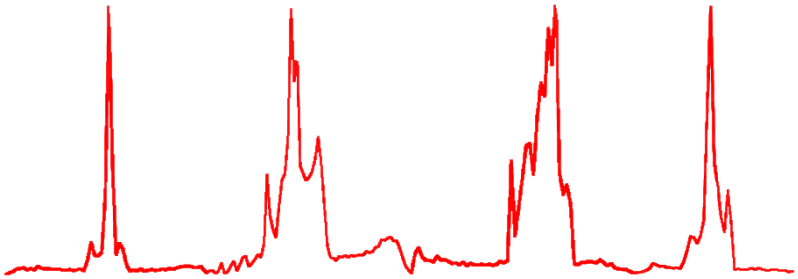
LIFELINE: the code for the simulation of the X-ray LIne proFiles in massivE coLliding wInd biNariEs



Issue: 1
Date: September 2020

Enmanuelle Mossoux
University of Liège
STAR Institute
Allée du 6 août, 19C
4000 Liège
Belgium

This manual describes LIFELINE, the ULiège program for the simulation of the X-ray **LI**ne pro**F**iles in massiv**E** co**L**liding w**I**nd bi**N**ari**E**s which is designed to compute the profile of emission lines emitted in X-rays in the colliding wind region of massive stars binaries. It was developed as a programmable, interactive tool for studying the physics of the winds in binary systems, the emission of X-ray lines and allowing the comparison with the observations. LIFELINE is written entirely in `Python` and is intended to be portable across most Unix operating systems. LIFELINE requires `Python` v2.7, its pyatomdb package and, optionally, the ATOMDB database. This manual is designed to serve as an introduction, user's guide, and definitive reference manual to the LIFELINE program.

Table of content

Chapter 1

Simulation of the stellar winds

We define a spherical coordinate system centered on the star with the less powerful wind. The velocity and direction of a small-volume material of the wind of a star is controlled by the equilibrium of the forces acting on it. In a binary system (see Fig. 1.1), the material is first accelerated by the line-radiation pressure and braked by the gravitation of the emitting star. Then, approaching the companion star, the material encounters the gravitation force created by the companion's mass (Stevens 1988), the dynamical impact of the companion's radiation field (Stevens et al. 1992) and the suppression of the line-driving which may be caused by the ionization of wind material (Stevens 1991). Stevens & Pollock (1994) computed the effects of the radiation and gravitation fields of both stars on the velocity of the wind along the line of centers between the stars. We extend their work to compute the velocity of the wind all around the emitting star. The spherical coordinates of the velocity are $(u_\mathrm{r}, u_\theta, u_\phi)$.

1.1 Navier-Stokes equations

We first express the equation of Navier-Stokes in spherical coordinates:

$$\rho\left(\frac{\partial\vec{v}}{\partial t}+(\vec{v}\cdot\nabla)\vec{v}\right)=-\nabla P+\sigma\left((\nabla^2\vec{v}+\nabla(\nabla\cdot\vec{v}))-\frac{2}{3}\nabla(\nabla\cdot\vec{v})\right)+\rho\vec{F} \tag{1.1}$$

This equation is solved at a constant time so that $\partial\vec{v}/\partial t = 0$. We develop all the terms one by one.

1.1.1 $(\vec{v}\cdot\nabla)\vec{v}$

$$\nabla(\vec{A}\cdot\vec{B})=\vec{A}\times(\nabla\times\vec{B})+\vec{B}\times(\nabla\times\vec{A})+(\vec{B}\cdot\nabla)\vec{A}+(\vec{A}\cdot\nabla)\vec{B} \tag{1.2}$$

$$\vec{v}\times(\nabla\times\vec{v})=(u_\mathrm{r},u_\theta,u_\phi)\times\left(\frac{1}{r\,\sin\theta}\left(\frac{\partial(\sin\theta\,u_\phi)}{\partial\theta}-\frac{\partial u_\theta}{\partial\phi}\right),\frac{1}{r}\left(\frac{1}{\sin\theta}\frac{\partial u_\mathrm{r}}{\partial\phi}-\frac{\partial(ru_\phi)}{\partial r}\right),\frac{1}{r}\left(\frac{\partial(ru_\theta)}{\partial r}-\frac{\partial u_\mathrm{r}}{\partial\theta}\right)\right) \tag{1.3}$$

$$\Rightarrow\vec{v}\times(\nabla\times\vec{v})=\begin{pmatrix}\frac{u_\theta}{r}\left(u_\theta+r\frac{\partial u_\theta}{\partial r}-\frac{\partial u_\mathrm{r}}{\partial\theta}\right)-\frac{u_\phi}{r}\left(\frac{1}{\sin\theta}\frac{\partial u_\mathrm{r}}{\partial\phi}-u_\phi-r\frac{\partial u_\phi}{\partial r}\right)\\ \frac{u_\phi}{r\sin\theta}\left(\cos\theta u_\phi+\sin\theta\frac{\partial u_\phi}{\partial\theta}-\frac{\partial u_\theta}{\partial\phi}\right)-\frac{u_\mathrm{r}}{r}\left(r\frac{\partial u_\theta}{\partial r}+u_\theta-\frac{\partial u_\mathrm{r}}{\partial\theta}\right)\\ \frac{u_\mathrm{r}}{r}\left(\frac{1}{\sin\theta}\frac{\partial u_\mathrm{r}}{\partial\phi}-u_\phi-r\frac{\partial u_\phi}{\partial r}\right)-\frac{u_\theta}{r\sin\theta}\left(\cos\theta u_\phi+\sin\theta\frac{\partial u_\phi}{\partial\theta}-\frac{\partial u_\theta}{\partial\phi}\right)\end{pmatrix} \tag{1.4}$$

$$\nabla(\vec{v}\cdot\vec{v})=\left(\frac{\partial}{\partial r},\frac{1}{r}\frac{\partial}{\partial\theta},\frac{1}{r\sin\theta}\frac{\partial}{\partial\phi}\right)\cdot(u_\mathrm{r}^2+u_\theta^2+u_\phi^2) \tag{1.5}$$

$$\Rightarrow\nabla(\vec{v}\cdot\vec{v})=2\begin{pmatrix}u_\mathrm{r}\frac{\partial u_\mathrm{r}}{\partial r}+u_\theta\frac{\partial u_\theta}{\partial r}+u_\phi\frac{\partial u_\phi}{\partial r}\\ \frac{u_\mathrm{r}}{r}\frac{\partial u_\mathrm{r}}{\partial\theta}+\frac{u_\theta}{r}\frac{\partial u_\theta}{\partial\theta}+\frac{u_\phi}{r}\frac{\partial u_\phi}{\partial\theta}\\ \frac{u_\mathrm{r}}{r\sin\theta}\frac{\partial u_\mathrm{r}}{\partial\phi}+\frac{u_\theta}{r\sin\theta}\frac{\partial u_\theta}{\partial\phi}+\frac{u_\phi}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi}\end{pmatrix} \tag{1.6}$$

By injecting Eq. 1.4 and 1.6 in 1.2, we find:

$$(\vec{v}\cdot\nabla)\vec{v}=\begin{pmatrix}u_\mathrm{r}\frac{\partial u_\mathrm{r}}{\partial r}+\frac{u_\theta}{r}\frac{\partial u_\mathrm{r}}{\partial\theta}+\frac{u_\phi}{r\sin\theta}\frac{\partial u_\mathrm{r}}{\partial\phi}-\frac{u_\phi^2+u_\theta^2}{r}\\ u_\mathrm{r}\frac{\partial u_\theta}{\partial r}+\frac{u_\theta}{r}\frac{\partial u_\theta}{\partial\theta}+\frac{u_\phi}{r\sin\theta}\frac{\partial u_\theta}{\partial\phi}-\frac{u_\phi^2}{r\tan\theta}+\frac{u_\mathrm{r}\,u_\theta}{r}\\ u_\mathrm{r}\frac{\partial u_\phi}{\partial r}+\frac{u_\theta}{r}\frac{\partial u_\phi}{\partial\theta}+\frac{u_\phi}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi}+\frac{u_\phi\,u_\theta}{r\tan\theta}+\frac{u_\mathrm{r}\,u_\phi}{r}\end{pmatrix} \tag{1.7}$$

1.1.2 $\nabla^2\vec{v}$

$$\nabla^2\vec{v} = \frac{1}{r^2}\begin{pmatrix} \frac{\partial}{\partial r}\left(r^2\frac{\partial u_\mathrm{r}}{\partial r}\right) + \frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial u_\mathrm{r}}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2 u_\mathrm{r}}{\partial\phi^2} - 2u_\mathrm{r} - \frac{2}{\sin\theta}\left(\frac{\partial}{\partial\theta}(\sin\theta u_\theta) + \frac{\partial u_\phi}{\partial\phi}\right) \\ \frac{\partial}{\partial r}\left(r^2\frac{\partial u_\theta}{\partial r}\right) + \frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial u_\theta}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2 u_\theta}{\partial\phi^2} - \frac{u_\theta}{\sin^2\theta} + 2\frac{\partial u_\mathrm{r}}{\partial\theta} - \frac{2\cos\theta}{\sin^2\theta}\frac{\partial u_\phi}{\partial\phi} \\ \frac{\partial}{\partial r}\left(r^2\frac{\partial u_\phi}{\partial r}\right) + \frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial u_\phi}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2 u_\phi}{\partial\phi^2} - \frac{u_\phi}{\sin^2\theta} + \frac{2}{\sin^2\theta}\frac{\partial u_\mathrm{r}}{\partial\phi} + \frac{2\cos\theta}{\sin^2\theta}\frac{\partial u_\theta}{\partial\phi} \end{pmatrix} \tag{1.8}$$

1.1.3 $\nabla(\nabla\cdot\vec{v})$

$$\nabla\cdot\vec{v} = \frac{1}{r^2}\frac{\partial(r^2 u_\mathrm{r})}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial(\sin\theta u_\theta)}{\partial\theta} + \frac{1}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi} \tag{1.9}$$

$$\nabla(\nabla\cdot\vec{v}) = \begin{pmatrix} \frac{\partial}{\partial r}\left(\frac{1}{r^2}\frac{\partial(r^2 u_\mathrm{r})}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial(\sin\theta u_\theta)}{\partial\theta} + \frac{1}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi}\right) \\ \frac{1}{r}\frac{\partial}{\partial\theta}\left(\frac{1}{r^2}\frac{\partial(r^2 u_\mathrm{r})}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial(\sin\theta u_\theta)}{\partial\theta} + \frac{1}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi}\right) \\ \frac{1}{r\sin\theta}\frac{\partial}{\partial\phi}\left(\frac{1}{r^2}\frac{\partial(r^2 u_\mathrm{r})}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial(\sin\theta u_\theta)}{\partial\theta} + \frac{1}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi}\right) \end{pmatrix} \tag{1.10}$$

By injecting Eq. 1.7, 1.8 and 1.10 in 1.1, we finally have:

$$\begin{pmatrix} u_\mathrm{r}\frac{\partial u_\mathrm{r}}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_\mathrm{r}}{\partial\theta} + \frac{u_\phi}{r\sin\theta}\frac{\partial u_\mathrm{r}}{\partial\phi} - \frac{u_\phi^2+u_\theta^2}{r} \\ u_\mathrm{r}\frac{\partial u_\theta}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_\theta}{\partial\theta} + \frac{u_\phi}{r\sin\theta}\frac{\partial u_\theta}{\partial\phi} + \frac{u_\phi^2}{r\tan\theta} + \frac{u_\mathrm{r}u_\theta}{r} \\ u_\mathrm{r}\frac{\partial u_\phi}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_\phi}{\partial\theta} + \frac{u_\phi}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi} + \frac{u_\phi u_\theta}{r\tan\theta} - \frac{u_\mathrm{r}u_\phi}{r} \end{pmatrix} = -\frac{1}{\rho}\begin{pmatrix} \frac{\partial P}{\partial r} \\ \frac{1}{r}\frac{\partial P}{\partial\theta} \\ \frac{1}{r\sin\theta}\frac{\partial P}{\partial\phi} \end{pmatrix} + \begin{pmatrix} F_\mathrm{r} \\ F_\theta \\ F_\phi \end{pmatrix}$$
$$+\frac{\sigma}{\rho r^2}\begin{pmatrix} \frac{\partial}{\partial r}\left(r^2\frac{\partial u_\mathrm{r}}{\partial r}\right) + \frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial u_\mathrm{r}}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2 u_\mathrm{r}}{\partial\phi^2} - 2u_\mathrm{r} - \frac{2}{\sin\theta}\left(\frac{\partial}{\partial\theta}(\sin\theta u_\theta) + \frac{\partial u_\phi}{\partial\phi}\right) \\ \frac{\partial}{\partial r}\left(r^2\frac{\partial u_\theta}{\partial r}\right) + \frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial u_\theta}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2 u_\theta}{\partial\phi^2} - \frac{u_\theta}{\sin^2\theta} + 2\frac{\partial u_\mathrm{r}}{\partial\theta} - \frac{2\cos\theta}{\sin^2\theta}\frac{\partial u_\phi}{\partial\phi} \\ \frac{\partial}{\partial r}\left(r^2\frac{\partial u_\phi}{\partial r}\right) + \frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial u_\phi}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2 u_\phi}{\partial\phi^2} - \frac{u_\phi}{\sin^2\theta} + \frac{2}{\sin\theta}\frac{\partial u_\mathrm{r}}{\partial\phi} + \frac{2\cos\theta}{\sin^2\theta}\frac{\partial u_\theta}{\partial\phi} \end{pmatrix}$$
$$+\frac{\sigma}{3\rho}\begin{pmatrix} \frac{\partial}{\partial r}\left(\frac{1}{r^2}\frac{r^2\partial u_\mathrm{r}}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial\sin\theta u_\theta}{\partial\theta} + \frac{1}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi}\right) \\ \frac{1}{r}\frac{\partial}{\partial\theta}\left(\frac{1}{r^2}\frac{r^2\partial u_\mathrm{r}}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial\sin\theta u_\theta}{\partial\theta} + \frac{1}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi}\right) \\ \frac{1}{r\sin\theta}\frac{\partial}{\partial\phi}\left(\frac{1}{r^2}\frac{r^2\partial u_\mathrm{r}}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial\sin\theta u_\theta}{\partial\theta} + \frac{1}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi}\right) \end{pmatrix} \tag{1.11}$$

