e.g., a hot star in the center and z'' corresponds to the intrinsic axis of revolution of the model and describe its properties (state quantities and velocity field) using these coordinates in this file. [3] Various transparent and nontransparent objects can be defined here. These include: STAR, COMPANION, SPOT, STREAM, RING, DISK, ENVELOPE, NEBULA, FLOW, JET, UFO, SHELL, BACKGROUND. Objects mentioned above were ranked according to their priority. If some of them happen to overlap in the space then higher priority objects will override the values defined by the lower priority objects. Most of these objects can be made nontransparent blackbody, dark matter, or empty space by setting their density to unrealistic values within certain density intervals. There is also a way to ascribe an intrinsic spectrum to nontransparent objects using parameters $lunt1, lunt2$ and $lunt3$.

Code can handle models with quite large optical depths. However, note that the treatment of the scattered light assumes that the medium is optically thin, more precisely that there is no significant obstruction between the source of the light (two stars) and scattering medium. Only the radiative transfer along the line of sight is solved. It means that objects could make eclipses along the line of sight but would cast no shadows into other directions. To allow for shadows the user can specify (in some objects) a type of shadow at a certain space point so that only scattering from the unobscured source (star) is taken into account. The objects that can be used to construct your model from are described below. The user can activate only some of them and can ignore the rest of objects.

[3] You can also load a precalculated model (e.g. from a 3D hydro-simulations) from an external file.

5.1.1. STAR

a central nontransparent object which can rotate as a solid body or possess a differential rotation with an optional inclination of the rotational axis and have a net space velocity. Can be treated as a blackbody or have its own spectrum. May be of the spherical or Roche shape. Limb and gravity darkening can be applied to it. Spherical star may have a circular spot on the surface of different temperature at a fixed location. Irradiation effect on its surface (from the COMPANION) can be considered. The light scattered in the circumstellar medium which originates from this object can be taken into account (neglecting its rotation, irradiation effect and assuming spherical shape). Differential rotation applies only to the spherical shape and does not affect the location of the spot. It depends on the latitude ϕ and angular velocity at the equator ω_e and pole ω_p. It may be smooth or a step function:

$$\begin{array}{l} \omega(\phi) = \omega_e - (\omega_e - \omega_p)\sin^2\phi \quad or \\ \omega(\phi) = \omega_e \quad for \quad \phi < \phi_0, \qquad \omega(\phi) = \omega_p \quad for \quad \phi > \phi_0. \end{array} \tag{135}$$

Designed to model mainly hotter or more luminous stellar components as well as extrasolar planets.

5.1.2. COMPANION

a nontransparent object which can rotate as a solid body with an optional inclination of the rotational axis and have a net space velocity. Can be treated as a black body or have its own spectrum. May be of the spherical or Roche shape. Limb and gravity darkening can be applied to it. Irradiation effect on its surface (from the STAR) can be considered. The light scattered in the circumstellar environment originating from this object can be taken into account (neglecting its rotation, irradiation effect and assuming spherical shape). Designed to model mainly a secondary (cooler or fainter component of a binary system).

5.1.3. SPOT

a spherical object which can rotate as a solid body with an optional inclination of the rotational axis and have a net space velocity. Designed to model mainly spots on accretion discs, direct impact regions, rotating circum-stellar (circum-binary) shells or third bodies.

5.1.4. STREAM

has the shape of a cylinder or cone with velocity varying linearly along the cone. Stream may rotate or be a subject of some rotational drag. It may also have a net space velocity so that it can e.g. follow the movement of some associated

star. Subsequently, its velocity field is described in the code by the following three terms:

$$\boldsymbol{v} = \boldsymbol{v}_i + \boldsymbol{v}_r + \boldsymbol{v}_n, \tag{136}$$

where $\boldsymbol{v}_i, \boldsymbol{v}_r, \boldsymbol{v}_n$ are its intrinsic velocity field, the rotational component, and the net space velocity, respectively. The intrinsic field is described by this equation:

$$\boldsymbol{v}_i = \frac{\boldsymbol{d}}{|\boldsymbol{d}|}[(v_2 - v_1)\frac{t}{|\boldsymbol{d}|} + v_1], \tag{137}$$

where $\boldsymbol{d} = \boldsymbol{r}_2 - \boldsymbol{r}_1$ is vector pointing from the starting point, $\boldsymbol{r}_1$, to the end of stream point, $\boldsymbol{r}_2$, t is distance along the stream

$$t = \frac{(\boldsymbol{r} - \boldsymbol{r}_1).\boldsymbol{d}}{|\boldsymbol{d}|} \tag{138}$$

and v_1, v_2 are non zero velocities at the start and end, respectively. Rotation component is:

$$\boldsymbol{v}_r = \boldsymbol{\omega} \times \boldsymbol{r}, \tag{139}$$

where $\boldsymbol{\omega}$ is angular velocity vector and $\omega = 2\pi/P$, where P is rotational period. If the center of rotation, $\boldsymbol{r}_0$, is different from the center of coordinates, one may set $v_n = -\boldsymbol{\omega} \times \boldsymbol{r}_0$ to take it into account. Note that, in the present version, the streamlines are made paralel to the main direction of the stream. If this is an essential drawback of the model then consider the jet option. Density of the stream changes along the stream to satisfy the continuity equation with some modification to allow for modelling of additional phenomena.

$$\rho(t) = \rho(0)\left(\frac{r_1}{r}\right)^2 \frac{v_1}{v} e^{t/R_\odot * edensm} \tag{140}$$

where $r_1, v_1, \rho(0)$ are radius, velocity, and density at the beginning of the stream. Gas and dust temperatures are constant. Designed to model the mass transfer streams, outflows, or shadows.

5.1.5. RING

is a circular ring or part of the ring (arc) with optional inclination and location. Mass in the center determines its Keplerian velocity. However, the velocity is uniform throughout the crosssection. The crossection of the ring, C, has shape of a rectangle and may vary along the arc. Density, dust density and electron number density may change along the arc to satisfy the continuity equation and/or additional phenomena:

$$\rho(t) = \rho(0)\frac{C(0)}{C(t)}[|t - t_0|/\pi + 1]^{edenrg}, \tag{141}$$

$$n_e(t) = n_e(0)\frac{C(0)}{C(t)}[|t - t_0|/\pi + 1]^{edenrg}, \tag{142}$$

$$\rho_d(t) = \rho_{d1}(0)\frac{C(0)}{C(t)}[|t-t_0|/\pi+1]^{edenrg} + \rho_{d2}(0)\frac{C(0)}{C(t)}[|t-t_0|/\pi+1]^{ede2rg}, \quad (143)$$

or

$$\rho(t) = \rho(0)\frac{C(0)}{C(t)}e^{|t-t_0|/\pi * edenrg}, \quad (144)$$

$$n_e(t) = n_e(0)\frac{C(0)}{C(t)}e^{|t-t_0|/\pi * edenrg}, \quad (145)$$

$$\rho_d(t) = \rho_{d1}(0)\frac{C(0)}{C(t)}e^{|t-t_0|/\pi * edenrg} + \rho_{d2}(0)\frac{C(0)}{C(t)}e^{|t-t_0|/\pi * ede2rg}. \quad (146)$$

Where t is angle along the arc in radians, $\rho(0), n_e(0), \rho_{d1}(0), \rho_{d2}(0), C(0), t_0$ are the density, electron number density, two component dust densities, crosssection and angle at the beginning of the arc. $edenrg, ede2rg$ are density exponents to model additional phenomena e.g. dust destruction etc. Gas and dust temperatures are constant. Designed to model rings around objects, arcs, comets.

5.1.6. DISC

is a disk-like structure surrounding a massive object with the mass of M_{dc}. It may have optional location and inclination. It has either the shape of a rotating wedge (space complement to two opposite cones) or of a slab, or of a rotational ellipsoid. It is farther constrained by two surfaces: its inner spherical surface with radius r_{in} and outer spherical or ellipsoidal surface with radius r_{out}. The velocity field $\boldsymbol{v}$ of the disc adopted depends on M_{dc} and is Keplerian within the disc plane, namely:

$$\omega(r) = \sqrt{G\frac{M_{dc}}{r^3}} \qquad \boldsymbol{v} = \boldsymbol{\omega} \times \boldsymbol{r} \quad (147)$$

where G is the gravitational constant, $\boldsymbol{\omega}, \boldsymbol{r}$ are the angular velocity and radius vectors, respectively. Densities may vary in the radial direction as a power law:

$$\begin{aligned} \rho(r) &= \rho(r_{in})(r/r_{in})^{edendc} \\ n_e &= n_e(r_{in})(r/r_{in})^{edendc}, \\ \rho_d(r) &= \rho_d(r_{in})(r/r_{in})^{edendc}\ . \end{aligned} \quad (148)$$

However, the electron number density can be calculated from the gas density, temperature, and the chemical composition assuming LTE. Temperature may be either constant, or have a radial power law dependence:

$$T(r) = T_{dc}(r/r_{in})^{etmpdc}, \quad (149)$$

where T_{dc} is temperature at inner disk radius r_{in}, or obey the radial accretion disk structure (Pringle 1981):

$$T(r) = T_{dc}\left(\frac{R_{dc}}{r}\right)^{\frac{3}{4}}\left(1-\sqrt{\frac{R_{dc}}{r}}\right)^{\frac{1}{4}}. \quad (150)$$

Here, T_{dc} is the characteristic disk temperature, and R_{dc} is radius of the central object. Note that maximum disc temperature in this representation is about $0.488T_{dc}$ and is reached at $r = (49/36)R_{dc}$. Designed to model mainly accretion discs.

5.1.7. ENVELOPE

is an object enclosing the central STAR (or STAR and COMPANION). It is subject to the Roche shape and can be detached or contact. It rotates as a solid body, does not have limb nor gracity darkening, has constant temperature and densities. It is possible to constrain this structure within a certain height above and belove the orbital plane. Designed to model envelopes and common envelopes of e.g. W UMa type stars.