1.2 Distribution of the wind velocity

We suppose that the wind is symmetric by rotation around the line of centers. We thus do not work with the ϕ component of Eq. 1.11 (last row of the matrices) and the $\partial/\partial\phi$ and u_ϕ terms are equal to zero. The norm of the wind velocity is thus $v = \sqrt{u_\mathrm{r}^2 + u_\theta^2}$. We also assume a viscosity $\sigma = 0$. The Navier-Stokes equation is thus resumed to:

$$\begin{pmatrix} u_\mathrm{r}\frac{\partial u_\mathrm{r}}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_\mathrm{r}}{\partial\theta} - \frac{u_\theta^2}{r} \\ u_\mathrm{r}\frac{\partial u_\theta}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_\theta}{\partial\theta} + \frac{u_\mathrm{r}u_\theta}{r} \end{pmatrix} = -\frac{1}{\rho}\begin{pmatrix} \frac{\partial P}{\partial r} \\ \frac{1}{r}\frac{\partial P}{\partial\theta} \end{pmatrix} + \begin{pmatrix} F_\mathrm{r} \\ F_\theta \end{pmatrix} \tag{1.12}$$

The pressure of an ideal gas is $P = a^2\rho/\gamma$ with γ the adiabatic index and $a = \sqrt{\gamma\, k_\mathrm{B}T/(\mu m_\mathrm{H})}$ the isothermal sound velocity with k_B the Boltzmann constant, T the temperature of the winds that we suppose to be equal to effective temperature (T_eff) and μ the mean molecular weight. We thus have:

$$\begin{aligned} \frac{\partial P}{\partial r} &= \frac{2a\rho}{\gamma}\frac{\partial a}{\partial r} + \frac{a^2}{\gamma}\frac{\partial\rho}{\partial r} \\ \frac{\partial P}{\partial\theta} &= \frac{a^2}{\gamma}\frac{\partial\rho}{\partial\theta} \end{aligned} \tag{1.13}$$

Considering an isothermal gas, $\partial a/\partial r = 0$.

The mass-loss rate is:

$$\dot{M} = 4\pi\,\rho\, r^2 v = 4\pi\,\rho\, r^2\sqrt{u_\mathrm{r}^2 + u_\theta^2} \tag{1.14}$$

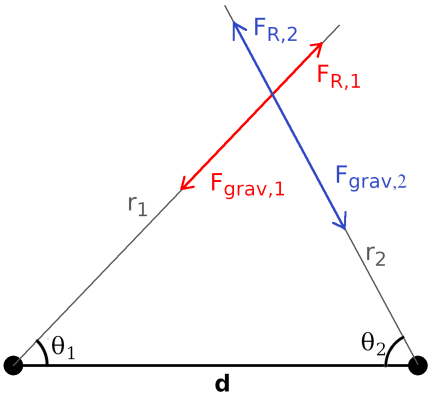


Figure 1.1: Representation of the forces acting upon a small element of material in the wind of star 1 on the left hand side of the figure.

It is considered as constant and uniform [1]. Its derivatives are thus:

$$\begin{aligned}\frac{\partial \dot{M}}{\partial r} &= 0 = 4\pi\left(\rho\, v\, 2r + \frac{\rho\, r^2}{2v}\frac{\partial(u_\mathrm{r}^2+u_\theta^2)}{\partial r} + r^2 v\frac{\partial \rho}{\partial r}\right)\\ \frac{\partial \dot{M}}{\partial \theta} &= 0 = 4\pi\left(\frac{\rho\, r^2}{2v}\frac{\partial(u_\mathrm{r}^2+u_\theta^2)}{\partial \theta} + r^2 v\frac{\partial \rho}{\partial \theta}\right)\end{aligned} \tag{1.15}$$

We thus have:

$$\begin{aligned}-\frac{\boldsymbol{a^2}}{\boldsymbol{\gamma\rho}}\frac{\partial \rho}{\partial r} &= \frac{\boldsymbol{a^2}}{\boldsymbol{\gamma\rho}}\frac{2\rho}{r} + \frac{\boldsymbol{a^2}}{\boldsymbol{\gamma\rho}}\frac{\rho}{v^2}\left(u_\mathrm{r}\frac{\partial u_\mathrm{r}}{\partial r} + u_\theta\frac{\partial u_\theta}{\partial r}\right)\\ -\frac{\boldsymbol{a^2}}{\boldsymbol{\gamma\rho r}}\frac{\partial \rho}{\partial \theta} &= \frac{\boldsymbol{a^2}}{\boldsymbol{\gamma\rho r}}\frac{\rho}{v^2}\left(u_\mathrm{r}\frac{\partial u_\mathrm{r}}{\partial \theta} + u_\theta\frac{\partial u_\theta}{\partial \theta}\right)\end{aligned} \tag{1.16}$$

Equation 1.16 can be injected in Eq. 1.13. The components of Eq. 1.12 are thus:

$$\left(1-\frac{a^2}{\gamma v^2}\right)u_\mathrm{r}\frac{\partial u_\mathrm{r}}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_\mathrm{r}}{\partial \theta} - \frac{a^2}{\gamma v^2}u_\theta\frac{\partial u_\theta}{\partial r} - \frac{u_\theta^2}{r} = \frac{2a^2}{\gamma r} + F_\mathrm{r} \tag{1.17}$$

$$\left(1-\frac{a^2}{\gamma v^2}\right)u_\theta\frac{\partial u_\theta}{\partial \theta} + r u_\mathrm{r}\frac{\partial u_\theta}{\partial r} - \frac{a^2}{\gamma v^2}u_\mathrm{r}\frac{\partial u_\mathrm{r}}{\partial \theta} + u_\mathrm{r}u_\theta = rF_\theta \tag{1.18}$$

The last terms are the forces components (F_r, F_θ) which are the sum of the gravity (F_grav) and the radiative acceleration (F_R) from both stars (see Fig. 1.1).

$$\begin{aligned}F_\mathrm{r} &= F_{\mathrm{R},1} - F_{\mathrm{grav},1} + F_{\mathrm{grav},2}\cos(\theta_1+\theta_2) - F_{\mathrm{R},2}\cos(\theta_1+\theta_2)\\ F_\theta &= F_{\mathrm{grav},2}\sin(\theta_1+\theta_2) - F_{\mathrm{R},2}\sin(\theta_1+\theta_2)\end{aligned} \tag{1.19}$$

The gravitational acceleration at a distance r_i of the star i is $F_{\mathrm{grav,i}} = G\,M_\mathrm{i}(1-\Gamma_\mathrm{i})/r_\mathrm{i}^2$ with G the gravitational constant, M_i the stellar mass and Γ_i the Eddington ratio computed as $\sigma_\mathrm{t}\,L_\mathrm{i}/(M_\mathrm{i}\,4\pi\,G\,c\,m_\mathrm{p})$ with σ_t the Thomson cross-section, L_i the luminosity of the star, c the speed of light and m_p the proton mass.

The CAK theory (Castor et al. 1975) allows the parametrization of the radiative pressure with two parameters (α and k):

$$g_\mathrm{rad} = \frac{\sigma_\mathrm{e}^{1-\alpha}k}{c\rho^\alpha V_\mathrm{th}^\alpha}\left|\frac{dv}{dr}\right|^\alpha \tag{1.20}$$

with $\sigma_\mathrm{e} = (1+H)\sigma_\mathrm{t}/(2m_\mathrm{H})$ the scattering electron opacity with H the hydrogen mass fraction and $V_\mathrm{th} = \sqrt{2k_\mathrm{B}T_\mathrm{eff}/m_\mathrm{H}}$ the ion thermal speed. The radiative acceleration is thus $F_{\mathrm{R,i}} = g_\mathrm{rad}F_\mathrm{i}K_\mathrm{i}$ with $F_\mathrm{i} = L_\mathrm{i}/(4\pi r_\mathrm{i}^2)$ and $K_\mathrm{i} = 1-(1-(R_\mathrm{i}/r_\mathrm{i})^2)^{1+\alpha}/((1+\alpha)(R_\mathrm{i}/r_\mathrm{i})^2)$ the finite disc correction factor. The constants α and k were computed by Abbott (1982) for a set of effective temperatures and wind densities.

[1] Future investigations should be done to consider the mass-loss rate varying with r and θ.

1.3 Solving the equations

1.3.1 Along the line of centers

Between the two stars, $\theta = 0$. We thus only work with the r component of Eq. 1.11 (first row of the matrices). This is the configuration considered by Stevens & Pollock (1994). We report here the resolution for the star 1. A similar solution can be found for the star 2. For star 1, we thus have to solve:

$$\left(1 - \frac{a^2}{\gamma\, v^2}\right) v \frac{dv}{dr} - 2\frac{a^2}{\gamma\, r_1} - \frac{G\, M_2\, (1-\Gamma_2)}{(d-r_1)^2} + \frac{G\, M_1\, (1-\Gamma_1)}{r_1^2} - \frac{\sigma_{\rm e}^{1-\alpha}\, k\, (F_1\, K_1 - F_2\, K_2)}{c\, V_{\rm th}^{\alpha}\, \rho^{\alpha}} \left|\frac{dv}{dr}\right|^{\alpha} = 0 \quad (1.21)$$

from the critical point $r_{\rm c} = 1.05\, R_1$ where the velocity is:

$$v_{\rm c} = a \sqrt{\frac{1 + \alpha\, h_{\rm c}}{(1-\alpha)\, \sqrt{h_{\rm c}\, (dB/du)/((1-\alpha)\, B_{\rm c}) - dh/du}}} \quad (1.22)$$

with $h_{\rm c} = 1 + 4\, M_2\, (1-\Gamma_2)\, (u_{\rm d}/(u_{\rm c}-u_{\rm d}))^2/(u_{\rm c}\, M_1\, (1-\Gamma_1))$,
$dh/du = 4\, M_2\, (1-\Gamma_2)\, (dA/du - A_{\rm c}/u_{\rm c})/(u_{\rm c}\, M_1\, (1-\Gamma_1))$,
$B_{\rm c} = K_1 - K_2\, A_{\rm c}\, M_2\, \Gamma_2/(M_1\, \Gamma_1)$,
$dB/du = dK_1/du - M_2\, \Gamma_2\, ((dK_2/du)\, A_{\rm c} + (dA/du)\, K_2)/(M_1\, \Gamma_1)$,
$A_{\rm c} = (u_{\rm d}/(u_{\rm c}-u_{\rm d}))^2$,
$dA/du = -2\, u_{\rm d}/(u_{\rm c}-u_{\rm d})^3$,
$K_1 = (1 - (1 - (R_1\, u_{\rm c}/C_{\rm z})^2)^{1+\alpha})/((1+\alpha)\, (R_1\, u_{\rm c}/C_{\rm z})^2)$,
$dK_1/du = 2\, R_1\, ((1 - (R_1\, u_{\rm c}/C_{\rm z})^2)^{\alpha}\, (1+\alpha)\, (R_1\, u_{\rm c}/C_{\rm z})^2 - 1 + (1 - (R_1\, u_{\rm c}/C_{\rm z})^2)^{1+\alpha})/(C_{\rm z}\, (1+\alpha)\, (R_1\, u_{\rm c}/C_{\rm z})^3)$,
$K_2 = (1 - (1 - (R_2/(d + C_{\rm z}/u_{\rm c}))^2)^{1+\alpha_2})/((1+\alpha_2)\, (R_2/(d + C_{\rm z}/u_{\rm c}))^2)$,
$dK_2/du = (2\, C_{\rm z}/((d + C_{\rm z}/u_{\rm c})\, u_{\rm c}^2))\, ((1 - (R_2/(d + C_{\rm z}/u_{\rm c}))^2)^{\alpha_2}\, (1+\alpha_2)\, ((R_2/(d + C_{\rm z}/u_{\rm c}))^2) - 1 + (1 - (R_2/(d + C_{\rm z}/u_{\rm c}))^2)^{1+\alpha_2})/((1+\alpha_2)\, (R_2/(d + C_{\rm z}/u_{\rm c}))^2)$ and
$C_{\rm z} = 2G\, M_1\, (1-\Gamma_1)/a^2$.
In these equations, α_2 is the Abbott (1982) parameter for the companion star.

The density is computed as $\rho = (d\dot{M}/d\Omega)/(4\pi\, v\, r_1^2)$ with $d\dot{M}/d\Omega$ the stagnation mass-loss rate:

$$\frac{d\dot{M}}{d\Omega} = \dot{M}_1 \left(1 - \alpha \left(\frac{u_{\rm d}}{u_{\rm c} - u_{\rm d}}\right)^2 (c_2 - c_1(1-\alpha))\right) \quad (1.23)$$

with $c_1 = M_1\, (1-\Gamma_1)/(M_2\, (1-\Gamma_2))$, $c_2 = L_1\, K_{1{\rm c}}/(L_2\, K_{2{\rm c}})$, $K_{2{\rm c}} = (1 - (1 - (R_2/(d - 1.05\, R_2))^2)^{1+\alpha_2})/((1+\alpha_2)\, (R_2/(d-1.05\, R_2))^2)$ and $K_{1{\rm c}} \sim (1 - 0.093^{1+\alpha})/(0.91(1+\alpha))$. $u_{\rm c} = -2G\, M_1\, (1-\Gamma_1)/(r_{\rm c}\, a^2)$ is the velocity at the critical point in the (u,w) coordinates where $u = -2GM_1(1-\Gamma_1)/(r_1\, a^2)$ and $w = v^2/a^2$ (Abbott 1980). $u_{\rm d}$ is the velocity at a distance d given by the same formula.

1.3.2 Outside the line of centers

We then solve the variation of the wind velocity outside the line of centers with an angle $\theta_1 \in]0, \pi]$. The distance between the star 2 and a point with an angle θ_1 and a distance r_1 from the star 1 is $r_2 = (d^2 + r_1^2 - 2\, r_1\, d\, \cos\theta_1)^{0.5}$. The angle between this point and the x-axis as viewed from the star 2 is $\theta_2 = {\rm acos}((d - r_1\, \cos\theta_1)/r_2)$ (see Fig. 1.1).

We have to solve an equation describing the r component of Eq. 1.11 (first row of the matrices):

$$\begin{aligned}&\left(1 - \frac{a^2}{\gamma v^2}\right) u_{\rm r} \frac{\partial u_{\rm r}}{\partial r} - \frac{u_\theta^2}{r_1} + \frac{u_\theta}{r_1}\frac{\partial u_{\rm r}}{\partial \theta} - \frac{a^2\, u_\theta}{\gamma v^2}\frac{\partial u_\theta}{\partial r} - \frac{G\, M_2\, (1-\Gamma_2)\, \cos(\theta_1+\theta_2)}{r_2^2} + \frac{G\, M_1\, (1-\Gamma_1)}{r_1^2} \\ &- g_{\rm rad}\, (F_1\, K_1 - F_2\, K_2\, \cos(\theta_1+\theta_2)) - 2\frac{a^2}{\gamma\, r_1} = 0\,,\end{aligned} \quad (1.24)$$

and an equation describing the θ component of Eq. 1.11 (second row of the matrices):

$$\left(1 - \frac{a^2}{\gamma v^2}\right) \frac{u_\theta}{r_1}\frac{\partial u_\theta}{\partial \theta} + \frac{u_\theta\, u_{\rm r}}{r_1} + u_{\rm r}\frac{\partial u_\theta}{\partial r} - \frac{a^2\, u_{\rm r}}{\gamma v^2 r_1}\frac{\partial u_{\rm r}}{\partial \theta} - \frac{G\, M_2\, (1-\Gamma_2)\, \sin(\theta_1+\theta_2)}{r_2^2} + g_{\rm rad}\, F_2\, K_2\, \sin(\theta_1+\theta_2) = 0 \quad (1.25)$$

with $\rho = (d\dot{M}/d\Omega)/(4\pi\, v\, r_1^2)$ and $g_{\rm rad} = \frac{\sigma_{\rm e}^{1-\alpha}\, k}{c\, (V_{\rm th}\, \rho\, v)^{\alpha}} \left(\left|u_{\rm r}\frac{\partial u_{\rm r}}{\partial r} + u_\theta \frac{\partial u_\theta}{\partial r}\right|\right)^{\alpha}$. Points located at $\theta_1 > \pi/2$ and $\theta_2 < {\rm atan}(R_1/d)$ are not influenced by the radiative pressure from star 2.

We use the finite-differences method to solve these equations: at a step m, $\partial f/\partial z = (f_{\rm m} - f_{\rm m-1})/\Delta z$. In 2D, a point can be defined by its position (i, j) on the grid with i the step along the first direction and j the

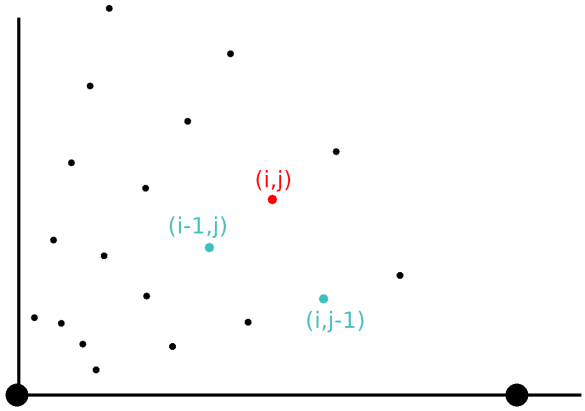


Figure 1.2: Representation of the polar grid used to compute the stellar wind distribution.

step along the second direction. In polar coordinates, let u_r and u_θ the radial and angular component of the wind velocity at the grid point (i,j) (see Fig. 1.2). Let u_rr and $u_{\theta\mathrm{r}}$ the radial and angular component at the point $(i-1,j)$. Let $u_{\mathrm{r}\theta}$ and $u_{\theta\theta}$ the radial and angular component at the point $(i,j-1)$. The equations we have to solve to determine u_r and u_θ are thus:

$$\begin{aligned} eq_1 = &\left(1-\frac{a^2}{\gamma v^2}\right) u_\mathrm{r}\,\frac{u_\mathrm{r}-u_\mathrm{rr}}{\Delta r} - \frac{u_\theta^2}{r_1} + \frac{u_\theta}{r_1}\,\frac{u_\mathrm{r}-u_{\mathrm{r}\theta}}{\Delta\theta} - \frac{a^2\,u_\theta}{\gamma v^2}\,\frac{u_\theta-u_{\theta\mathrm{r}}}{\Delta r} - \frac{G\,M_2\,(1-\Gamma_2)\,\cos(\theta_1+\theta_2)}{r_2^2} \\ &+ \frac{G\,M_1\,(1-\Gamma_1)}{r_1^2} - g_\mathrm{rad}\,(F_1\,K_1 - F_2\,K_2\,\cos(\theta_1+\theta_2)) - 2\frac{a^2}{\gamma\,r_1}\,, \end{aligned} \tag{1.26}$$

$$\begin{aligned} eq_2 = &\left(1-\frac{a^2}{\gamma v^2}\right) \frac{u_\theta}{r_1}\,\frac{u_\theta-u_{\theta\theta}}{\Delta\theta} + \frac{u_\theta\,u_\mathrm{r}}{r_1} + u_\mathrm{r}\frac{u_\theta-u_{\theta\mathrm{r}}}{\Delta r} - \frac{a^2\,u_\mathrm{r}}{\gamma v^2 r_1}\,\frac{u_\mathrm{r}-u_{\mathrm{r}\theta}}{r_1\Delta\theta} - \frac{G\,M_2\,(1-\Gamma_2)\,\sin(\theta_1+\theta_2)}{r_2^2} \\ &+ g_\mathrm{rad}\,F_2\,K_2\,\sin(\theta_1+\theta_2) \end{aligned} \tag{1.27}$$

with $g_\mathrm{rad} = \frac{\sigma_\mathrm{e}^{1-\alpha}\,k}{c\,(V_\mathrm{th}\,\rho\,v)^\alpha}\left(\left|u_\mathrm{r}\frac{u_\mathrm{r}-u_\mathrm{rr}}{\Delta r} + u_\theta\frac{u_\theta-u_{\theta\mathrm{r}}}{\Delta r}\right|\right)^\alpha$. The (u_r, u_θ) are found to minimize $(eq_1^2+eq_2^2)^{0.5}$ using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm.

The derivative of these equations are

$$\begin{aligned}
\frac{\partial eq_1}{\partial u_\mathrm{r}} =& \frac{(2u_\mathrm{r}-u_\mathrm{rr})(\gamma u_\theta^2-a^2)+\gamma u_\mathrm{r}^2(4u_\mathrm{r}-3u_\mathrm{rr})}{\Delta r\,\gamma v^2} + \frac{2u_\mathrm{r}(a^2u_\theta(u_\theta-u_{\theta\mathrm{r}})-u_\mathrm{r}(u_\mathrm{r}-u_\mathrm{rr})(v^2\gamma-a^2))}{\Delta r\,\gamma v^4} + \frac{u_\theta}{r_1\,\Delta\theta} & -C(F_1\,K_1-F_2\,K\\
\frac{\partial eq_1}{\partial u_\theta} =& \frac{-a^2(2u_\theta-u_{\theta\mathrm{r}})+2\gamma u_\mathrm{r}u_\theta(u_\mathrm{r}-u_\mathrm{rr})}{\Delta r\,\gamma v^2} + \frac{2u_\theta(a^2u_\theta(u_\theta-u_{\theta\mathrm{r}})-u_\mathrm{r}(u_\mathrm{r}-u_\mathrm{rr})(v^2\gamma-a^2))}{\Delta r\,\gamma v^4} - \frac{2u_\theta}{r_1} + \frac{u_\mathrm{r}-u_\mathrm{rr}}{r_1\Delta\theta} & -C(F_1\,K_1-F_2\,K\\
\frac{\partial eq_2}{\partial u_\mathrm{r}} =& \frac{-a^2(2u_\mathrm{r}-u_{\mathrm{r}\theta})+2\gamma u_\mathrm{r}u_\theta(u_\theta-u_{\theta\theta})}{r_1\,\Delta\theta\,\gamma v^2} + \frac{2u_\mathrm{r}(a^2u_\mathrm{r}(u_\mathrm{r}-u_{\mathrm{r}\theta})-u_\theta(u_\theta-u_{\theta\theta})(v^2\gamma-a^2))}{r_1\Delta\theta\,\gamma v^4} + \frac{u_\theta}{r_1} + \frac{u_\theta-u_{\theta\mathrm{r}}}{\Delta r} & +C\,F_2\,K_2\,\delta_2(2u_\mathrm{r}\\
\frac{\partial eq_2}{\partial u_\theta} =& \frac{(2u_\theta-u_{\theta\theta})(\gamma u_\mathrm{r}^2-a^2)+\gamma u_\theta^2(4u_\theta-3u_{\theta\theta})}{r_1\Delta\theta\,\gamma v^2} + \frac{2u_\theta(a^2u_\mathrm{r}(u_\mathrm{r}-u_{\mathrm{r}\theta})-u_\theta(u_\theta-u_{\theta\theta})(v^2\gamma-a^2))}{r_1\Delta\theta\,\gamma v^4} + u_\mathrm{r}\left(\frac{1}{\Delta r}+\frac{1}{r_1}\right) & +C\,F_2\,K_2\,\delta_2(2u_\theta
\end{aligned} \tag{1.28}$$

with $C = \frac{\alpha\,\sigma_\mathrm{e}\,k}{c}\left(\frac{4\pi r^2}{\sigma_\mathrm{e}\,V_\mathrm{th}\,dM/d\Omega\,\Delta r}\right)^\alpha (u_\mathrm{r}(u_\mathrm{r}-u_\mathrm{rr})+u_\theta(u_\theta-u_{\theta\mathrm{r}}))^{\alpha-1}$. The Jacobian of $(eq_1^2+eq_2^2)^{0.5}$ is thus

$$J = (eq_1^2+eq_2^2)^{-0.5}\begin{pmatrix} eq_1\frac{\partial eq_1}{\partial u_\mathrm{r}} + eq_2\frac{\partial eq_2}{\partial u_\mathrm{r}} \\ eq_1\frac{\partial eq_1}{\partial u_\theta} + eq_2\frac{\partial eq_2}{\partial u_\theta} \end{pmatrix} \tag{1.29}$$

An example of wind distribution solved by this method is shown in Fig. 1.3.