5.1.8. NEBULA

is another disk-like structure. It is located around the central object with mass M_{nb}, has a Keplerian rotation (similar but not identical to the above mentioned DISK), and may also have a net space velocity. It is defined in the cylindrical coordinates (rc, z). Surface density, Σ, varies as a power law with the distance from the center:

$$\Sigma(rc) = \Sigma(rc_{in})(rc/rc_{in})^{ednnb}. \tag{151}$$

Mid-plane density, ρ_0, is then determined (as a function of the distance) from the surface density and the vertical scale height, H, using the relation

$$\Sigma(rc) = \sqrt{2\pi}H(rc)\rho_0(rc). \tag{152}$$

Vertical density behaviour is Gaussian:

$$\rho(rc, z) = \rho_0(rc)exp(-\frac{z^2}{2H^2}). \tag{153}$$

There is an option to modify this steep vertical drop in the density to mimic e.g. a disk wind or adopt a precalculated vertical density dependence in the form of a table. Electron number density and the dust density are proportional to the gas density

$$n_e(rc, z) = \frac{n_e(rc_{in}, 0)}{\rho(rc_{in}, 0)}\rho(rc, z), \qquad \rho_d(rc, z) = \frac{\rho_d(rc_{in}, 0)}{\rho(rc_{in}, 0)}\rho(rc, z). \tag{154}$$

Alternatively, the electron number density can be calculated from the gas density, temperature, and chemical composition assuming LTE. Temperature structure in the radial direction is similar to DISC and may be either constant, or a power law

$$T(rc) = T_{nb}(rc/rc_{in})^{etmpnb}, \tag{155}$$

where T_{nb} is the temperature at the inner disk radius rc_{in}, or may have the following radial stratification:

$$T(rc) = T_{nb}\left(\frac{R_{nb}}{rc}\right)^{\frac{3}{4}}\left(1-\sqrt{\frac{R_{nb}}{rc}}\right)^{\frac{1}{4}}. \tag{156}$$

Here, T_{nb} is the characteristic disk temperature (Pringle 1981), and R_{nb} is the radius of the central object. There is an option to introduce a simple step function or linear vertical gas temperature dependence (e.g. thermal inversion).

Vertical scale height is a function of the Keplerian velocity, v, and the sound speed, c_s:

$$H(rc) = hcnb\frac{c_s(rc)}{v(rc)}rc. \tag{157}$$

The *hcnb* is an optional free parameter that might be adjusted to fit the observations or modify the model. The speed of sound is:

$$c_s(rc) = \sqrt{\gamma kT(rc)/\mu} \tag{158}$$

where $\gamma = 5/3, 7/5, 1$ for monoatomic, diatomic, or a more complicated molecule with many degrees of freedom, respectively. k is Boltzmann constant. Mean molecular weight is calculated assuming solar chemical composition and a mixture of $H_2 + He$:

$$\mu = \frac{45\times 2 + 10\times 4}{45+10}m_H \tag{159}$$

where m_H is mass of a hydrogen atom.

The velocity field is calculated like this:

$$\omega(rc) = \sqrt{G\frac{M_{nb}}{rc^3}} \qquad \boldsymbol{v} = \boldsymbol{\omega}\times\boldsymbol{r} = \boldsymbol{\omega}\times\boldsymbol{rc} \tag{160}$$

where G is the gravitational constant, $\boldsymbol{\omega}, \boldsymbol{r}$ are the angular velocity and radius vectors, respectively.

This is another option to model flared accretion or protoplanetary disks.

5.1.9. FLOW

is identical to the STREAM but has a lower priority. Designed to model the mass transfer streams, outflows, holes, shadows, or structures symmetric to the STREAM.

5.1.10. JET

has the shape of one or two opposite cones emerging from a center. It allows for optional location and inclination and is farther limited by its inner, r_{in}, and

outer, r_{out}, radii. The velocity field is either polynomial:

$$v(r) = v_{jt} \left(\frac{r}{r_{in}} \right)^{eveljt} \tag{161}$$

where v_{jt} is the velocity at the inner radius or:

$$v(r) = v_{jt} \left(1 - \frac{r_c}{r} \right)^{eveljt} \tag{162}$$

where v_{jt} is the velocity at the infinity. The velocity has only the radial component and the stream lines diverge accordingly. r_c is a free parameter which may be associated with the radius of the central star. It may have a net space velocity so that it can move e.g. with the central STAR. Gas and dust temperatures behave as a power law.

$$T(r) = T_{jt} \left(\frac{r}{r_{in}} \right)^{etmpjt} \tag{163}$$

where T_{jt} is the velocity at the inner radius. Densities vary along the jet to satisfy the continuity equation.

$$\rho(r) = \rho(r_{in})(r_{in}/r)^2 v(r_{in})/v(r), \tag{164}$$

where $\rho(r_{in})$ are gas, dust, or electron (number) densities at the inner radius. Designed to model mainly jets or, e.g., 'shadows' cast by a cool, extended secondary from a more compact hot primary.

5.1.11. UFO

is identical to the DISC but has lower priority. Intended to model an extension or atmosphere of the DISC or a second disc.

5.1.12. SHELL

has the shape of a shell surrounding the central object. A few velocity fields are built in:

$$v(r) = v_{sh} \quad , \quad v(r) = v_{sh} \left(\frac{r}{r_{in}} \right)^{evelsh} \quad , \tag{165}$$

$$v(r) = v_{sh} \left(1 - \frac{r_c}{r} \right)^{evelsh} \tag{166}$$

while the temperature is kept fixed and densities are either constant or satisfy the continuity equation (see 'shellspec.in' for more information). It may have a net space velocity.

5.1.13. BACKGROUND

is designed to add more flexibility to the code and to fill the region not occupied by any of the previous objects at least with a uniformly radially expanding medium when necessary. It may help to reduce a numerical noise too by the appropriate choice of state quantities.