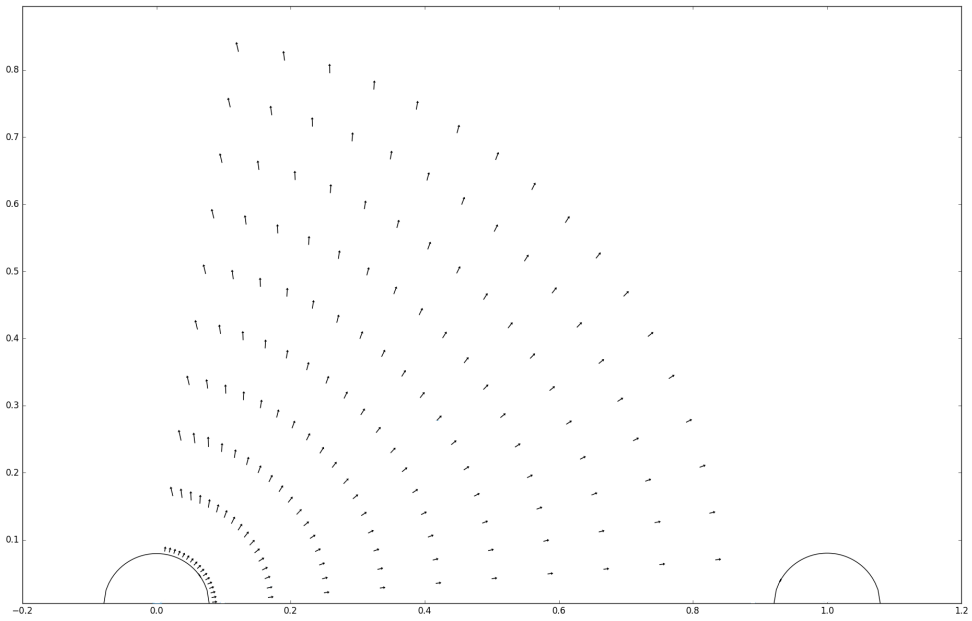


Figure 1.3: Effect of the radiative inhibition on the wind of star 1 with $d = 150\,\mathrm{R}_\odot$, $M_1 = M_2 = 26\,\mathrm{M}_\odot$, $R_1 = R_2 = 12\,\mathrm{R}_\odot$ and $\dot{M}_1 = 0.5\,10^{-6}\,\mathrm{M}_\odot\,\mathrm{yr}^{-1}$.

Chapter 2

Simulation of the wind shock

2.1 Position of the contact discontinuity

The contact discontinuity (or discontinuity surface) is the very thin surface (of the order of the mean free path of the particles) separating the two hydrodynamic shocks created by each wind.

In adiabatic shocks, it is often supposed that the winds have reached their terminal velocities. However, because of the radiative inhibition described in Sect. 1, the winds may not have reached their terminal velocities before colliding, especially in the region close to the line of center. We thus can not use the analytical solution of Canto et al. (1996) to compute the position of the contact discontinuity.

However, to get a good sampling of the shock in 2D, we use the Canto et al. (1996)'s model as a first approximation. We consider a range of angles θ_1 from 0 to the opening angle of the shock θ_∞ with 30 steps. The angle θ_∞ is defined as

$$\theta_\infty - \tan\theta_\infty = \frac{\beta\pi}{\beta - 1} \tag{2.1}$$

with $\beta = \dot{M}_2\, v_{\infty,2}/(\dot{M}_1\, v_{\infty,1})$ the wind momentum ratio considering the terminal velocities v_∞ computed as 2.6 times the escape velocity (Kudritzki & Puls 2000). The radial distance r_1 from the star 1 is

$$r_1 = d\sin\theta_2 \csc(\theta_1 + \theta_2) \tag{2.2}$$

with θ_1 and θ_2 the angles between the position of the point on the discontinuity surface and the line of center from star 1 and 2, respectively (see Fig. 2.1). θ_2 is defined from θ_1:

$$\theta_2 \cot\theta_2 = 1 + \frac{\theta_1 \cot\theta_1 - 1}{\beta} \tag{2.3}$$

We then use the formalism of Antokhin et al. (2004) to compute the exact solution of the position of the contact discontinuity considering that the winds do not have reached their terminal velocities before colliding. The stagnation point xs of the contact discontinuity is the result of the non-linear equation $x = d/(1+\sqrt{\eta(x,0)})$ with the wind force balance varying with the distance from the star 1:

$$\eta(x,y) = \frac{\dot{M}_2\, v_2(x,y)}{\dot{M}_1\, v_1(x,y)} \tag{2.4}$$

with $v_1(x,y)$ and $v_2(x,y)$ the velocities of the winds. The positions of the discontinuity surface are the solutions of the differential equation:

$$\frac{dx}{dy} = \frac{1}{y}\left(x - \frac{d\, r_1^2(x,y)\sqrt{\eta(x,y)}}{r_1^2(x,y)\sqrt{\eta(x,y)} + r_2^2(x,y)}\right) \tag{2.5}$$

with $r_1(x,y)$ and $r_2(x,y)$ the distances to the star 1 and 2, respectively (see Fig. 2.1). The values of y for which we will compute the corresponding value of x are given by $r_1 \sin\theta_1$.

2.2 Characteristics of the hydrodynamic shocks

The evolution of the characteristics of the hydrodynamic shocks depends on the cooling regime of the plasma constituting them. Stevens et al. (1992) established a criterion to determine whether the shock is in an adiabatic or in a radiative regime: the shock is considered as adiabatic if $v^4 xs/\dot{M} > 1$ with $\dot{M}$ the mass-loss rate in $10^{-7}\,\mathrm{M_\odot\,yr^{-1}}$ and v the pre-shock velocity in $1000\,\mathrm{km\,s^{-1}}$ at the stagnation point xs (in 10^7 km).

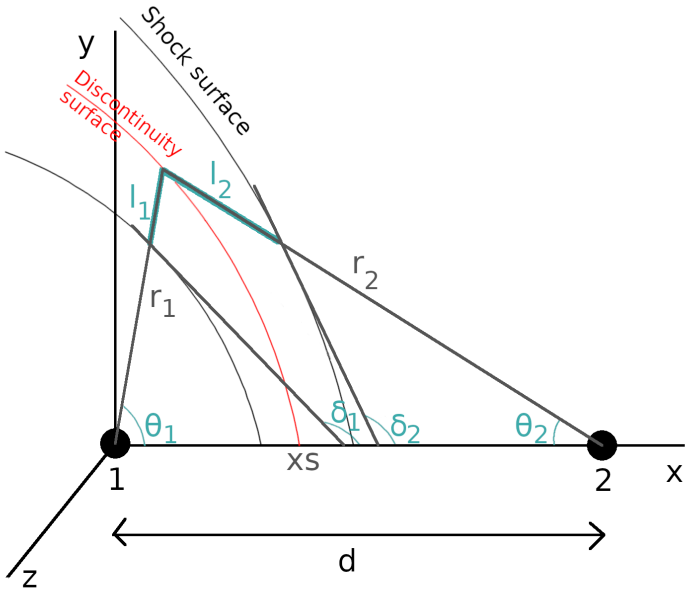


Figure 2.1: Schematic view of the wind shock region. The star 1 orbits around the star 2 in the direction of the z-axis.

2.2.1 Adiabatic shock

The width of the shock is computed as the ratio between the surface density and the volume density at the discontinuity surface. The surface density is

$$\sigma = \frac{\sigma_0 \sin(\theta_1+\theta_2)\csc\theta_1 \csc\theta_2 (\beta(1-\cos\theta_1)+\frac{v_2(x,y)}{v_1(x,y)}(1-\cos\theta_2))^2}{\sqrt{(\beta(\theta_1-\sin\theta_1\cos\theta_1)+(\theta_2-\sin\theta_2\cos\theta_2))^2+(\beta\sin^2\theta_1-\sin^2\theta_2)^2}} \tag{2.6}$$

with $\sigma_0 = \dot{M}_1/(2\pi\beta d v_1(x,y))$. The volume density is the average density of the mixed gas inside the interaction region:

$$\rho = 0.5\left(\frac{\dot{M}_1}{v_1(x,y)\pi r_1^2}+\frac{\dot{M}_2}{v_2(x,y)\pi r_2^2}\right) \tag{2.7}$$

with $r_2 = r_1 \sin\theta_1/\sin\theta_2$ the distance from the star 2 to the discontinuity surface.

The temperature at the shock is

$$kT_\mathrm{i} = \frac{3 m_\mathrm{p} v_\mathrm{p,i}^2}{16 k_\mathrm{b}} \tag{2.8}$$

with k_B the Boltzmann constant and $v_{\mathrm{p},1} = v_1(x,y)\sin(\delta_1-\theta_1)$ and $v_{\mathrm{p},2} = v_2(x,y)\sin(\pi-\delta_2-\theta_2)$ the component of the velocities which is normal to the shock surface with δ_i the slope of the shock surface.

2.2.2 Radiative shock

The width of the shock on the side of star i is

$$l_\mathrm{i} = \frac{15\,\bar{\mu}^2 m_\mathrm{p}^2}{512\sum(X_z Z X_H)}\frac{(v_\mathrm{i}(x,y)\sin(\delta_\mathrm{i}))^3}{\rho_0 \lambda(T_0)} \tag{2.9}$$

with x and y the position of the shock surface, $\bar{\mu} = 1.3$, $\sum(X_z Z X_H) = 0.99$, $\rho_0 = 4\dot{M}_\mathrm{i}/(v_\mathrm{i}(x,y)4\pi r_\mathrm{i}^2)$ the post-shock density and $\lambda(T_0)$ the cooling function for $T_0 = 1.21(\mu/0.62)(v_\mathrm{i}(x,y)\sin\delta_\mathrm{i})^2$ the post-shock temperature.

From the post-shock temperature, we compute the evolution of the temperature across the interaction zone as

$$\frac{dT}{dl} = -C(T)\frac{\lambda(T)}{T^2} \tag{2.10}$$

with $C(T) = 9\sum(X_z Z X_H)\mu^3 m_\mathrm{p}\rho_0\,(v_\mathrm{i}(x,y)\sin\delta_\mathrm{i})^3/(40\bar{\mu}^2 k_\mathrm{B}^3)$, k_B the Boltzmann constant and $v_\mathrm{i}(x,y)$ the velocity at the shock surface. The temperature is thus decreasing towards the discontinuity surface.

In the same manner, considering an isobaric gas, the variation of the ideal gas law over the shock width is:

$$\frac{k}{\mu\, m_\mathrm{p}}\left(\rho\frac{dT}{dl} + T\frac{d\rho}{dl}\right) = 0 \tag{2.11}$$

Considering Eq. 2.10 we have:

$$\frac{d\rho}{dl} = C(T)\rho\frac{\lambda(T)}{T^3} \tag{2.12}$$

The density is thus increasing towards the discontinuity surface.

The computation of the surface density depends on the off-axis angle from the line of centers. Close to the line of centers ($y < 0.2xs$), we use Eq. A2 or A3 of Antokhin et al. (2004) depending on the side of the interaction region to compute the parameter σ_0. The surface density is thus

$$\sigma = \sigma_0\left(1 - \frac{y^2(1+2xs\,z_0)}{6xs^2}\right) \tag{2.13}$$

with

$$z_0 = \frac{\frac{4\sqrt{1-\sqrt{1/\eta}}}{xs\sqrt{1/\eta}} + \frac{xs\sqrt{1/\eta}(c_1-c_2)}{1+\sqrt{1/\eta}}}{6 - xs^2\sqrt{1/\eta}\frac{c_1+c_2\sqrt{1/\eta}}{1+\sqrt{1/\eta}}} \tag{2.14}$$

with $c_1 = \beta_1/(xs^2((xs/R_1)-1))$ and $c_2 = \beta_2/((d-xs)^2((d-xs/R_2)-1))$ and β_i the parameter of the beta-law of velocities.

Far from the line of centers ($y > 0.2xs$), we solve the differential equation A7 of Antokhin et al. (2004) to compute the auxiliary function ζ_i from the value of σ when $y = 0.2xs$:

$$\frac{d\zeta_\mathrm{i}}{dy} = \frac{\dot{M}_\mathrm{i}\,\cos\theta_\mathrm{i}\,y}{4\pi\, r_\mathrm{i}^2\,\sin\delta_\mathrm{i}}\,. \tag{2.15}$$

The surface density is:

$$\sigma = \frac{\zeta_\mathrm{i}}{y\, v_\mathrm{i}(x,y)\cos\delta_\mathrm{i}} \tag{2.16}$$

2.3 Coriolis deflection

The methodology is based on the work of Parkin & Pittard (2008). The shock is first divided into the shock cap and the shock tail. We define the shock cap as the region where the tangential velocity is lower than 85% of the lowest terminal velocity. The Coriolis force has two effects on the shape of the shock: the skewing of the entire shock around the center of mass of the system and the curvature of the shock tail.

The skewing angle is:

$$s = \arctan\left(\frac{v_\mathrm{orb}}{v_1(xs,0)}\right) \tag{2.17}$$

with $v_1(xs,0)$ the wind velocity of the star 1 at the stagnation point and $v_\mathrm{orb} = \sqrt{G\,(M_1+M_2)\,(2/d-1/a)}$ the orbital velocity of the star with d the distance between the stars and a the semi-major axis of the orbit.

The deflection (curvature) is then computed going back along the orbit of the stars. Indeed, at a time T_1, considering the stars at rest, the particles at the end of the shock cap move freely with a rectilinear trajectory. At a time $T_2 > T_1$, the stars have moved while the particles have continued their a rectilinear trajectory. At this time, a new shock cap is created. Particles at the end of the shock cap are also ejected along a rectilinear trajectory. The stellar motion continues leading to the escaping particles creating a tail of the shock which is curved behind the star 1.

The tail of the shock is thus created by considering several times in the past, computing the shock cap and deducing the rectilinear trajectory of the particles at the end of the shock cap. The distance traveled by the particles is proportional to the velocity they had as they left the shock cap and to the time interval between the past times and the present time.

The creation of the tail is shown in Fig. 2.2.

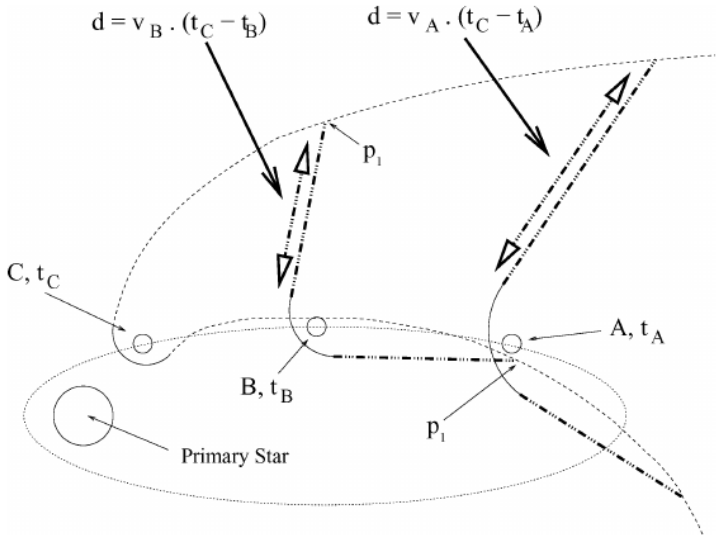


Figure 2.2: Schematic view of the Coriolis deflection (Figure 7 of Parkin & Pittard 2008). Here, $t_A < t_B < t_C$. Note that their primary star is the star 2 in our code.

Chapter 3

Computation of the line profile

The line profile is built as an histogram of the flux received by the observer as a function of the radial velocities of the cells of material that emit the received photons. The radial velocity is the velocity of the cell along the line of sight. The velocity of the cell is defined as the velocity component of the emitting material tangential to the shock surface. The observed flux from each cell is a combination of emission and absorption.

Let's consider a cell inside the wind interaction region. This cell is characterized by a temperature T, a volume V and a volume density ρ. The temperature of the cell defines the emissivity (q for its logarithm in base 10) depending on the studied ion. The density defines the electron ($n_\mathrm{e} = 1.17 n_\mathrm{H}$) and particle densities (n_H) with, considering solar abundance,

$$n_\mathrm{H} = \frac{\rho}{1.34 m_\mathrm{H}} \,. \tag{3.1}$$

The absorption depends on the medium crossed by the photon. If the photon crosses a cell inside the shock, the cross-section cs of the hot plasma inside the crossed cell depends on the temperature of this cell and the energy of the photon. The cross-sections are taken from by the AtomDB database (Foster et al. 2012) for several temperatures of the plasma and several energies of photons. The optical depth of the crossed cell is

$$\tau_\mathrm{shock} = \frac{cs\,\sigma}{1.3 m_\mathrm{p}} \frac{1}{\vec{P} \,.\, \vec{N}} \tag{3.2}$$

with σ the surface density, $\vec{P}$ the position vector of the cell and $\vec{N}$ the direction of the observer.

The optical depth due to the cool, unshocked wind material along the line of sight is computed by the method presented by Rauw (2007) assuming a simplified wind velocity law $v(r) = v_\infty(1 - R/r)$. We first create a new coordinate system centered on the star that is located in front at the considered orbital phase. The xt axis points towards the observer (characterized by an inclination i and a phase ϕ) and the axis yt and zt are perpendicular to xt (see Fig. 3.1). The coordinate pt is defined as $\sqrt{yt^2 + zt^2}$. We test whether or not the cell is occulted by the body of any of the stars, and identify the star whose wind is in front of the cell along the line of sight. The optical depth is thus:

$$\tau_\mathrm{wind} = \begin{cases} \infty & \text{for } pt < R \\ \dfrac{2\tau_0}{\alpha - 1} & \text{for } pt = R \\ \dfrac{2\tau_0}{\sqrt{pt^2/R^2 - 1}} \left(\dfrac{\pi}{2} - \arctan\left(\dfrac{\alpha\, pt/R - 1}{\sqrt{pt^2/R^2 - 1}} \right) \right) & \text{for } pt > R \end{cases} \tag{3.3}$$

with

$$\alpha = \begin{cases} \tan\left(\dfrac{\arctan\left(pt/xt\right)}{2} \right) & \text{for } xt < 0 \\ \tan\left(\dfrac{\pi}{4} \right) & \text{for } xt = 0 \\ \tan\left(\dfrac{\pi - \arctan\left(pt/xt\right)}{2} \right) & \text{for } xt > 0 \end{cases} \tag{3.4}$$

and $\tau_0 = \kappa_\lambda \dot{M}_\mathrm{i} / (4\pi\, v_\mathrm{i}\, R_\mathrm{i})$ with κ_λ computed from the model of Nazé et al. (2004) and v_i the velocity of the wind at the considered photon position. In case the absorption is produced by the winds of both stars, the contributions of the optical depths of both stars are added together.

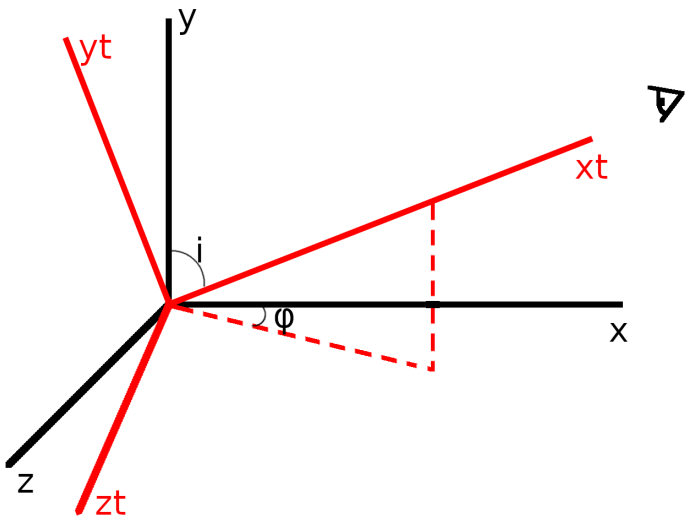


Figure 3.1: Coordinate system used to compute the absorption by the stellar winds.

Note that this definition of the new coordinate system will provide a positive radial velocity when the material in the cell is moving moving towards the observer. However, by convention, a positive radial velocity defines material receding from the observer. We thus invert the sign of the radial velocity in the code. Since the redshift (blueshift) is defined when an object is receding (approaching) the observer, the conversion between the radial velocity and the wavelength shift is defined as:

$$v_{\rm rad} = \frac{\lambda_{\rm m} - \lambda_0}{\lambda_0} c \tag{3.5}$$

with λ_0 the wavelength of the line at rest and $\lambda_{\rm m}$ the observed wavelength.

We finally test the visibility of each cell. Indeed, if the cell is hidden by one of the stars, its emission will not be recorded (which is equivalent to setting the wind optical depth to ∞). If the emitting cell is visible, the absorbed emission by the cell of volume V and of radial velocity $v_{\rm rad}$ with respect to the observer, computed as $E = 10^q \, n_{\rm H} \, n_{\rm e} \, V \, e^{-\sum(\tau_{\rm shock}+\tau_{\rm wind})}$ is added to the histogram of the observed fluxes at a Doppler shift corresponding to $v_{\rm rad}$.

Chapter 4

The code

4.1 Download LIFELINE

LIFELINE is an interactive program written in Python (v2.7) and located on GitHub [1]. LIFELINE is collaborative: if the user computes the line profile for a new binary, a new inclination or phase, or a new ion, he/she can increase the database for future users.

4.2 Organization of LIFELINE

The repository is composed of the following files:

- filemap: files used by LIFELINE to locate the files used to read the emissivities
- XIFU_cc_baselineconf_thickfilter_2018_10_10.arf and XIFU_cc_baselineconf_thickfilter_2018_10_10.rmf: the response matrix and ancillary response files used to compute the line profile as observed by X-IFU onboard the future Athena mission.
- code_params.txt: an example of input file for the main code
- cross_section.tab: the file containing the cross-section values
- Manual.pdf: this manual
- stellar_params_set.txt: the binary systems for which the winds, the shocks and the line profiles have already been computed
- stellar_params.txt: an example of the binary system defined by the user
- GNU_GENERAL_PUBLIC_LICENSE_v3: the GNU General Public License
- ionisation_fraction.tab: the ionization balance file

The repository also contains the following directories:

- codes: where the LIFELINE codes are recorded
- cooling_functions: where the already computed cooling functions are saved
- winds_set: contains the wind distribution of the binary systems listed in stellar_params_set.txt
- plots_set: contains the plots of the wind shock of the binary systems listed in stellar_params_set.txt
- histograms_set: contains the files used to compute the line profiles of the binary systems listed in stellar_params_set.txt
- Fe_XXV_[reson/intercomb1/intercomb2/forbid]_set and Fe_XXVI_set: contains the line profiles of the iron XXV helium-like triplet and the iron XXVI hydrogen-like ion for the binary systems listed in stellar_params_set.txt

[1] https://github.com/emossoux/LIFELINE

4.3 Prerequisite

The `Cloudy` program [2] is needed to construct the cooling function of a specific atomic species. This program can be freely downloaded. In the file `cooling_func.in` located in the `codes` directory, the user has to change the fourth line (species "Fe26" levels=all) according to the ion he wants to study (Lykins et al. 2013). The user then runs `Cloudy` from the directory where the file `cooling_func.in` is located with the command line: `/PATH/cloudy.exe cooling_func.in`. Cooling functions were already computed with the `Cloudy` version 17 for all levels of the ions emitting between 2 and 10 keV considering the solar abundances from Grevesse et al. (2010) in a collisonally ionized gas. The resulting files named `cooling_function_[ion].tab` are saved in the directory `cooling_functions`. If a new cooling function file is created by the user, it can be saved in the `cooling_functions` directory and renamed as `cooling_function_[ion].tab`. The main code will then copy and rename the file of the corresponding ion as `cooling_function.tab` in the main directory.

The radiative codes also need the Python interface of ATOMDB: pyatomdb [3] which will use the file map `filemap` given in the repository of LIFELINE.

The user must also write a file containing the stellar parameters of the studied binaries. The values are (in order and space-separated): the spectral type, mass-loss rate (in $M_\odot\, yr^{-1}$), the mass (in $M_\odot$), the radius (in $R_\odot$) and the effective temperature (in K) of each star, the semi-major axis (a in $R_\odot$), the eccentricity and the argument of periapsis (in degree) of the orbit, and the phase (from 0 to 1) and the inclination (in degree) at which the line profile must be computed. For example:
O3I O5I 2.28559880338e-06 1.07398941234e-06 58.34 37.28 13.84 11.08 44616 41540 1327 0.5 170 0.1 60
The file can contain several lines to define several systems or several phases and inclinations. LIFELINE will compute the line profiles for each set of parameters.