5.1.14. Description of individual parameters

A detailed description of all the parameters and their units from shellspec.in follows. It can also be found in the end of the 'shellspec.in' file to facilitate the orientation in such "a jungle" of free parameters.

```
!       Definition of the input quantities:
!       alam1, alamn, alams -start, end and step of wavelength in [A]
!         Note that if loglam=0 the gas continuum opacity is calculated
!         only at alam1 & alamn and then interpolated.
!         So keep the iterval <alam1, alamn> short enough in that case.
!       loglam=0 equidistant step in lambda. Good for short interval.
!         H continuum opacity is calculated only at alam1 & alamn
!         and interpolated for given lambda to speed up calculations.
!       loglam=1 equidistant step in log(lambda), for long intervals.
!         Number of steps will be the same as for loglam=0.
!         H continum opacity is calculated at each lambda.
!       loglam=2 lambda's are read from file lambda. It sets nfreq.
!         Good for comparison with the observations.
!         H continum opacity is calculated at each lambda.
!         (Dust opacity is interpolated to lambda independ. of loglam)
!       cutoff - extension of the <alam1,alamn> interval in [A] when
!                reading the gas_opac table. Assuming that
!                broadening by the velocity field dominates:
!                cutoff>maximal radial velocity/c*lambda
!       imodel=1  calculate your own input shell model
!       imodel=2  read input shell model from 'shellspec.mod'.
!                You can ignore most of input below defining geometry,
!                the velocity field and state quantities of objects but
!                you must still input the data for the scattering:
!                  rstar,tstar,vxst,vyst,vzst
!                for the coordinate rotation:
!                  temp0,ane0,xcp,ycp,zcp
!                and for the limb darkening:
!                  istar,rstar,tstar,dlst,dlst2,
!                  icomp,rcp,tempcp,dlcp,dlcp2,xcp,qq
!                and switches: lunt1,lunt2,lunt3,ithom,irayl,
!                  imie,imiepf,ihyd,iopac,iline,eps
!       irotat -option of interpolation from the body frozen grid
!                to the line of sight grid during the coord. rotation
!                0=linear interpolation, good for continuous fields,
!                  otherwise the result may depend on discontinuities
!                  or background (temp0,ane0,...)
!                  does not support shadows (sets lshade=3)
!                1=nearest neighbour approximation, may be less smooth
!                  but can handle discontinuities and shadows
!       ipart  -option of partition functions
!                [1-built in Dworetsky & Smalley, 2-Irwin]
!                (only ipart=1 is implemented so far)
!       ichemc -option of abundances, if ielnd=1 then ichemc=1
!                [0-default solar, 1-read from file 'abundances']
!       ielnd=1 electron number densities provided in the input model
!                are ignored and code calculates el.num.dens.
!                assuming LTE, from known temperature, density and
```

```
!               chemical composition. File 'abundances' is read and must
!               contain 3.column which specifies which elements are
!               considered in Ne calculations, this sets ichemc=1
!       ielnd=0 electron number densities are known apriori and are
!               specified in the input model
!       ithom=0 Thomson scattering is off
!       ithom=1 Thomson scattering from stars is on
!               (assumes optically thin environment)-check also shadows
!       irayl=0 Rayleigh scattering on neutral hydrogen is off.
!               If Lyman lines are treated explicitely in the linelist
!               set irayl=0 not to count the contribution twice
!       irayl=1 Rayleigh scattering from stars on neutral hydrogen is on
!               (assuming optically thin environment)-check shadows
!       imie=0  Mie scattering and absorption on dust is off
!       imie=1  Mie scattering+absorption opacity is on.
!               Several species or input files can be included.
!               dust_opac file with tables must be provided.
!               Mie thermal and scattering emissivity on dust is on.
!               It is scattering of light from the stars assuming
!               optically thin medium.
!               Scattering emission can be isotropic or
!               non-isotropic (see imiepf).
!       imie=2  Mie scattering+absorption opacity is on
!               Mie thermal emissivity is on, but
!               Mie scattering emissivity is off
!       imie=3  Mie scattering+absorption opacity is on
!               Mie thermal emissivity is on
!               Mie scattering emissivity is on but is isotropic and
!               assumes J=B(T) i.e. it is not scattered light from stars
!               Check for shadows.
!       imiepf  angular dependence of the scattered light from stars,
!               has an effect only if imie=1
!       imiepf=1 angular dependent scattering emissivity,
!               reads extra table with phase functions (mie_phase),
!               otherwise it is isotropic
!               In case there are several species in dust_opac
!               this will redistribute the total scattering opacity.
!       ihyd=1  hydrogen bound-free and free-free opacity is turned on
!               assuming only atomic H (no molecules)
!       iopac=1 additional tabulated gas true opacity is added
!               reads extra table with gas opacities (no scattering)
!       iline=0 No line opacity
!       iline=1 line opacity is included. Spectral line parameters must
!               be specified in the file 'line.dat'
!       eps -artificial number <0.,1.> for test purpose which splits
!               the line opacity (emissivity) into the true
!               absorbtion (eps->1.) and coherent scattering (eps->0.).
!               In LTE eps=1. ( S=eps*B+(1-eps)*J )
!               If ithom=irayl=0 set also eps=1. for consistency
!       ionu, ior, iot -sequential indexes of frequency, x, and y point
!               for which you want a more detailed output along the line
!               of sight (specified by x,y)
!       offset -vertical shift applied to the normalized spectra output
!               to plot many spectra from different rotation phases
!       phase1, phasen - start, end of the phase interval you want
!               to cover [deg] (e.g. if xcp>0,ycp=zcp=0, dinc=90 then
!               phase1=-90 will start from the primary eclipse)
!       nphase -number of rotations (different view points) within
!               the interval above
!               if nphase=0 then phase1 and phasen are ignored, and it
!               reads one column from the file 'phases' with phases.
!               These are values <0,1> and count from the x axis
!               so that phase=0.0 or 1.0 is primary eclipse
!               if xcp>0,ycp=zcp=0, dinc=90
!       dinc   -angle between rotation axis of the model and the line
```

```
!               of sight [deg], dinc=90.0 is edge on.
!       dd   -distance from the Earth in [pc]
!       vgamma -gamma velocity. It is applied only to the spectra and
!               lightcurves in the end
!               (independently on the phase/view point)
!               2D images correspond to the original=final lambda(ionu)
!       iext=0  no reddening/extinction
!       iext=1  reddening according to Cardelli, Clayton & Mathis 1989
!               it is applied to 2D images, spectra & lightcurves
!       rv      =A(V)/E(B-V)
!       ebv     =E(B-V) [mag]
!----------------intrinsic spectra specifications:
!       lunt1=0 all objects with density from <dcut1,dcut2> interval are
!               nontransparent blackbodies with the same temperatures as
!               in the case of transparency.
!       lunt1>0 all objects with density within <dcut1,dcut2> are
!               nontransparent and have an intrinsic intensity spectrum.
!               The spectrum is read from file 'starspec1'.
!       lunt1=1 the x,y column input required with wavelength [A] and
!               H_lambda flux [erg/cm^2/s/A] (as an output of SYNSPEC)
!       lunt1=2 the x,y column input required with wavelength [A] and
!               I_nu intensity [erg/cm^2/s/Hz/sterad]
!       lunt1=3 the 4 column input required with idummy,frequency [Hz],
!               dummy, F_nu flux [erg/cm^2/s/Hz]
!               (output of coolTlusty, unit 21, first 2 rows are dummy)
!       xunt1  -multiplication factor applied to starspec1 x-column
!               if it is not in the correct-required units
!               (otherwise set it =1.)
!       yunt1  -multiplication factor applied to starspec1 y-column
!               if it is not in the correct-required units
!               (otherwise set it =1.)
!       lunt2,xunt2,yunt2 -the same meaning as above except that these
!               deal with density interval <dcut2,dcut3> and
!               the spectrum is read from file 'starspec2'.
!       lunt3,xunt3,yunt3 -the same meaning as above except that these
!               deal with density interval <dcut3,dcutn> and
!               the spectrum is read from file 'starspec3'.
!-----------definitions of grids:
!       There are two main grids:
!         -body frozen grid - defines your model 'frozen' at a moment
!         -line of sight grid - defines grid for rad.transfer calcul.
!       Each grid can be composed of two subgrids: A,B
!       If subgrids overlap then the denser subgrid has priority.
!       Subgrids A,B are merged into one grid.
!       rmdfx1<rmdfx2, rmdfy1<rmdfy2, rmdfz1<rmdfz2 - define
!         the box A in the body frozen frame (if imodel=1) [R_sol]
!       rmdfx3<rmdfx4, rmdfy3<rmdfy4, rmdfz3<rmdfz4 - define
!         the box B in the body frozen frame (if imodel=1) [R_sol]
!       stepfxyz>0.,stepfxyzb>0. -is a mean distance between the x,y,z
!         grid points of box A,B respectively [R_sol]
!         It determines the number of grid points:
!       nbodf1, nbodf2, nbodf3 -number of grid points in x, y, z
!         direction in body frozen coordinates of the model.
!         (Points are overridden
!         by the values from 'shellspec.mod' if imodel=2)
!       gainfxyz, gainfxyzb -grid step multiplication factors
!         of the body frozen grids A,B to allow for logarithmic grid
!         [gainfx=(x_{i+1}-x_{i})/(x_{i}-x_{i-1})]
!         e.g. gainfx=1. for equidistant step
!         gainfx>1. step increases symetrically from the middle to
!         the left and to the right
!       rmdx1<rmdx2, rmdy1<rmdy2, rmdz1<rmdz2 - define the box A
!         of the observer's line of sight frame all in R_sol.
!       rmdx3<rmdx4, rmdy3<rmdy4, rmdz3<rmdz4 - define the box B
!         of the observer's line of sight frame all in R_sol.
```

```
!         Observer looks along the opposite z-direction.
!       stepxyz>0.,stepxyzb>0. -is a mean distance between the x,y,z
!         grid points in box A,B respectively [R_sol]
!         They determine the number of grid points:
!       nbod1, nbod2, nbod3 -number of grid points in x, y, z
!         in the line of sight observer's frame.
!       gainxyz, gainxyzb -grid step multiplication factors
!         (common ratio of the geometric sequence) of the line of sight
!         grids A,B respectively [gainx=(x_{i+1}-x_{i})/(x_{i}-x_{i-1})]