To compute the radial velocity of each cell in the shock, the phase $\phi = 0$ is defined at the conjunction (i.e., when the radial velocities of the stars are equal to zero) with the star with the most powerful wind in front. This definition is the same as usually used for circular orbits (the argument of the periapsis is thus $\omega = 0$ in this case). But, for eccentric orbits, the phase is usually defined to 0 at the periapsis (this is the convention used for the computation of the distance between the stars). The code will thus convert the phase given in the stellar parameter file with the latter convention to the convention used in the code with:

$$E_{\rm conj} = 2 \arctan\left(\sqrt{\frac{1-e}{1+e}}\, \tan\left(\frac{0.5\pi - \omega}{2}\right)\right) \tag{4.1}$$

$$M_{\rm conj} = E_{\rm conj} - e\, \sin(E_{\rm conj}) \tag{4.2}$$

with $E_{\rm conj}$ and $M_{\rm conj}$ the eccentric and mean anomaly at the first conjunction. A particular attention must thus be taken in this case: if the first star listed in the parameter file (the star O3I in our example) is the star in front at the first conjunction (i.e. the first conjunction occurring after the periapsis), the phase used in the code will be $\phi = \phi_{\rm file} - (M_{\rm conj} + \pi)$. Otherwise, the phase used in the code will be $\phi = \phi_{\rm file} - M_{\rm conj}$ with $\phi_{\rm file}$ the phase referenced in the parameter file.

For binaries where the Coriolis effect is not taken into account, if the user is not sure about the position of the stars at the conjunction, he/she can list the characteristics of the binary in the stellar parameters file considering a circular orbit and the distance between the stars equal to the distance at the considered phase.

4.4 The main file: `line_profile.py`

LIFELINE is interactive so that no parameter has to be changed in any of the code files. The main code file is named `line_profile.py`. The code parameters can be given as arguments when the main file is called with:
`python /PATH/line_profile.py [code_parameter_file]`
The optional code parameter file is a text file containing the parameters needed to run the code (one parameter per line). The parameters are:

- atom - Element for which you want to compute the line (ex: Fe, Ca,...)
- ion - Ion of this element (ex: 25, 14, ...)
- energy - Rest energy of the line [keV]

[2] nublado.org
[3] pypi.org/project/pyatomdb

- directory - Path where to save the results. This directory must contain the *_set directories, the cross-section file `cross_section.tab`, the cooling function table `cooling_function.tab`, the stellar parameter file, the `apec_linelist.fits` file, and optionally the ionization fraction file `ionisation_fraction.tab` if you use the already computed file (crea_ion_file=no) or the `filemap` file if not. This can be the directory where the user saved the repository.
- param - File containing the stellar parameters.
- mode - In which mode do you want to compute? [1/2/3] 1 - Perform the overall computation, i.e., the velocity distribution, the shock characteristics and the line profile; 2 - Compute shock characteristics and the line profile using pre-computed; 3 - Compute only the line profile using the pre-computed shock characteristics and velocity distribution.
- crea_ion_file - compute the ionization fraction file for radiative shocks? [yes/no]
- convolve - Convolve the theoretical light curve with the instrumental response? [yes/no]
- direct_rmf_arf - Only if convolve=yes. Complete path to the response matrix file (RMF) and ancillary response file (ARF)
- RMF - Only if convolve=yes. Response matrix file
- ARF - Only if convolve=yes. Ancillary response file
- distance - Only if convolve=yes. The distance to the binary in kpc. Default value: 1.5
- mu - Mean molecular weight. Default value: 0.62 (fully ionized gas)
- H - Fractional mass abundance of H. Default value: 0.7381 (Sun)
- beta1 - Parameter of the beta-law of the wind velocity for the star 1. Default value: 1.
- beta2 - Parameter of the beta-law of the wind velocity for the star 2. Default value: 1.

An example of code parameter file is given in the repository with `code_params.txt`. If the name of the code parameter file is not given, LIFELINE will ask the parameters one by one interactively and create the `code_params.txt` so that it can be used for futur utilisation of the code.

The `crea_ion_file` parameter determines whether the ionization balance must be computed. This is used for the computation of the evolution of the temperature inside a radiative shock. The ionization balance was already tabulated for $Z \in [1, 26]$ and temperatures from 0.1 to 10 keV in the file `ionisation_fraction.tab`. This computation is long (about 14 h for one core on a computer with Intel Pentium CPU 3550M @ 2.30GHz x 2 and 7.7 Go of RAM). So, do not compute it if not necessary. Moreover, the pyatomdb package prints the path to all ion files leading to a very long list which is printed in the terminal. If you want to compute the ionization balance but without printing all these lines, you have to change the `atomdb.py` file given in the pyatomdb package (usually recorded in /usr/local/lib/python2.7/dist-packages/pyatomdb).

The `cross_section.tab` is a large file whose first line is a range of gas temperatures (from 0.5 to 10 keV) while the first column is a range of photon energies (from 0.05 to 10 keV) for which the cross-section was computed as following: for each temperature and energy, we performed a loop on Z until 27. For each Z, we loop on the temperatures where we read the ionization fraction assuming ionization equilibrium and the considered temperature from ATOMDB. The cross-section is the sum of the cross-sections of each ground level of each ion of the atom Z for the considered photon energy. The total cross-section for each gas temperature and photon energy is thus the sum of the cross-section of all atoms Z up to 27. If needed, this cross-section can be recomputed by running the code `cross_section.py` from the directory where the `ionisation_fraction.tab` file is located.

A file named `results_line_profile.txt` is created as an output reporting what LIFELINE did. Warning: this file will be overwritten at each run of the main code!

The main code first reads the emissivity in unit of photons $\mathrm{cm}^3\,\mathrm{s}^{-1}$ for the studied ion in the file `apec_linelist.fits`. It then computes the parameter q as the logarithm in base 10 of the emissivity in $10^{-23}\,\mathrm{erg}\,\mathrm{cm}^3\,\mathrm{s}^{-1}$. It then plots the emissivity as a function of the temperature in the file `Emissivity_[atom]_[ion]_[energy].pdf`. The cross-sections corresponding to the energy of the line are also read from the file `cross_section.tab`.

LIFELINE then reads the stellar parameters from the file `param`. By convention, LIFELINE was written considering the star 1 (at the origin of the coordinates) as the star with the less powerful wind. The code thus defines the star 1 according to the stellar parameters.

If asked (mode=1), LIFELINE then computes the velocity distribution accounting for the radiative inhibition by running the `radiative_inhibition.py` file (Sect. 4.5). If the user does not require the velocity

distribution computation, the velocities of both stars are read from the HDF5 file whose the spectral type and the distance between the stars are the closest to those of the studied binary (the set of already studied binaries with mode=1 is described in Sect. 4.12).

LIFELINE then launches the computation of the shock characteristics if required (mode=1 or 2). It first determines whether the shock is adiabatic or radiative (criterion in Sect. 2.2).

LIFELINE then determines if the Coriolis effects must be included in the wind shock construction. The Coriolis effect is included if $v_{\rm orb} > 0.1 \min(v_1, v_2)$ with $v_{\rm orb}$ the orbital velocity defined in Sect. 2.3.

One of the four configurations is then computed: adiabatic wind shock with or without Coriolis deflection or radiative wind shock with or without Coriolis deflection (Sect. 4.6 to 4.8). The shock characteristics (size, temperature, position, density,...) are computed before following the emitting photons to compute the absorption on each photon path (Sect. 4.10). These information are saved in different files in the directory `directory/histograms`.

The resulting files are then used to compute the line profile of any lines and, optionally, the convolution with the instrumental response (Sect. 4.11). The profiles are saved in a text file reporting the tangential velocities and corresponding emission as well as in a pdf file. These files are stored in a directory `directory/[atom]_[ion]`.

4.5 Simulation of the stellar winds

The simulation of the radiative inhibition of the stellar winds is performed by the file `radiative_inhibition.py`. The main equations used in this code are reported in Sect. 1. The parameters needed in this code are: the directory where the resulting files are saved, the period of the binary (P), the mass ratio, the distance between the stars (d), and the radii ($R_{\rm i}$), effective temperatures ($T_{\rm eff,i}$) and mass-loss rates ($\dot{M}_{\rm i}$) of each star. Since the code computes the distribution of the wind velocity for each star, two loops are performed if the stars have different spectral types.

The α and k parameters are taken from Abbott (1982) for the closest effective temperature and density. The mean density is computed for each star as $\dot{M}_{\rm i}/(m_{\rm H} v_{\rm i} 4\pi (0.5d)^2)$ with $v_{\rm i}$ the velocity at $0.5d$ computed with a beta-law. The stagnation mass-loss rate $d\dot{M}/d\Omega$ and the critical velocity $v_{\rm c}$ are then be computed with Eq. 1.23 and 1.22.

The luminosity of each star is computed as $L_{\rm i} = 4\sigma_{\rm k}\pi T_{\rm eff,i}^4 R_{\rm i}^2$ with $\sigma_{\rm k}$ the Stefan-Boltzmann constant. The Eddington ratios are thus:

$$\Gamma_{\rm i} = \frac{\sigma_{\rm t} L_{\rm i}}{M_{\rm i} \pi G c m_{\rm p}} \tag{4.3}$$

The velocity along the line of centers is computed from the critical point $r_{\rm c}$ to $2d$ using a radial step equal to $d/1000$. We use the finite-difference method to find, at each step, the value of the velocity minimizing Eq. 1.21. The companion star is first considered as point-like since the values of the velocity behind the companion will be used to compute the velocities outside the line of centers. The velocity of the points located behind the surface of the companion will then be set to zero.

Above the line of centers, we use an angular step of 1°. At the critical radius, we assume $u_\theta = 0$ and $u_{\rm r} = v_{\rm c}$ since the element of material feels only the radial forces from the star 1. The radial and angular component of the velocity are computed to minimize Eq. 1.26 and 1.27. We first solve the equations radially before increasing the angle θ. The angular and radial components of the velocity are then converted into Cartesian coordinates leading to $u_{\rm x}$ and $u_{\rm y}$ components.

The table of velocity components for each point are then saved into the HDF5 binary format with the name `wind_star[i]_par[ipar].h5` with *ipar* the line number of the stellar parameter file `param` (beginning at 0). Then, the code launches the `plots_winds.py` file to draw the wind distribution of each star. This plotting code can be launched separately in python with:

```
from plots_winds import plots
res=plots(R1,R2,vinf1,d,2,istar,ipar,direct)
```

with `R1` the stellar radii in cm of the considered star and `R2` the stellar radii of the companion star in cm, `d` the orbital separation in cm, `istar` equal to 1 or 2 for the plot of the wind of the primary or secondary star (you must thus launch the code twice for the two stars), and `direct` the complete path to the directory containing the h5 files. A representation of the wind for each star is also saved in the `wind_star[i]_par[ipar].pdf` file in the directory `winds`.

Note: The `radiative_inhibition.py` code is really slow (about nine hours for one core on a computer with Intel Pentium CPU 3550M @ 2.30GHz x 2 and 7.7 Go of RAM.). Future investigations must be done to speed up the code.

4.6 Simulation of the adiabatic shock

The shock characteristics are computed by the file `ashock.py` or `ashock_coriolis.py` depending on the computation of the Coriolis deflection. The main equations used in this code are reported in Sect. 2.2.1.

If the Coriolis deflection is computed, we first divide the shock in the shock cap and the shock tail else the overall shock is computed.

We first compute the general shape of the discontinuity surface in 2D with Eq. 2.5. We then consider the 3D positions of the shock assuming a symmetry around the line of centers. We then compute the width of the shock which depends on the separating distance between the stars. We also compute the tangential and normal component of the velocity of the wind of each star as well as the temperatures at the shock surface. If requested, we also compute the skewing and deflection on the shock positions and on the tangential velocities of the particles. If the shock is now below the surface of the star 1, we impose the shock to follow the surface of the star.

The characteristics of the medium crossed by the emitted photons are then computed (Sect. 4.10) to be used for the computation of the line profile.

4.7 Simulation of the radiative shock

The LIFELINE code first computes the ionization fraction file if requested (crea_ion_file=yes). The shock characteristics are computed by the file `rshock.py` or `rshock_coriolis.py` depending on the computation of the Coriolis deflection. The main equations used in these codes are reported in Sect. 2.2.2. If the Coriolis deflection is computed, we divide the shock in the shock cap and the shock tail else the overall shock is computed.

We first compute the stagnation point of the radiative shock and use Eq. 2.5 to compute the x values of the discontinuity surface. If the stagnation point is located below the surface of the star 1, the skewing angle is zero and we artificially put the stagnation point equal to the radius of the star 1.