!         e.g. gainx=1. for equidistant step
!         gainx>1. step increases symetrically from the middle to
!         the left and to the right
!       To turn off the B grids simply set their dimension to zero e.g.
!         rmdx3=rmdx4, rmdy3=rmdy4, rmdz3=rmdz4
!-----------------object definitions:
!       istar,icomp,ispot,ism,iring,idisc,ienv,
!       inebl,iflow,ijet,iufo,ishell
!       These are main on/off switches for the objects.
!       They are ordered according to priority.
!       Priority is determined in smod1.
!       It is important in case objects happen to overlap.
!-------primary star (central object)----------
!       istar=0 accompanied by rstar=0 will switch off the primary
!       istar=1 central object is a nontransparent uniformly rotating
!         sphere. Its density is set to <dcut1,dcut2>. It can be either
!         black body with T=tstar if lunt1=0 or has its intrinsic
!         intensity spectrum if lunt1>0. In case of scattering or
!         reflection of its light by other objects its rotation is
!         ignored.
!         Code ignores: dgst,ffst,qq
!       istar=2 central object is a detached component of a binary.
!         It has a Roche shape defined by ffst<=1, synchronous rotation,
!         is nonstrasparent with density within <dcut1,dcut2>.
!         It can be either black body with T=tstar if lunt1=0 or
!         has its intrinsic intensity spectrum if lunt1>0.
!         You must also set: xcp>0,qq>0
!         Code also calculates/ignores: xstar,ystar,zstar,vrotst
!         ,drotst,hst,rstar
!       istar=3 central object is a figure 8 contact system. It has
!         a Roche shape defined by 1<ffst<=2, synchronous rotation,
!         is nonstrasparent with density within <dcut1,dcut2>.
!         It can be either black body with T=tstar if lunt1=0 or
!         has its intrinsic intensity spectrum if lunt1>0.
!         You must also set: xcp>0,qq>0
!         Code also calculates/ignores:
!           xstar,ystar,zstar,vrotst,drotst,hst,rstar,icomp
!       if istar>1 or icomp>1 or (istar>0 and icomp>0 and vxst>clight)
!         then code calculates (from emstar,xcp,qq):
!         ycp,zcp,vxst,vyst,vzst,vxcp,vycp,vzcp
!         assuming circular orbit.
!       rstar -radius of the central star in [R_sol]
!         if istar>1 (Roche Geometry) this value will be used for
!         scattering in the circumstellar matter and irradiation effect
!         on the companion which use spherical approximation
!       tstar -effective temperature of the central star in [K]
!         without gravity darkening and irradiation. This value will
!         be used for scattering in the circumstellar matter (in case
!         of black body) and irradiation effect on the companion
!         if istar=2 it is the temperature at the rotation pole
!         if istar=3 it is the temperature at the rotation pole of
!         the more massive star
!       emstar -mass of the central star in [M_sol]
!       xstar,ystar,zstar -define unit aiming vector of the rotational
!         axis of the central star
!       vrotst -equatorial rotation velocity of the central star [km/s]
```

```
!         in case istar=1 corresponding to the equatorial angular vel.
!       idifst -on/off differential rotation only for istar=1
!       idifst=0 no differential rotation
!       idifst=1 smooth differential rotation
!         omega(phi)=omega_eq-(omega_eq-omega_pol)*sin(phi)**2
!       idifst=2 step function differential rotation
!         omega(phi)=omega_eq  for z/rstar<hst
!         omega(phi)=omega_pol for z/rstar>hst
!       drotst - the ratio of angular velocity at the rotation pole to
!         the angular vel. at the equator: drotst=omega_pol/omega_eq.
!       hst -break in the step function =z/rstar for idifst=2
!       vxst, vyst, vzst -net velocity components
!         of the center of mass of the central star [km/s]
!         (if vxst>clight and istar>0 and icomp>0 then see istar)
!       dlst -limb darkening coefficient of the central star
!       dlst2 -second limb darkening coefficient
!            I=1-dlst*(1-mu)-dlst2*(1-mu)**2
!       dgst -gravity darkening coefficient (beta) of the central star
!         (0.25 for radiative, 0.08 for convective atmospheres)
!         It is dummy if istar=1.
!       ffst<=1 -Roche lobe fill-in factor of the primary. Its is
!         the distance of the inner substellar point of the primary
!         (between the stars) from the center of the primary relative
!         to the distance to L1, the Roche lobe is reproduced if ff=1
!       1<ffst<=2 -Roche lobe fill-out factor of the contact system
!               ffst=(C1-C)/(C1-C2)+1
!               It is dummy if istar=1.
!       irrst=0 -irradiation and reflection effect is off
!           (ialbst,albst,htst,htsta have no meaning in this case)
!       irrst=1 -irradiation of the object from the companion is on.
!           Irradiation (heating) applies only if istar=1,2.
!           Reflection of the sp. of companion applies if istar=1,2
!           (rcp,tempcp>0 ... are presumed).
!       ialbst=1 monochomatic albedo is red from file=albedo1
!           (if irrst=1). It should be compatible with Bond albedo.
!       albst  -Bond albedo <0,1>
!       htst   -heat redistribution parameter in case of irradiation,
!           fraction of the heat absorbed on the day side which is
!           redistributed over the day-night sides, <0,1>,
!           0-nothing is redistributed and nothing goes to the night,
!           1-all the energy (which is not reflected) impinging on
!           the planet is evenly distributed over the day-night sides.
!           It is analoguous to the so called Pn parameter of A.Burrows
!           (a fraction of the irradiating energy impinging on
!           the day side which is transfered to and irradiated from
!           the night side), Pn=(1-albst)*htst/2
!       htsta  -degree of the inhomegenity of the heat transport, <0,1>.
!           1-homegeneous, 0-cosine dependence
!           T**4=T0**4(htsta+4(1-htsta)/pi*cos_latitude)
!       ispst=1/0 will turn on/off a spot on the star if istar=1
!           (it has the shape of a circle)
!       xspst,yspst,zspst -define unit aiming vector of the location
!           of the spot center on the surface
!       aspst -angular radius of the spot in [deg]
!       tspst -ratio of the spot temperature to the ambient temperature
!           (i.e. temperature accounting for the reflection effect...)
!-------
!       temp*,dens*,ane* - state quantities in various objects
!               temperature, density, electron number density [K,CGS]
!       vtrb* - microturbulent velocity in various objects [km/s],
!               it does not apply to nontransparent objects
!       dstd* - density of dust in various objects [g/cm^3]
!               you must also set dens*>0. to have an effect
!       dstt* - temperature of dust in various objects [K]
!               dust temperatures must be higher than the condensation
```

```
!               temperatures of the species (see dust_opac)
!               to have an effect
!       ishd* - define shadows for scattering (in some objects)
!             =0 no scattering
!             =1 scattering from central star only
!             =2 scattering from secondary star only
!             =3 scattering from both stars
!-------companion or secondary star
!       icomp=0 secondary off
!       icomp=1 secondary on, it is a uniformly rotating nontransparent
!               sphere. It may be a blackbody with T=tempcp if lunt2=0
!               or has its own spectrum if lunt2>0. Its density is set
!               to <dcut2,dcut3>. Code ignores: dgcp,ffcp,qq
!       icomp=2 secondary is a detached component of a binary.
!         It has a Roche shape defined by ffcp<=1, synchronous rotation,
!         is nonstrasparent with density within <dcut2,dcut3>.
!         It can be either black body with T=tempcp if lunt2=0 or
!         has its intrinsic intensity spectrum if lunt2>0.
!         You must set: xcp>0,qq>0,emstar>0
!         Code also calculates/ignores: vrxcp,vrycp,vrzcp,vrotcp,rcp
!       rcp  -radius of the spherical companion [R_sol],
!         if icomp=2 this input is used only for the scattering
!         and irradiation from the object otherwise it is superfluous
!       tempcp -see primary star above, this value is used for
!         the scattering on the circumstellar material and irradiation
!         of the primary [K]
!       qq -mass ratio (companion/star), important only for Roche geom.
!           if istar>1 or icomp>1
!       vrxcp, vrycp, vrzcp -define unit aiming vector of the rotational
!         axis of the secondary star (companion)
!       vrotcp -equatorial rotation velocity of the companion [km/s]
!       xcp,ycp,zcp -location of the center (of mass) of
!               the companion [R_sol]
!       vxcp,vycp,vzcp -components of the velocity vector of the center
!               (of mass) of the companion [km/s]
!       dlcp -limb darkening coefficient of the secondary star
!       dlcp2 -second limb darkening coefficient (the same as dlst2)
!       dgcp -gravity darkening coefficient (beta) of the secondary
!       ffcp<=1 -Roche lobe filling factor of the secondary is
!         the distance of the inner substellar point of the secondary
!         from the center of the secondary relative to 1-L1,
!         the Roche lobe is reproduced if ffcp=1
!       irrcp=0 -irradiation and reflection effect is off
!           (ialbcp,albcp,htcp,htcpa have no meaning in this case)
!       irrcp=1 -irradiation of the secondary from the primary is on.
!           Irradiation (heating) applies only if icomp=2.
!           Reflection (of the spectrum of primary) applies if icomp=1,2
!           (istar=1,2 and rstar,tstar>0 are presumed)
!       ialbcp=1  monochomatic albedo is red from file=albedo2
!           (if irrcp=1). It should be compatible with the Bond albedo.
!       albcp  -Bond albedo <0,1>
!       htcp   -heat transport parameter in case of the irradiation.
!           The same as htst, <0,1>.
!       htcpa  -degree of the inhomegenity of the heat transport, <0,1>.
!           1-homegeneous, 0-cosine dependence, the same as htsta.
!-------spot or third star
!       ispot=0 spot is off
!       ispot=1 spot is on, it is a uniformly rotating sphere
!       vrxsp, vrysp, vrzsp -define unit aiming vector of the rotational
!               axis of the spot
!       vrotsp -equatorial rotation velocity of the spot [km/s]
!       rsp  -radius of the spherical spot [R_sol]
!       xsp,ysp,zsp -location of the center of the spot [R_sol]
!       vxsp,vysp,vzsp -components of the velocity vector of the center
!               of the spot [km/s]
```

```
!       tempsp -constant temperature [K]
!       denssp -gas density [g/cm^3]
!       anesp -electron number density [cm^-3]
!       dstdsp -dust density [g/cm^3]
!               you must also set denssp>0. to have an effect
!       dsttsp -dust temperature [K]
!               dust temperatures must be higher than the condensation
!               temperatures of the species (see dust_opac)
!               to have an effect
!       vtrbsp -microturbulence [km/s]
!-------stream
!       ism=0/1  -stream off/on
!       v1sm   -stream velocity at the beginnig of stream [km/s]
!       v2sm   -stream velocity at the end of stream [km/s]
!         velocity is directed from beginning to end
!       r1sm    -radius of the stream at the beginning [R_sol]
!       r2sm    -radius of the stream at the end [R_sol]
!         notice that although the radius changes the streamlines
!         are parallel (contrary to jet)
!       x1sm,y1sm,z1sm -position of the beginning of the stream [R_sol]
!       x2sm,y2sm,z2sm -position of the end of the stream [R_sol]
!       vxsm, vysm, vzsm -net velocity [km/s]
!         you can use it also to mimic orbital drag or if the center
!         of rotation is not at the center of coordinates
!       xsm,ysm,zsm -rotational vector of stream
!       psm -rotational period of stream in days
!       tempsm -temperature [K], constant along the stream
!       denssm - gas density at the beginning [g/cm^3] and scales along
!         the stream to satisfy the continuity equation:
!         density=denssm*v1sm*r1sm**2/(vsm*rsm**2)*exp(t/rsol*edensm)
!         where t is distance along the stream and exp term allows
!         e.g. for a dust destruction
!       anesm - electron number density at the beginning [cm^-3],
!         similar to the density but if ielnd=1 then it is overriden by
!         the calculation from the state quantities
!       edensm -density dependence exponent to enable the modeling
!         of additional phenomena
!       dstdsm -dust density [g/cm^3], it changes along the stream like
!         the gas density
!               you must also set denssm>0. to have an effect
!       dsttsm -dust temperature [K], constant along the stream
!               dust temperatures must be higher than the condensation
!               temperatures of the species (see dust_opac)
!               to have an effect
!       vtrbsm -microturbulence velocity [km/s]
!-------ring
!       iring>0 ring is on
!       rrg -radius of the ring [R_sol]
!       emrg -mass in its center to calculate velocities [Msol]
!       b1rg, b2rg -specifies the arc from-to in [deg], b1rg><b2rg
!         The location of the zero angle is not simple to explain so
!         test it first or consult subroutine trans. In many cases
!         it will be along the x axis.
!       a1rg,a2rg -vertical half width of the ring at the beginning
!         and end of the arc in [R_sol]
!       dr1rg, dr2rg -horizontal half thickness the ring at
!         the beginning and end of the arc in [R_sol]
!         The crosssection, C, of the ring may vary along the arc and
!         is C1=4*a1rg*dr1rg at the beginning.
!       xrg, yrg, zrg -location of the center in [R_sol]
!       xpolrg, ypolrg, zpolrg -orientation of the polar axis
!       vxrg, vyrg, vzrg -net overall space velocity [km/s]
!       edenrg, ede2rg -density dependence exponent to enable
!         the modeling of additional phenomena. Density, dust density
!         and electron number density change along the ring (arc)
```

```
!         to safisfy continuity equation+additional phenomenon
!         e.g. destruction (lifetime) of dust grains along the arc.
!       dstdrg, dst2rg -dust density at the beginning (b1rg) [g/cm^3].
!              you must also set densrg>0. to have an effect
!       itrg=1 then
!         gas density=densrg*C1/C*[|t-b1rg|/pi+1]**edenrg
!         electron num. density=anerg*C1/C*[|t-b1rg|/pi+1]**edenrg
!         dust density=dstdrg*C1/C*[|t-b1rg|/pi+1]**edenrg+
!                 dst2rg*C1/C*[|t-b1rg|/pi+1]**ede2rg
!       itrg>or< 1 then
!         gas density=densrg*C1/C*dexp[|t-b1rg|/pi]**edenrg
!         electron num. density=anerg*C1/C*dexp[|t-b1rg|/pi]**edenrg
!         dust density=dstdrg*C1/C*dexp[|t-b1rg|/pi]**edenrg+
!                dst2rg*C1/C*dexp[|t-b1rg|/pi]**ede2rg
!         where t-is angle along the arc.
!       densrg -gas density at b1rg [g/cm^3]
!       anerg -electron number density at b1rg [cm^-3]
!       temprg -constant gas temperature [K]
!       dsttrg -constant dust temperature [K]
!              dust temperatures must be higher than the condensation
!              temperatures of the species (see dust_opac)
!              to have an effect
!       vtrbrg -microturbulence [km/s]
!-------disk (accretion disk around some object)
!       idisc=0 switch off the disc
!       idisc=1 disc has the shape of a rotating wedge
!              limited by inner and outer radii (spherical surfaces)
!              tip of the wedge is at the center of the object
!       idisc=2 disc has the shape of a slab
!              limited by inner and outer radii (spherical surfaces)
!       idisc=3 disc has the shape of a rotating ellipsoid
!              limited by inner spherical and outer ellipsoidal surface
!       adisc -angular halfwidth of the disc wedge [deg]
!             (if idisc=1)
!            -half of the thickness of the disc slab [R_sol]
!             (if idisc=2)
!            -semiaxis of the ellipsoid along the rotational axis
!             [R_sol] (if idisc=3)
!       rindc -inner radius of the disc [R_sol]
!       routdc -outer radius of the disc [R_sol] or
!              -semiaxis of the ellipsoid perpendicular to the rotation
!               axis, if idisc=3, [R_sol]
!       emdc -mass of the object in the disk center [M_sol]
!         it determines its Keplerian velocity
!       rdc  -radius of the object in the disk center [R_sol]
!         it determines its temperature structure if itdc=2
!       xdc,ydc,zdc -location of the disk center in [R_sol]
!       xdisc,ydisc,zdisc -components of the unit aiming vector of
!              the rotational axis of the Keplerian disc around emstar
!       vxdc, vydc, vzdc -net velocity components
!              of the center of the disc [km/s]
!       densdc -gas density at rindc [g/cm^3]
!       anedc  -electron num. density at rindc [cm^-3]
!       tempdc -characteristic gas temperature [K], see below
!       edendc -radial density dependence exponent
!              (dens, ane and dust density are a function of r)
!              Rho(r) ~ Ne(r) ~ densdc*(r/rindc)**edendc
!       itdc=1  gas & dust temperatures are constant
!       itdc=2  gas & dust temperatures are a function of r
!              gas:  T(r)=tempdc*(rdc/r)**0.75*(1-(rdc/r)**0.5)**0.25
!              dust: T(r)=dsttdc*(rdc/r)**0.75*(1-(rdc/r)**0.5)**0.25
!       itdc=3  gas & dust temperatures as a power law
!              gas:  T(r)=tempdc*(r/rindc)**etmpdc
!              dust: T(r)=dsttdc*(r/rindc)**etmpdc
!       etmpdc  -exponent of the radial temperature dependence
```

```
!       dstddc  -dust density at rindc [g/cm^3]
!               you must also set densdc>0. to have an effect
!       dsttdc  -characteristic dust temperature [K]
!               dust temperatures must be higher than the condensation
!               temperatures of the species (see dust_opac)
!               to have an effect
!       vtrbdc -microturbulence [km/s]
!-------envelope around the primary star
!       ienv,emen,ggen,ffen have similar meaning to istar,emstar,qq,ffst
!       ienv=2 envelope is on, has a detached Roche shape
!       ienv=3 envelope is on, has a contact Roche shape
!           (common envelope)
!       emen -mass of the central star [M_sol]
!       qqen -mass ratio (companion/star)
!       ffen<=1 -Roche lobe fill-in factor of the detached envelope.
!         It is radius of the substellar point of the envelope
!         relative to the radius of the L1. Roche lobe has ffen=1.
!       1<ffen<=2 -Roche lobe fill-out factor of the contact envelope
!               ffen=(C1-C)/(C1-C2)+1
!       hen -vertical limit [R_sol], limits the envelope in
!         the direction perpendicular to the orbital plane to z<+-hen
!       tempen -constant temperature [K]
!       densen -constant gas density [g/cm^3]
!       aneen -constant electron number density [cm^-3]
!       dstden -constant dust density [g/cm^3]
!               you must also set densen>0. to have an effect
!       dstten -constant dust temperature [K]
!               dust temperatures must be higher than the condensation
!               temperatures of the species (see dust_opac)
!               to have an effect
!       vtrben -microturbulence [km/s]
!-------nebula (protoplanetary disk/nebula around central object)
!              it is defined in cylindrical coordinates (r,z)
!       inebl not=1  -nebula is off
!       inebl=1 flared protoplanetary or accretion disk
!               vertical scale height is:
!               H(r)=hcnb*(gamma*k*T_gas/m)**0.5
!             Vertical structure:
!               fdens=dens0*dexp(-erz**2/H**2/2.d0)
!               dens0 is midplane density calculated from surface dens.
!               Density may have wind region.
!               Gas temperature may have vertical temperature inversion.
!             Radial structure:
!               surface density decreases ~(r/rinnb)**edennb
!               dust dens & electron num. dens are ~ density.
!               Gas and dust temperatures change with radius.
!       aneb    -vertical extent of nebula at particular r in [H]
!               extent(r)=+-aneb* H(r)
!       rinnb   -inner radius of the nebula [R_sol]
!       routnb  -outer radius of the nebula [R_sol]
!       emnb    -mass of the object in the nebula center [M_sol]
!       rnb     -radius of the object in the nebula center [R_sol]
!       iinvnb,hinvnb,tinvnb -describe vertical gas temp. inversion
!               (only if itnb=3).
!               There is no inversion in dust temperature.
!       hinvnb  -start of vertical gas temp. inversion in [H]
!               for z(r)>hinvnb*H(r) (except itnb=1 or 2)
!               hinvnb>aneb or itnb=1 or itnb=2 means no inversion
!       tinvnb  -temperature multiplication factor in the inversion
!       iinvnb=0 no inversion
!       iinvnb=1 step fuction inversion
!               gas temp(z,r)=temp0(r)*tinvnb
!       iinvnb=2 linear inversion from temp0 at hinvnb*H(r) to
!               tinvnb*temp0(r) at aneb*H(r)
!               temp0 is midplane gas temperature
```

```
!       hwindnb -start of the wind region in vertical scale-heights
!               rho(z)=rho(0)*dexp(-erz**2/H**2/2.d0)
!               but for z>hwindnb*H
!               rho(z)=rho(0)*dexp(-hwindnb**2/2.d0)
!               i.e. rho(z)=rho(hwindnb*H)= const
!               electron n.d. and dust density are proportional to gas
!               and thus will also have wind region
!               hwindnb>aneb will turn off the wind region
!       idennb  allows to use optional vertical density profile
!       idennb=1 reads file wind_prof with rho=f(z)
!               hwindnb is then ignored
!       hcnb    allows to multiply the classical vertical scale-height
!               by some factor
!       ishdnb  =0,1,2,3 (describes shadows for scattering)
!       hshdnb  -start of the scattering region in [H]
!               if z>hshdnb*H then kshade=ishdnb else kshade=0
!       xneb,yneb,zneb -components of the unit aiming vector of
!               the rotational axis of the Keplerian disc around emnb
!       vxnb, vynb, vznb -net velocity components
!               of the center of the nebula [km/s]
!       tempnb  -characteristic gas temperature [K]
!       itnb    -regulates temperature structure
!       itnb=1  nebula gas and dust temperatures are constant
!               gas temp=tempnb, dust temp=dsttnb (no inversion)
!       itnb=2  nebula gas and dust temp. are a function of r only
!               T(r)=tempnb*(Rnb/r)**0.75*(1-(Rnb/r)**0.5)**0.25
!               T(r)=dsttuf*(Rnb/r)**0.75*(1-(Rnb/r)**0.5)**0.25
!               no inversion