For each value of y, we compute the width of the shock on each side of the discontinuity surface. To do so, we first compute the values of the post-shock temperatures, density and velocities at the shock surface for a set of widths and define the width of the shock as the value of $l_{\rm i}$ solving Eq. 2.9. We then retrieve the velocity of the wind of each star at the position of the shock surface and compute its tangential and normal component. From the normal component of the velocity, we then compute the evolution of the temperature, volume density and surface density on each side of the shock (Eq. 2.10, 2.12, 2.13 and 2.16). We also compute the volume of each shock cell.

If the opening angle is very low (leading to a shock which is below the surface of the star 1), we impose the discontinuity surface to follow the surface of the star 1. The width of the shock as well as the temperature and density inside the shock on the side of star 1 are zero.

Assuming a symmetry around the line of centers, we then compute the 3D positions of the shock associating the corresponding characteristics (temperature, density,...).

If requested, we computed the skewing and deflection on the shock positions and on the tangential velocities of the particles. Once again, if the shock is now below the surface of the star 1, we impose the shock to follow the surface of the star.

We finally save the positions, velocity components, temperatures, volume densities, surface densities and volume values to reuse them during the ray-tracing.

If the wind force balance is too large, this may leads to a crashing of the wind shock onto the surface of the star having the less powerful winds. In this case, the line profile is computed but an extension `_crashing` is added to the name of the resulting file. These line profiles can not be considered as reliable since other physical phenomena must be taken into account in this case.

4.8 The Coriolis deflection

We use the Parkin & Pittard (2008) method described in Sect. 2.3 to compute the Coriolis deflection. The shock is divided in the shock cap and the shock tail. The shock cap is skewed by the angle defined in Eq. 2.17 while the shock tail is skewed and deflected leading to a shock curved around the star 1. The Coriolis deflection is computed over 20 positions of the binary with a step in phase of 0.006π using ballistic motion.

We first study the characteristics of the shock cap in 2D with the adiabatic or radiative model. We then apply the skewing angle s on the positions of the shock cap with:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos(s) & 0 & -\sin(s) \\ 0 & 1 & 0 \\ \sin(s) & 0 & \cos(s) \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{4.4}$$

This changes the position x, y, z of the cells inside the shock but also the direction of the three components of the velocity ($v_{\rm x}$, $v_{\rm y}$, $v_{\rm z}$).

To compute the shape of the shock tail, we first retrieve the characteristics of the shock prevailing at the end of the shock cap. For each of the 20 past positions of the binary, we first compute the time $T_1 = (E - e\sin(E))P/2\pi$ with E the corresponding eccentric anomaly and the distance d between the stars at the corresponding phase ϕ_1. We then compute the rectilinear motion of the particles of the two shock surfaces and discontinuity surface using the tangential velocity that they have at the end of the shock cap and the time difference $T_1 - T_2$ (with T_2 the time of the consecutive position corresponding to phase ϕ_2). The coordinates of the particles inside the shock at each of the 20 positions of the binary are then linearly distributed between the shock surface and the discontinuity surface. We then compute the center of mass of the system as $CM = d\,M_2/(M_1 + M_2)$. We finally rotate the particles from the oldest to the closest position with:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos(\phi_2 - \phi_1) & 0 & -\sin(\phi_2 - \phi_1) \\ 0 & 1 & 0 \\ \sin(\phi_2 - \phi_1) & 0 & \cos(\phi_2 - \phi_1) \end{pmatrix} \begin{pmatrix} x - CM \\ y \\ z \end{pmatrix} + \begin{pmatrix} CM \\ 0 \\ 0 \end{pmatrix} \tag{4.5}$$

At each loop l, we rotate the particles ejected at the positions occurring before position l. Hence, at the end of the 20 loops, the oldest position was rotated 20 times while the closest position has rotated only once.

4.9 Plots

The characteristics of the shock are represented in several plots saved in the directory `plots`. We plot the temperatures inside the shock along two planes (xy and xz) for the Coriolis deflection and along the xy plane without Coriolis deflection. These plots are named `temperature_[xy/xz]_par[ipar].pdf` with *ipar* the line number of the stellar parameter file `param` (beginning at 0). We also plot the volume density in xy in the file `density_par[ipar].pdf`. The densities and temperatures at each position are also saved in the tabulated file `char_shock_par[ipar].dat`.

4.10 Following the photons towards the observer

This part of the code is written in the file `raytracing.py`. It is not useful to compute several times the overall shock characteristics if the user only wishes to compute the line profile for different lines but for the same characteristics of the binary system. We thus decided to save the characteristics of the positions crossed by the photons towards the observer in order to use them afterwards to compute the line profile for different lines.

We first compute the 3D trees for the position of the discontinuity surface, the position of the two shock surfaces, and the positions of the cells inside the shock. A nD tree is a space-partitioning data structure organizing large data sets in nD space to improve the search of, for example, the nearest neighbor. We also compute the 2D trees for the positions at which we computed the distribution of the stellar winds.

We then define the observer's coordinate system with the xt-axis pointing towards the observer (see Fig. 3.1):

$$\begin{pmatrix} xt \\ yt \\ zt \end{pmatrix} = \begin{pmatrix} \sin\phi\sin i & \sin\phi\cos i & \cos\phi \\ \cos\phi\sin i & \cos\phi\cos i & -\sin\phi \\ -\cos i & \sin i & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{4.6}$$

For each position in the shock, we then verify the visibility of the emitting cell by the observer. The cell is not visible by the observer if $yt < 0$ and $zt^2 + xt^2 < R^2$ with R the radius of the star.

If the cell is visible, we can compute the emission and the trajectory of the emitted photon. The emission of the cell depends on its temperature and the cooling function of the studied line. We thus save the temperature of the cell and the factor multiplying the emissivity of the line, i.e., $n_{\rm e}n_{\rm H}V$. We then compute the radial velocity of the photon as $v_{\rm rad} = -v_{\rm x}\cos\phi\sin i - v_{\rm y}\cos i + v_{\rm z}\sin i\sin\phi$ in order to have a positive radial velocity

when the emitting material is receding from the observer. We thus know in which bin of the line profile the emission of this cell will account for.

We now move the photon forward. We define the step Δ of the photon towards the observer as a tenth of the maximum width of the shock. The step along each axis in the star's coordinate system (Δx, Δy and Δz) are thus the projections of the step along each axis according to the phase ϕ and inclination i of the orbit:

$$\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \Delta \begin{pmatrix} \cos\phi \sin i \\ \cos i \\ -\sin\phi \sin i \end{pmatrix} \tag{4.7}$$

For each step Δ until the photon is outside the simulation box, we first determine on which side of the discontinuity surface the photon is. This is done be retrieving the nearest position of the overall shock cells. If the nearest cell is in the side of star 1, the characteristics of this star will be used in the following computations. We then checked whether the photon is still inside the shock by comparing the distance between the photon and the closest point of the discontinuity surface and the width of the shock corresponding to this closest point. If the distance between the photon and the discontinuity surface is smaller than the width, the photon is still inside the shock.

If the photon is inside the shock, we retrieve the temperature associated with the closest cell. This temperature will be then used to compute the cross-section *cs* and then the optical depth $\tau_{\rm shock}$. The optical depth is computed as in Eq. 3.2. In this equation, only *cs* depends on the energy of the studied line. We thus save the temperature of the cell crossed by the photon (needed to compute *cs*) and the factor $\sigma/(1.3\, m_{\rm p}\, \vec{P}\cdot\vec{N})$ (further referenced as the multiplying factor F). These two parameters are then used to compute the absorption of any line by the so constructed shock (Sect. 4.11).

Once the photons have left the shock region, they will suffer absorption by the wind of the star. We thus compute the temperature of the wind as $T_{\rm wind} = 0.4T_{\rm eff} + 0.6T_{\rm eff}(R/r)^{1.9}$ with r the distance between the photon and the star of the corresponding side of the shock. We also retrieve the velocity of the wind at the position of the photon. The optical depth is computed with Eq. 3.3. As for the optical depth inside the shock, only *cs* depends on the energy of the studied line. We thus save $T_{\rm wind}$ and the value of the optical depth for $\tau_0 = \dot{M}_{\rm i}/(4\pi\, v_{\rm i}\, R_{\rm i})$ (further referenced as the multiplying factor F) to compute the absorption of any line (Sect. 4.11).

We need to determine whether the photon will cross again the shock before reaching the observer. To do so, we compute the minimum distances between each further step of the photon and the shock surface. The minimum distance will first increase as the photon moves away from the shock. If the photon will cross the shock again, it then gets closer to the shock surface. The minimum distance will thus decrease until the photon crosses the surface and then the distance will increase again. If a minimum value is present in the evolution of the minimum distance, this means that the photon will enter inside the shock. We thus move the photon until it enters the shock and compute the optical depth though the wind. The photon then continues its way until it leaves the shock and never enters it again.

The diagram of the ray-tracing process and the needed parameters is shown in Fig. 4.1.

The different parameters needed for the computation of the line profiles are saved is several files recorded in the directory `histograms`. We first create three files whose number of lines is equal to the number of bins in the line profile. The first file named `temp_emiss_par[ipar].data` contains the temperatures of the visible cells with *ipar* the line number of the stellar parameter file `param` (beginning at 0). Each line lists the space-separated temperatures (in keV) involved in the emission at the corresponding radial velocity (and thus bin of the line profile). The file `emiss_part_shock_par[ipar].data` has the same structure as the temperature file but lists the multiplying factor of the emission which does not depend on the temperature. The file `bin_par[ipar][_crashing].data` is a one column file listing the radial velocity (in km/s) corresponding to each bin.

Then, for each bin K of the line profile, we write a file named `ray_tracing_par[ipar]_bin[K].data`[4]. This file lists the temperature (T) and the multiplying factor (F) involved in the calculation of the optical depth. The temperature and the multiplying factor are listed side by side, i.e., on one line we write T_1 F_1 T_2 F_2 ... T_n F_n. Each line corresponds to one cell temperature at the line K of the file `temp_emiss_par[ipar].data`.

[4]For the already computed shocks of the binary systems listed in `stellar_params_set.txt`, the line profiles where computed for different couples of (inclination, phase) for each binary. The names of the files are thus distinguished by `ray_tracing_par[ipar]_bin[K]_incl[incl]_phase[phase].data`.

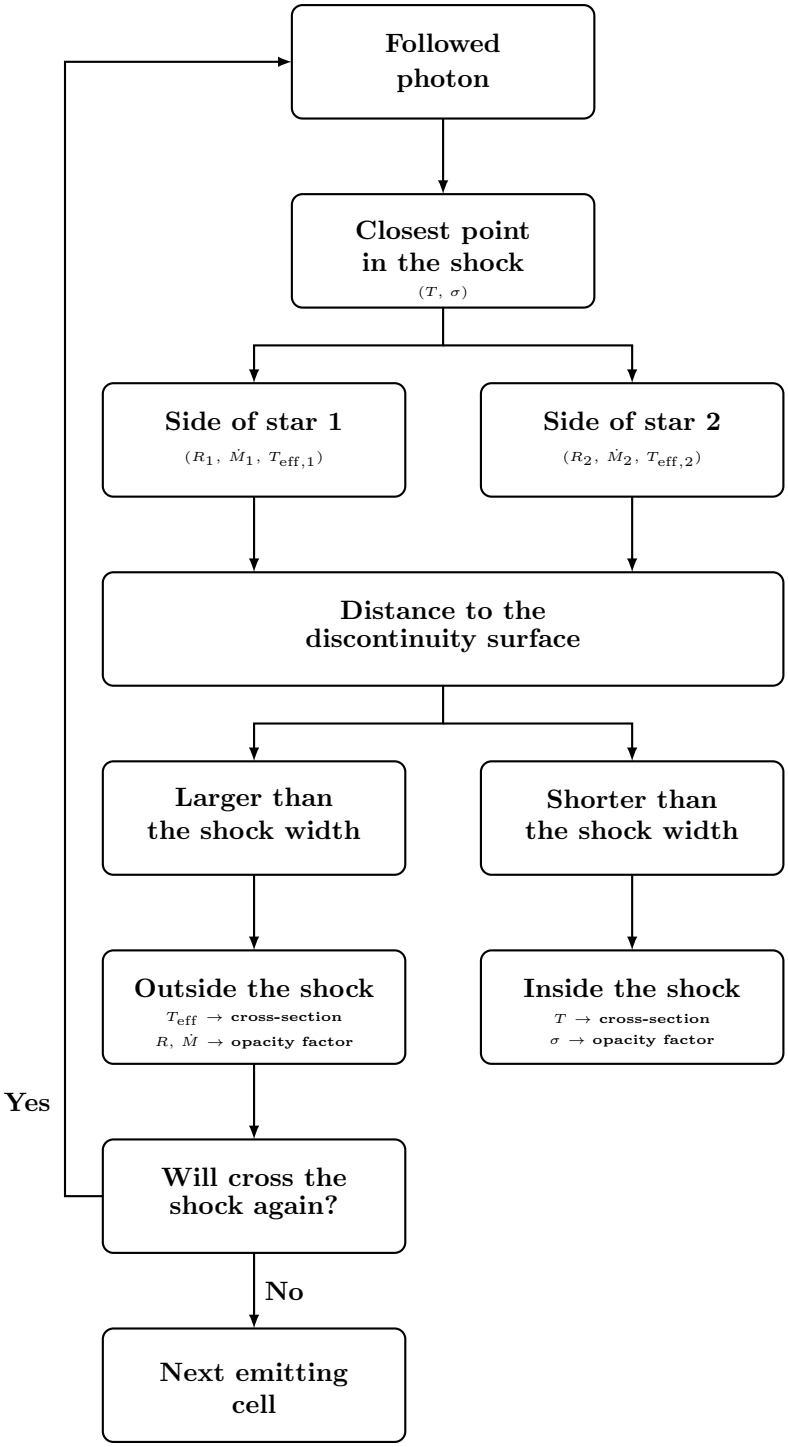


Figure 4.1: The ray-tracing code.