!       itnb=3  disc temperature as a power law (e.g. protopl. discs)
!               gas:  T(r)=tempnb*(r/rinnb)**etmpnb
!               dust: T(r)=dsttnb*(r/rinnb)**etmpnb
!               there may be a gas temperature inversion in z
!       etmpnb  -exponent of radial temperature dependence
!       densnb -gas density at rinnb (at midplane) [g/cm^3]
!       anenb  -electron num. density at rinnb (at midplane) [cm^-3]
!       edennb -radial density dependence exponent of surface density
!               (dens, ane and dust density are a function of r)
!               Ne(r,z) ~ Rho_dust(r,z) ~ Rho_gas(r,z)
!       dstdnb -dust density at rinnb [g/cm^3] (at midplane)
!               you must also set densnb>0. to have an effect
!       dsttnb -characteristic dust temperature [K]
!               dust temperatures must be higher than the condensation
!               temperatures of the species (see dust_opac)
!               to have an effect
!       vtrbnb -microturbulence [km/s]
!-------flow
!         it is identical to the stream but lower priority
!       iflow=0/1  -stream off/on
!       v1fw   -stream velocity at the beginnig of stream [km/s]
!       v2fw   -stream velocity at the end of stream [km/s]
!         velocity is directed from beginning to end
!       r1fw    -radius of the stream at the beginning [R_sol]
!       r2fw    -radius of the stream at the end [R_sol]
!         notice that although the radius changes the streamlines
!         are made paralel (contrary to jet)
!       x1fw,y1fw,z1fw -position of the beginning of the stream [R_sol]
!       x2fw,y2fw,z2fw -position of the end of the stream [R_sol]
!       vxfw, vyfw, vzfw -net velocity [km/s]
!         you can use it also to mimic orbital drag or if the center
!         of rotation is not at the center of coordinates
!       xfw,yfw,zfw -rotational vector of stream
!       pfw -rotational period of stream in days
!       tempfw -temperature [K], constant along the stream
!       densfw - is density at the beginning [g/cm^3] and scales along
!         the stream to satisfy the continuity equation:
```

```
!         density=densfw*v1fw*r1fw**2/(vfw*rfw**2)*exp(t/rsol*edenfw)
!         where t is distance along the stream
!       anefw - electron number density at the beginning [cm^-3]
!         similar to the density but if ielnd=1 then it is overriden by
!         the calculation from the state quantities
!       edenfw -density dependence exponent to enable the modeling
!         of additional phenomena
!       dstdfw -dust density [g/cm^3], it changes along the stream like
!         the gas density
!               you must also set densfw>0. to have an effect
!       dsttfw -dust temperature [K], constant along the stream
!               dust temperatures must be higher than the condensation
!               temperatures of the species (see dust_opac)
!               to have an effect
!       vtrbfw -microturbulence velocity [km/s]
!-------jet
!       ijet=0   switch off the jet
!       ijet=1  jet has only one -primary cone
!       ijet=2  jet has two cones: the primary cone and the opposite one
!       ajet -angle halfwidth of the jet cones [deg]
!         streamlines flare according to the opening angle
!       rinjt, routjt -radius boundaries of the jet cones [R_sol]
!       xjt,yjt,zjt -location of the jet origin [R_sol]
!       xjet,yjet,zjet -components of the unit aiming vector
!               of the primary jet cone
!       ivjt -the velocity field switch
!       ivjt not equal 2 -radial velocity is polynomial
!                v(r)=vjt*(r/rinjt)**eveljt
!       ivjt=2  radial velocity is: v(r)=vjt*(1-rcjt/r)**eveljt
!       vjt  -radial velocity at the inner edge or terminal velocity
!                depending on ivjt [km/s]
!       eveljt -velocity exponent
!       rcjt -radius of the object in the jet, only if ivjt=2, [R_sol]
!       vxjt, vyjt, vzjt -net velocity component [km/s]
!       tempjt -characteristic gas temperature [K]
!               jet gas temperature is a power law
!               gas:  T(r)=tempjt*(r/rinjt)**etmpjt
!       densjt -gas density [g/cm**3] at rinjt, it scales along the jet
!         to satisfy the continuity equation
!         density=densjt*rinjt**2/routjt**2*v(rinjt)/v(r)
!       anejt - electron number density [cm**-3] at rinjt. It changes
!         along the jet like the gas density but if ielnd=1 then
!         it is overriden by the calculation from the state quantities
!       dstdjt -dust density [g/cm**3] at rinjt, changes along the jet
!         like the gas density
!               you must also set densjt>0.0 to have an effect
!       dsttjt -characteristic dust temperature [K]
!               dust temperature is a power law
!               dust:  T(r)=dsttjt*(r/rinjt)**etmpjt
!               dust temperatures must be higher than the condensation
!               temperatures of the species (see dust_opac)
!               to have an effect
!       vtrbjt -microturbulence [km/s]
!-------ufo
!         it is identical to DISK (same subroutine) but lower priority
!       iufo=0 switch off the ufo
!       iufo=1 ufo has the shape of a rotating wedge
!               limited by inner and outer radii (spherical surfaces)
!               tip of the wedge is at the center of the object
!       iufo=2 ufo has the shape of a slab
!               limited by inner and outer radii (spherical surfaces)
!       iufo=3 ufo has the shape of a rotating ellipsoid
!               limited by inner spherical and outer ellipsoidal surface
!       aufo -angular halfwidth of the ufo wedge [deg]
!              (if iufo=1)
```

```
!             -half of the thickness of the ufo slab [R_sol]
!              (if iufo=2)
!             -semiaxis of the ellipsiod along the rotational axis
!              [R_sol] (if iufo=3)
!       rinuf -inner radius of the ufo [R_sol]
!       routuf -outer radius of the ufo [R_sol] or
!              -semiaxis of the ellipsoid perpendicular to the rotation
!               axis, if iufo=3, [R_sol]
!       emuf -mass of the object in the ufo center [M_sol]
!       ruf  -radius of the object in the ufo center [R_sol]
!       xuf,yuf,zuf -location of the disk center in [R_sol]
!       xufo,yufo,zufo -components of the unit aiming vector of
!               the rotational axis of the Keplerian disc around emuf
!       vxuf, vyuf, vzuf -net velocity components
!               of the center of the ufo [km/s]
!       tempuf  -temperature [K]
!       ituf=1  ufo gas and dust temperatures are constant
!       ituf=2  ufo gas and dust temperatures are a function of r
!               T(r)=tempuf*(Ruf/r)**0.75*(1-(Ruf/r)**0.5)**0.25
!               T(r)=dsttuf*(Ruf/r)**0.75*(1-(Ruf/r)**0.5)**0.25
!       ituf=3  ufo gas and dust temperatures as a power law
!               T(r)=tempuf*(r/rinuf)**etmpuf
!               T(r)=dsttuf*(r/rinuf)**etmpuf
!       etmpuf  -exponent of radial temperature dependence
!       densuf -gas density at rinuf [g/cm^3]
!       aneuf  -electron num. density at rinuf [cm^-3]
!       edenuf -radial density dependence exponent
!               (dens, ane and dust density are a function of r)
!               Rho(r) ~ Ne(r) ~ densuf*(r/rinuf)**edenuf
!       dstduf -dust density at rinuf [g/cm^3]
!               you must also set densuf>0. to have an effect
!       dsttuf -characteristic dust temperature [K]
!               dust temperatures must be higher than the condensation
!               temperatures of the species (see dust_opac)
!               to have an effect
!       vtrbuf -microturbulence [km/s]
!-------shell
!       ishell=0   switch off the shell
!       ishell=1   velocity, dens, temp, ane are constant
!       ishell=2   radial velocity is v(r)=vsh*(r/rinsh)**evelsh
!           Ne(r)~Rho(r)=denssh*(rinsh/r)**2*vsh/v(r)), temp=const.
!       ishell=3   radial velocity is v(r)=vsh*(1-rcsh/r)**evelsh
!           Ne(r)~Rho(r)=denssh*(rinsh/r)**2*v(rinsh)/v(r)),temp=const.
!       rinsh, routsh -inner, outer radius of the shell in [R_sol]
!       vsh -velocity of the uniformly expanding shell [km/s]
!       evelsh -exponent of velocity dependence
!       rcsh - core/photospheric radius of the star in shell [R_sol]
!       vxsh, vysh, vzsh -net velocity [km/s]
!       tempsh -temperature [K]
!       denssh -gas density at rinsh [g/cm^3]
!       anesh -electron number density at rinsh [cm^-3]
!       dstdsh -dust density at rinsh [g/cm^3],
!         it changes as the gas density
!               you must also set denssh>0. to have an effect
!       dsttsh -dust temperature [K]
!               dust temperatures must be higher than the condensation
!               temperatures of the species (see dust_opac)
!               to have an effect
!       vtrbsh -microturbulence [km/s]
!-------background
!       v0 -constant uniformly expanding velocity of background [km/s]
!       temp0 -temperature [K]
!       dens0 -gas density [g/cm^3](note dust density is =0 in the code)
!       ane0 -electron number density [cm^-3]
!-----------------------------------------------------------------------
```

```
!       If the objects happen to overlap, priority is given by the order
!       of 'if'-blocks in the subroutine smod1 and it is as follows:
!               star,companion,spot,stream,ring,disc,envelope,nebula,
!               flow,jet,ufo,shell,and background.
!       temp and ane are assumed to have reasonable values all along
!       the beam. An empty space can be defined as dens<denvac.
!       Four types of nontransparent objects can be defined as:
!       dcut1<dens<dcut2 -central star,
!       dcut2<dens<dcut3 -secondary star(=companion),
!       dcut3<dens<dcutn -3.body (it can be anything)
!       dcutn<dens -any opaque dark matter.
!         Note that lunt1, lunt2, lunt3 are in fact associated with
!       density intervals (<dcut1,dcut2>, <dcut2,dcut3>, <dcut3,dcutn>)
!       rather then with objects (star,companion,...) and thus can be
!       used to ascribe the spectrum to any nontransparent object
!       setting its density within a particular density interval.
!       However, limb darkening is applied to star and companion only
!       and it must be switched off (dlst=dlcp=0.) if you want to use
!       these density intervals for other objects (without limb dark.).
!       Roche geometry assumes synchronous rotation around z axis with
!       star in the center and companion at xcp>0,ycp=zcp=0 revolving
!       towards (0,1,0).
!         Treatment of the scattered light assumes that the medium is
!       optically thin, more precisely, that there is no significant
!       obstruction between the source of the light (the two stars)
!       and scattering medium. Only the radiative transfer along
!       the line of sight is solved. It means that objects could make
!       eclipses along the line of sight but would cast no shadows
!       into other directions (i.e. would be transparent when
!       considering scattered light from the two stars).
!       To allow objects to cast shadows we introduced a 3D field
!       where the user can specify whether a certain space point
!       is in the shadow so that only scattering from the unobscured
!       source (star) is taken into account.
!         Input variables which are supposed to be components of a unit
!       vector do not need to be normalized.
        dcut1=0.5d15
        dcut2=1.5d15
        dcut3=2.5d15
        dcutn=3.5d15
        denvac=1.d-50
```

5.2. line.dat

The file is read if $iline = 1$. It contains the atomic data for the spectral lines in the format identical to the SYNSPEC code. Each line of input corresponds to one spectral line with:

```
!       dll -wavelength [nm]
!       cod -element.ion cod, e.g. 26.02. It is interpreted as:
!         26=atomic number=iron, 02=2xtimes ionized i.e. FeIII line
!       gf -log_10 (gf)
!       elo,eup -energy of the lower and upper level in [1/cm]
!       qlo -quantum number -J of the lower level[=>stat.weight=2*J+1]
!       qup -quantum number -J of the upper level[=>stat.weight=2*J+1]
!       gr0,gs0,gw0-radiative, Stark, Van der Waals damping constants
```

5.3. shellspec.mod

The file is read if $imodel = 2$. It contains the model of the shell and is read with the following commands. Consult the example file if necessary.

```
!       reads the input model of the shell from the file
```

```
!       nbodf1,nbodf2,nbodf3 -number of x,y,z grid points
!       far,fat,faz -define the x,y,z grid points [cm]
!       ftemp, fdens - gas temperature [K] and density [g/cm^3]
!       fne -electron number density [cm^-3]
!       fdustd, fdustt -dust density [g/cm^3] and temperature [K]
!       fvr,fvt,fvz,fvtrb -x,y,z components of the velocity field [cm/s]
!       fvtrb -turbulence [cm/s]
!       kshade - shadows
!       kshade=1 scattering only from the central star
!       kshade=2 scattering only from the companion star
!       kshade=3 scattering from both stars (default)
!       kshade=0 no scattering
        read(10,*)nbodf1,nbodf2,nbodf3
        if(nbodf1.gt.ndimf1.or.nbodf2.gt.ndimf2.or.nbodf3.gt.ndimf3)then
          write(*,*)' error: space dimension exceeded, stop'
          write(3,*)' 1 error: space dimension exceeded, stop'
          goto 100
        endif
        read(10,*)(far(i),i=1,nbodf1)
        read(10,*)(fat(i),i=1,nbodf2)
        read(10,*)(faz(i),i=1,nbodf3)
        do i=1,nbodf1
        do j=1,nbodf2
        do k=1,nbodf3
          read(10,*)ftemp(i,j,k),fdustt(i,j,k),fdustd(i,j,k)            &
     &    ,fdens(i,j,k),fne(i,j,k)                                      &
     &    ,fvr(i,j,k),fvt(i,j,k),fvz(i,j,k),fvtrb(i,j,k),kshade(i,j,k)
        enddo
        enddo
        enddo
100     return
        end
```

5.4. abundances

The file is read if $ichemc = 1$ or $ielnd = 1$, otherwise built in solar abundances are assumed. The number of abundances (input lines) is read from the first line ($nichem$). $nichem$ lines follow with atomic number (ii), abundance ($abii$) of the elements whose abundance you wish to change, and an indicator ($necod$) whether the element is taken into account in electron number density calculation. The third column with $necod$ is mandatory only if $ielnd = 1$. Abundance is the element number density relative to hydrogen. Consult the example file if necessary.

5.5. phases

One column is read from this file if $nphase = 0$. The column contains the phases $< 0, 1 >$ which you want to calculate. These phases count from the x axis so that e.g. $phase = 0.$ or $1.$ is primary eclipse if $xcp > 0, ycp = zcp = 0, dinc = 90$.

5.6. lambda

One column is read from this file if $loglam = 2$. The column contains a list of wavelengths in Anstrom which you want to calculate.

5.7. starspec1

The file which defines the names of the individual files with intrinsic, not rotationally broadened spectra of the central star (1-st density interval, $< dcut1, dcut2 >$) and their effective temperatures. Kindly consult an example file. Program will interpolate in these precalculated spectra to the proper surface temperature of the object. If $lunt1 > 0$, two columns (xstar1, star1) are read from the individual files. If $lunt1 = 1$ then xstar1 is wavelength in Å and star1 is Eddington flux H_λ in [erg/cm^2/s/Å] i.e. the same as output of SYNSPEC. If $lunt1 = 2$ then xstar1 is wavelength in Å and star1 is central intensity I_ν in [erg/cm^2/s/Hz/sterad]. If $lunt1 = 3$ then second and forth column of the output of cool-TLUSTY (unit 21) are read with frequency [Hz], F_ν flux $[erg/cm^2/s/Hz]$. If your data are not in the units required you can use $xunt1, yunt1$ parameters to convert (multiply with) otherwise set $xunt1 = yunt1 = 1.0$.

5.8. starspec2

The same input as in starspec1 except that the relevant quantities are named $lunt2, xunt2, yunt2$ corresponding to the object named 'companion' (2-nd density interval, $< dcut2, dcut3 >$).

5.9. starspec3

The same input as in one of the individual files from starspec1 except that the relevant quantities are named $lunt3, xunt3, yunt3$, corresponding to the 3-rd density interval, $< dcut3, dcutn >$.

5.10. albedo1

The file is read if $ialbst = 1$ and $irrst = 1$. It contains the monochromatic albedo of the surface of the STAR as a function of wavelength. Albedo refers only to the scattered (reflected) light. Two columns: wavelength [Å] and albedo.

5.11. albedo2

Similar to albedo1 but for the COMPANION. The file is read if $ialbcp = 1$ and $irrcp = 1$. Two columns: wavelength [Å] and monochromatic albedo of the companion.

5.12. dust_opac

The file is read if $imie > 0$. It contains info about dust opacities. How many dust species or input files are provided, temperature range for each dust species (files), mass fraction of each species relative to the total mas fraction of all dust species, and file names with tables for each species. Each individual file (table) should contain four columns: dummy column, frequency [Hz], scattering opacity

per gram of dust material $[cm^2/g]$, absorption opacity per gram of dust material $[cm^2/g]$. We have precalculated tables in this format available for many species assuming Deirmendjian particle size distribution of spherical homogeneous grains. For example, take file `forsterite_opac_all`, which has many blocks for different modal particle sizes. Then extract a block for the particle size needed, save, and use as an input.

5.13. mie_phase

The file is read if $imiepf = 1$. It contains the dust phase functions. The file has several blocks, each block is for one frequency:
frequency [Hz]
two columns: angle [Deg.], phase function
frequency [Hz]
...
Phase function will be normalized in the code to 4π. We also have precalculated tables in this format available for many species. They have 65 phase angles. For example, take file `forsterite_phase_all`, which has many super-blocks for different modal particle sizes. Then extract a super-block for the particle size needed, save, and use as an input. If you have different number of phase angles change npfang parameter in param.inc and recompile.

5.14. gas_opac

The file is read if $iopac = 1$. It contains true absortion cross-section of the gas as a function of frequency and temperature similar to the EXOMOL data format. Fist line: number of frequences, temperatures, and molecule mixing ratio (relative number density of a particular molecule with respect to total hydrogen nuclei number density). Second line: temperatures. Several columns follow. First column is wavenumber $1/\lambda$ $[cm^{-1}]$. Other columns are cross-section $[cm^2]$ for each temperature. After some modification it could be used to feed the code directly with opacities.

5.15. chem_eq_tab

The file is read if $iopac = 1$. It contains table of log10 of molecular populations (number densities) as a function of log10 gas temperature and log10 gas density all in cgs units. They may be e.g. a result from the chemical equilibrium calculations. Consult the example file for the format. If not available user can uncomment one line in the code and use a constant molecule mixing ration from the previous file instead.

6. Output

Here is a list of all output files with their unit numbers.

shellspec.out - (2) more detailed output
2Dimage_xxx - (100+iang) 2D images at some frequency
shellspectrum - (4) spectrum of the shell
lightcurve - (11) light curve or trailed spectrogram
errors - (3) error messages

6.1. shellspectrum

The file contains several blocks separated by a blank line. Each block corresponds to one rotation (the view point or phase) of the shell. The block has 6 columns: (1) lambda [Å], (2) velocity corresponding to lambda ($c\Delta\lambda/\lambda_{lu}$) [$km\,s^{-1}$] ($\lambda_{lu}$ of the first line from line.dat or alam1), (3) F_ν, absolute flux at the Earth [$erg/cm^2/s/Hz$], (4) F_λ, absolute flux at the Earth [$erg/cm^2/s/cm$], (5) normalized F_ν flux, (6) normalized F_ν flux shifted in y-axis for each subsequent rotation (phase) by the value 'offset' for easy plotting.

6.2. lightcurve

This file contains several blocks separated by a blank line. Each block corresponds to one rotation (phase) of the shell. The block has 5 columns: (1) internal phase. If $nphase = 0$ then $internal\ phase + 0.25$ will result in a value from $< 0, 1 >$ where 0 corresponds to the primary eclipse provided that $xcp > 0, ycp = zcp = 0, dinc = 90$; (2) radial velocity [$km\,s^{-1}$]; (3) magnitude $= -2.5\log_{10} F_\lambda$; (4) lambda [Å]; and (5) F_ν, absolute flux at the Earth [$erg/cm^2/s/Hz$].

6.3. shellspec.out

This file contains more detailed output of various quantities (opacities, emissivities, optical depth, ...), mainly details along one particular ray (line of sight) and frequency specified in the input by 'ionu, ior, iot'.

6.4. 2Dimage_xxx

Here, xxx is an integer corresponding to a particular view point (rotation) of the model. These are 2D-xy projection images of the shell at different phases at the frequency specified by *ionu*. Each file corresponds to one phase. Each file consists of several blocks separated by a blank line. Each block corresponds to one x-value (y-varies) and has 3 columns: (1) x [cm], (2) y [cm], (3) I_ν in [$erg/cm^2/s/Hz/sterad$].

6.5. errors

This file contains error messages which are also written on the screen. It will read ‘ 0 errors found’ in case no errors were detected. This does not guarantee that the result and code are error/bug free.

7. Demonstration of an Artificial Model

As an illustration, we include a few pictures calculated for an artificial spectral line and an artificial test model. It includes two stars, a Keplerian equatorial disc around a bigger primary, a slightly inclined jet and a slowly expanding shell surrounding the system. Stars are treated as blackbodies, the primary is a sphere with limb darkening imposed on while the secondary fills its Roche lobe and is subject to gravity darkening only. The centers of the jets and shell have no net space velocity while the net velocity of the center of the disc corresponds to that of the primary. Jets presess with the orbital period. Calculations were performed for about 50 phases as seen from the orbital plane. The model was defined in a cube with 101x101x101 points and spectrum was calculated at 241 frequency points. For the sole purpose of this illustration, the input values were manipulated so that a contribution from each object could be seen. Figure 3 shows a 2D projection image ($I_\nu^{out}(x,y)$) of a test model in the continuum (at a frequency in the far wing of the spectral line) taken roughly at quadrature. Different shapes of the primary and secondary illustrate the effects of limb darkening and gravity darkening. Figure 4 shows the overall light curve of the model as it revolves with apparent primary and secondary minima at each frequency. Figure 5 displays the trailed spectrogram. The central emission comes from the slowly expanding shell. The double wave is caused by the two jet cones. The two single waves (blue and red) originate in the disc and reflect its orbital motion tracing that of the primary. Observe a depression in both single waves near the primary minimum caused by the eclipse of the approaching and receding part of the disc by the secondary star.

Acknowledgements. I would like to thank Drs. J. Nemravova, A. Tkachenko, Prof. I. Hubeny, and mainly Dr. M.Broz for their comments and discussions. JB gratefully acknowledges grant support from Penn State University, NSF-NATO fellowship (NSF DGE-0312144) and Marie Curie International Reintegration Grant FP7-200297. This study made use of the Vienna Atomic Line Data Base (VALD) services.