The number of lines in the file `ray_tracing_par[ipar]_bin[K].data` is thus the same as the number of the cell temperatures at the line K.

This part of the code is a bit long: about five hours for one core on a computer with Intel Pentium CPU 3550M @ 2.30GHz x 2 and 7.7 Go of RAM.

4.11 Computation of the line profile

The line profile is computed by the file `compute_profile.py`.

The code computes the emission and absorption in each bin of the profile. It thus reads the lines of the files `emiss_part_shock_par[ipar].data`, `temp_emiss_par[ipar].data`, and `bin_par[ipar][_crashing].data` one by one. For each line K, it saves the cell temperatures and multiplying factors (F) in a vector. For each cell temperature, it retrieves the emissivity q for the studied line. The code also computes, for each cell temperature, the total optical depth τ experienced by the photon. To do this, it reads the corresponding line of the `ray_tracing_par[ipar]_bin[K].data` file containing the temperature and the multiplying factor of the optical depths. The cross-section is computed for each temperature and then multiplied by the corresponding factor. The so constructed optical depths are then summed to retrieve the total optical depth τ. The total emissivity in this bin is thus the sum of all the $10^{-q}\,F\,e^{-\tau}$ of the line K.

The line profile is saved in a text file named `line_profile_par[ipar][_crashing].data` and in a pdf file named `line_profile_par[ipar][_crashing].pdf` in the directory `[atom]_[ion]`. If mode=3, *ipar* is the index of the binary whose the parameters are the closest to the binary defined by the user. Moreover, the values of the inclination and phase are also recorded in the name of the files (see Sect. 4.12).

This line profile can then be convolved with the instrumental response of the observing instrument in order to be compared to observations. This operation can be made separately from the main code by launching the file `observed_line_profile.py`. This code first asks the path and the name of the line profile file to convolve. Then, the user should enter the path and the names to the response matrix (RMF) and ancillary response (ARF) files. Finally, if known, the user can enter the distance to the binary in kpc. If the distance is unknown, a typical distance of 1.5 kpc is assumed. The convolution can also be computed in the main code if the parameter `convolve` is set to `yes`. In this case, the directory and the names of the RMF and ARF, and, optionally, the distance to the binary must also be given as parameter of the main code. The main code runs the file `observed_line_profile_main.py`.

At the date of the writing of this document, Athena has not been launched yet. The newest version of the theoretical RMF and ARF are thus given in the LIFELINE package for the user willing to obtain the line profile as it would be observed by Athena. They are computed for the X-IFU instrument assuming an active mirror aperture radius of $259 - 1183$ mm, a 2.3 mm rib spacing, a on-axis observation and a thick filter. The RMF is XIFU_cc_baselineconf_thickfilter_2018_10_10.rmf and the ARF is XIFU_cc_baselineconf_thickfilter_2018_10_10.arf.

The luminosity in each energy bin E of the line profile (computed in $10^{27}\,\mathrm{erg\,s^{-1}}$) is first converted to a flux $F(E)$ expressed in $\mathrm{erg\,s^{-1}\,cm^{-2}}$. The flux at energy E is then convolved thanks to:

$$LP_{\mathrm{conv}}(E) = \int_0^\infty R(E,E')\,A(E')\,F(E')\,dE'\,, \tag{4.8}$$

with $R(E,E')$ the probability that a photon of energy E' is recorded with an energy E, and $A(E')$ the effective area at the energy E' (in cm^2). Finally, the convolved line profile ($LP_{\mathrm{conv}}(E)$) is expressed in $\mathrm{count\,s^{-1}}$ using:

$$S(E) = 6.242\,10^8\,\frac{LP_{\mathrm{conv}}(E)}{E} \tag{4.9}$$

to represent the observed spectrum. The number of counts that should be observed during 10 ks is also computed.

The convolved line profile is saved in a text file named `[name_of_the_line_profile_file]_convolved.data` and in a pdf file named `[name_of_the_line_profile_file]_convolved.pdf` in the directory `[atom]_[ion]`.

The user can also directly retrieve the line profiles already computed without running the code. To do so, the user needs to run the `find_LP.py` code to determine the line profile created for the binary system whose the parameters and view angle are the closest to those defined by the user in their stellar parameter file. The code must be launched from the directory where the stellar parameter file, the `stellar_params_set.txt` file and the `[atom]_[ion]_set` directory are located. The code is launched as

`python find_LP.py param`

with `param` the stellar parameter file defined by the user. The code first asks the ion to retrieve the line profile. For each user's binary, the code then determines the binary system whose the parameters are the closest to

Table 4.1: Table of parameters of the computed systems.

Spectral type	$T_{\rm eff}$ (K)	R $(R_\odot)$	M $(M_\odot)$	$\log(\dot{M})$ $(M_\odot\, yr^{-1})$
O3V	44616	13.84	58.34	-5.641
O5V	41540	11.08	37.28	-5.969
O7V	35531	9.37	26.52	-7.340
O9V	31524	7.73	18.03	-7.818
O3III	42942	16.57	58.62	-5.445
O5III	39507	15.26	41.48	-5.63
O7III	34638	14.51	31.17	-6.804
O9III	30737	13.69	23.07	-6.812
O3I	42551	18.47	66.89	-5.347
O5I	38520	19.48	50.87	-5.561
O7I	33326	21.14	40.91	-5.995
O9I	29569	22.6	31.95	-6.385

those of the user's binary. Then, for this binary parameters, the code finds the inclination and phase which are the closest to those of the user. Finally, the code lists the name of line profile files satisfying these conditions. This list is also saved in the file `results_find_LP.txt`.

4.12 The grid of stellar parameters

We have already computed the wind distribution, the shock characteristics and the iron 26 line profiles of a large set of O-type binary systems. The parameters of these systems are recorded in the file `stellar_params_set.txt`. We considered every combination of two stars characterized by the parameters reported in Table 4.1. We adopted the temperatures, masses and radii from Table 1–3 of Martins et al. (2005) and the mass-loss rates from Muijres et al. (2012). For each couple of stars, we considered ten distances separating them up to $2000 R_\odot$. The six shortest distances are chosen to create a radiative shock while the four largest distances lead to an adiabatic shock. This leads to 780 configurations with the star 1 having the less powerful wind. The `stellar_params_set.txt` file also reports the mode at which the main code was launched. Since mode=3 only computes the line profile without any computation of the winds and the shock, only systems computed in mode=1 or mode=2 are reported in this file. We performed the computations with the ATOMDB version 3.0.9 [5] for a mono-atomic fully ionized gas ($\mu = 0.62$) with the fractional mass abundance of H equal to that of the Sun (0.7381).

The overall computation was performed (mode=1) for five different phases (from 0 to 2π) and five inclinations (from 18 to 90°). We considered only an eccentricity equal to zero since a larger eccentricity leads to a change of the distance between the stars at different phases but 10 different distances were already taken into account in the binary set. We first computed the distribution of the stellar winds which are saved in HDF5 files and represented in pdf files in the directory `winds_set`. For each binary system *ipar* (beginning at 0), the wind of the star i is named `winds_star[i]_param[ipar].h5/pdf`. We then computed the shock characteristics and saved the plots in `plots_set`. The files written for the computation of the ray-tracing are saved in `histograms_set` and their names also list the rounded value of inclination in degree and the decimal value of the phase (e.g., for an inclination of 15.3° and a phase of 0.35, the name of the ray-tracing file will be `ray_tracing_par[ipar]_bin[K]_incl15_phase35.data`). As explained in Sect. 4.3, the code defines the phase $\phi = 0$ to the conjunction. The values of the phases used in the naming thus follow this convention. The line profiles of the iron XXV helium like triplet and the iron XXVI hydrogen-like ion were finally computed for each binary system and each inclination and phase. The profiles are saved in the directories `Fe_XXV_set` and `Fe_XXVI_set` and their names also list the value of inclination and phase.

4.13 Update the LIFELINE database

Since LIFELINE is collaborative, the user is kindly asked to update the database (i.e., the `*_set` directories on GitHub) with their resulting files. First, the user has to update the files and directory on his local repository (i.e., on his/her computer). If mode=1 or mode=2, the user should add their system parameters to the parameter file `stellar_params_set.txt`. The *ipar* of their resulting file names must thus be changed to correspond to the line in the `stellar_params_set.txt` file to be saved in the data set directories. The user can

[5] If you use an other version of ATOMDB, please, change the filemap and the apec_linelist.fits file.

then save the resulting graphs and files (saved in the directories `winds`, `histograms`, `plots` and `[atom]_[ion]`) in the corresponding set directories (i.e., `winds_set`, `histograms_set`, `plots_set` and `[atom]_[ion]_set`). Do not forget to rename the files from the `histograms` and `[atom]_[ion]` directories to report the inclination and phase used with the phase $\phi = 0$ at the time of the conjunction.

We have written a `Python` code to automatically rename and copy the files to the corresponding directories. The user can thus run `python rename_files.py`. This code asks for the mode under which the main code was run, the complete path containing the `winds[_set]`, `histograms[_set]`, `plots[_set]` and `[atom]_[ion][_set]` directories, the name of the [atom]_[ion] directory, the name of your stellar parameter file, and the number of the binary system from your stellar parameter file to include in the database. This last parameter corresponds to the number of the line in the stellar parameter file. It begins at 0 and can be a list of space-separated values (e.g., 0 1 3 5 6). The code first reads the stellar parameter file to record the system parameters. It then reads the `stellar_params_set.txt` file to determine the new number of the binary system. If mode=3, it searches the number of the binary system which was used to compute the line profiles, otherwise, the binary system will be a new entry to the `stellar_params_set.txt` file. The code then automatically renames the resulting files and copies them to the corresponding directory [6].

[6] As explained in Sect. 4.3, the code defines the phase $\phi = 0$ to the conjunction. The values of the phases used in the naming thus follow this convention implying that the value of the phase given in the stellar parameter may differ from those of the renaming for eccentric orbit.

Bibliography

Abbott, D. C. 1980, ApJ, 242, 1183

Abbott, D. C. 1982, ApJ, 259, 282

Antokhin, I. I., Owocki, S. P., & Brown, J. C. 2004, ApJ, 611, 434

Canto, J., Raga, A. C., & Wilkin, F. P. 1996, ApJ, 469, 729

Castor, J. I., Abbott, D. C., & Klein, R. I. 1975, ApJ, 195, 157

Foster, A. R., Ji, L., Smith, R. K., & Brickhouse, N. S. 2012, ApJ, 756, 128

Grevesse, N., Asplund, M., Sauval, A. J., & Scott, P. 2010, Astrophysics and Space Science, 328, 179

Kudritzki, R.-P. & Puls, J. 2000, ARA&A, 38, 613

Lykins, M. L., Ferland, G. J., Porter, R. L., et al. 2013, MNRAS, 429, 3133

Martins, F., Schaerer, D., & Hillier, D. J. 2005, A&A, 436, 1049

Muijres, L. E., Vink, J. S., de Koter, A., Müller, P. E., & Langer, N. 2012, A&A, 537, A37

Nazé, Y., Rauw, G., Vreux, J.-M., & De Becker, M. 2004, A&A, 417, 667

Parkin, E. R. & Pittard, J. M. 2008, MNRAS, 388, 1047

Rauw, G. 2007, PhD thesis, University of Liège

Stevens, I. R. 1988, MNRAS, 235, 523

Stevens, I. R. 1991, ApJ, 379, 310

Stevens, I. R., Blondin, J. M., & Pollock, A. M. T. 1992, ApJ, 386, 265

Stevens, I. R. & Pollock, A. M. T. 1994, MNRAS, 269, 226