References

Bisikalo, D.V., Boyarchuk, A.A., Chechetkin, V.M., Kuznetsov, O.A., Molteni, D.: 1998, *Mon. Not. R. Astron. Soc.* **300**, 39

Bradstreet, D.H., Steelman, D.P.: 2002, *Bull. Am. Astron. Soc.* **34**, 1224

Budaj, J., 2011, AJ, 141, 59

Budaj, J., Dworetsky, M.M.: 2002, *Mon. Not. R. Astron. Soc.* **337**, 1340

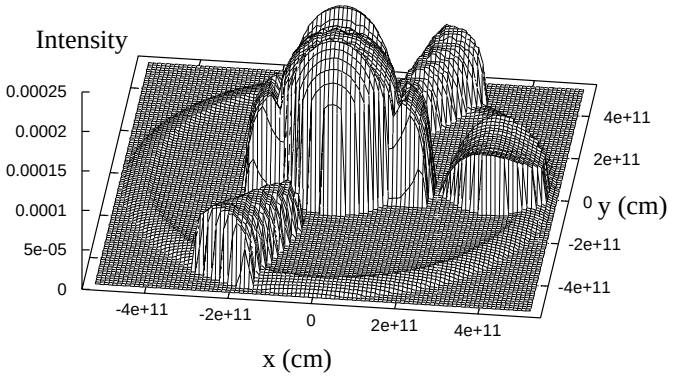

Figure 3. 2D projection image, $I_\nu^{out}(x, y)$, of a test model in one particular phase.

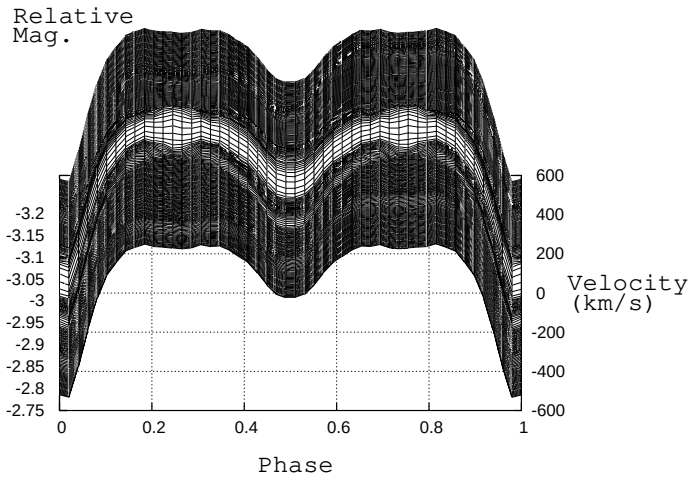

Figure 4. Overall light curve of the test model.

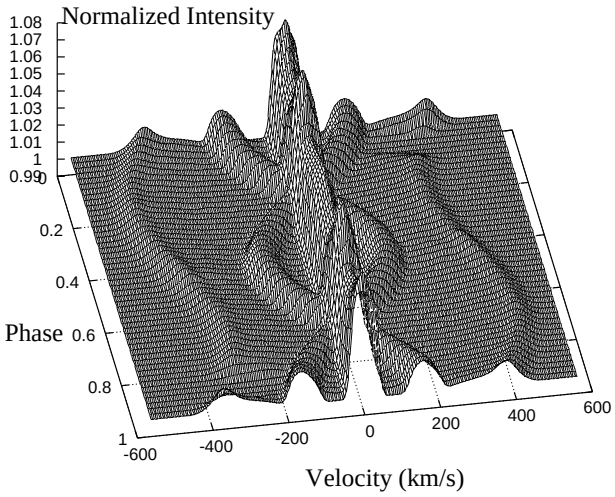

Figure 5. Trailed spectrogram of the test model.

Budaj, J., Dworetsky, M.M., Smalley, B.: 2002, *Comm. Univ. London Obs.* **82** URL=http://www.ulo.ucl.ac.uk/ulo_comms/82/index.html
Budaj, J., Hubeny, I., Burrows, A. 2012, A&A, 537, 115
Budaj, J., Kocifaj, M., Salmeron, R, Hubeny, I. 2015, MNRAS, 454, 2
Burrows, A., Sudarsky, D., Hubeny I.: 2006, *ApJ* **650**, 1140
Cardelli, J.A., Clayton, G.C., Mathis, J.S.: 1989, *ApJ* **345**, 245
Djurasevic, G.: 1992, *Astrophys. Space Sci.* **197**, 17
Drechsel, H., Haas, S., Lorenz, R., Mayer, P.: 1994, *Astron. Astrophys.* **284**, 853
Grevesse, N., Sauval, A.J.: 1998, *Space Sci. Rev.* **85**, 161
Gray, D.F.: 1976, *Observation and analysis of stellar photospheres*, A Wiley-Interscience publ., New York
Hadrava, P.: 1997, *Astron. Astrophys., Suppl. Ser.* **122**, 581
Hill, G.: 1979, *Publ. Dom. Ap. Obs. Victoria* **15**, 297
Hilditch, R.W.: 2001, *An introduction to close binary stars*, Cambridge Univ. Press, Cambridge
Horne, K., Marsh, T.R.: 1986, *Mon. Not. R. Astron. Soc.* **218**, 761
Horvat, M., Conroy, K.E., Jones, D., Pra, A.: 2019, *ApJS* **240**, 36
Hubeny, I., Lanz, T.: 1995, *Astrophys. J.* **439**, 875
Hubeny, I., Lanz, T., Jeffery, C.S.: 1994, in Newsletter on Analysis of Astronomical spectra No.20, ed. C.S. Jeffery (CCP7; St. Andrews: St. Andrews Univ.), 30
Ibgui, L., Hubeny, I., Lanz, T., Stehl, C.: 2013, *A&A* **549**, A126
Jackson, B.K., Lewis, N.K., Barnes, J.W., Drake D.L., Showman, A.P., Fortney, J.J.: 2012, *Astrophys. J.* **751**, 112

Kallrath, J., Milone, E.F., Terrell, D., Young, A.T.: 1998, *Astrophys. J.* **508**, 308
Kopal, Z.: 1959, *Close Binary Systems*, Chapman & Hall Ltd., London
Korčáková, D., Kubát, J.: 2005, *A&A* **440**, 715
Kupka, F., Piskunov, N.E., Ryabchikova, T.A., Stempels, H.C., Weiss, W.W.: 1999, *Astron. Astrophys., Suppl. Ser.* **138**, 119
Kurucz, R.L.: 1970, SAO Special Report 309
Kurucz, R.L.: 1993a, SYNTHE spectrum synthesis programs and line data (Kurucz CD-ROM 18)
Linnell, A.P., Hubeny, I.: 1996, *Astrophys. J.* **471**, 958
Lucy, L.B.: 1968, *Astrophys. J.* **153**, 877
Mihalas, D.: 1978, *Stellar Atmospheres*, W.H. Freeman and company, San Francisco
Mochnacki, S.W., Doughty, N.A.: 1972, *Mon. Not. R. Astron. Soc.* **156**, 51
Mourard, D., Brož, M., Nemravová, J. A., Harmanec, P., Budaj, J., Baron, F., Monnier, J. D., Schaefer, G. H. et al.: 2018, *A&A* **618**, A112
Nazarenko, V.V., Glazunova, L.V.: 2006, *Astronomy Reports* **50**, 369
Orosz, J. A., Hauschildt, P. H.: 2000, *A&A* **364**, 265
Pavlovski, K., Burki, G., Mimica, P. 2006, A&A, 454, 855
Plavec, M., Kratochvil, P.: 1964, *Bull. Astron. Inst. Czechosl.* **15**, 165
Pontoppidan, K.M., Meijerink, R., Dullemond, C.P., Blake, G.A.: 2009, *ApJ* **704**, 1482
Popper D.M., Etzel P.B.: 1981, *AJ* **86**, 102
Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: 1992, *Numerical Recipes*, Cambridge Univ. Press, Cambridge
Pribulla, T.: 2004, in *Spectroscopically and Spatially Resolving the Components of Close Binary Stars*, eds.: R. W. Hidlitch, H. Hensberge and K. Pavlovski, ASP Conf. series., 318, 117
Pringle, J.E.: 1981, *Ann. Rev. Astron. Astrophys.* **19**, 137
Prša, A., Zwitter, T.: 2005, *Astrophys. J.* **628**, 426
Prša, A., Harmanec, P., Torres, G. et al.: 2016, *Astron. J.* **152**, 41
Richards, M.T., Ratliff, M.A.: 1998, *Astrophys. J.* **493**, 326
Rucinski, S.M.: 1973, *Acta Astron.* **23**, 79
Smith, K.C.: 1992, PhD thesis, University of London
Smith, K.C., Dworetsky, M.M.: 1988, in *Elemental Abundance Analyses*, eds.: S.J. Adelman and T. Lanz, Institut d'Astronomie de l'Univ. de Lausanne, Switzerland, 32
Southworth, J., Maxted, P. F. L., Smalley, B.: 2004, *Mon. Not. R. Astron. Soc.* **349**, 547
Sejnova, K., Votruba, V., Koubsky, P.: 2012, in *Circumstellar Dynamics at High Resolution*, eds.: A. Carciofi and Th. Rivinius, ASP Conference Proceedings, 464, 219
Tkachenko, A., Lehmann, H., Mkrtichian, D.: 2010, *AJ* **139**, 1327
Vinkó, J., Hegedüs, T., Hendry, D.: 1996, *Mon. Not. R. Astron. Soc.* **280**, 489
Tamuz O., Mazeh T., North P.: 2006, *MNRAS* **367**, 1521
Wilson, R.E.: 1990, *Astrophys. J.* **356**, 613
Wilson, R.E., Devinney, E.J.: 1971, *Astrophys. J.* **166**, 605
Wood, D.B.: 1971, *AJ* **76**, 701
Zhang, E.-H., Robinson, E.L., Nather, R.E.: 1986, *Astrophys. J.* **305**, 740