NUMERICAL METHODS (1D, ...) FOR NON-LTE RADIATIVE TRANSFER. APPLICATIONS TO SOLAR PROMINENCES, FILAMENTS AND SEMI-INFINITE ATMOSPHERES

Martine Chane-Yook (IAS)
martine.chane-yook@ias.u-psud.fr

Reviewed by :

Frédéric Paletou (IRAP)
frederic.paletou@univ-tlse3.fr

April 2020

Contents

I Non-LTE radiative transfer in 1D 5

1 Case of a two-level atom 7

1.1 Semi-infinite atmosphere . . . 7

1.1.1 Semi-infinite atmosphere modeling . . . 8

1.1.2 Discretization of radiative transfer equation and boundary conditions 9

1.1.3 Λ-iteration method . . . 15

1.1.4 ALI method (Accelerated Lambda Iteration) . . . 17

1.1.5 Acceleration of ALI scheme by Ng method . . . 21

1.1.6 Description of numerical code . . . 24

1.1.7 Description of subroutines . . . 25

1.1.7.1 Set of variables used in the program . . . 25

1.1.7.2 Subroutine description in each module . . . 27

1.1.8 Running numerical program . . . 29

1.2 Solar filament . . . 29

1.2.1 Filament modeling . . . 30

1.2.2 Two-level atom and statistical equilibrium . . . 30

1.2.3 Implementation . . . 36

1.2.3.1 Construction of the atmosphere and boundary conditions . 36

1.2.3.2 Method for solving Non-LTE RTE using accelerated ALI scheme . . . 39

1.2.4 Description of numerical code . . . 40

1.2.5 Description of subroutines . . . 41

1.2.5.1 Set of variables used in the program . . . 41

1.2.5.2 Description of subroutines in each module . . . 45

1.2.6 Running numerical program . . . 48

1.3 Solar prominence . . . 48

1.3.1 Prominence modeling . . . 48

1.3.2 Accelerated ALI method and boundary conditions . . . 49

1.3.2.1 Boundary conditions . . . 49

1.3.2.2 Method for solving Non-LTE RTE using accelerated ALI scheme . . . 51
1.3.3 Description of numerical code . . . 52
1.3.4 Description of subroutines . . . 53
1.3.4.1 Set of variables used in the program . . . 53
1.3.4.2 Description of subroutines in each module . . . 56
1.3.5 Running numerical program . . . 57
1.4 Conclusion . . . 58

2 Case of a multilevel atom **59**
2.1 Multilevel formulation . . . 59
2.2 Iterative method and operator choice . . . 64
2.2.1 MALI method . . . 64
2.2.2 Choice of approximate lambda operator $\Lambda^*_{\mu\nu}$ . . . 64
2.3 Local operator with no background continuum (no continuum absorption). Application to a semi-infinite atmosphere . . . 65
2.3.1 Statistical equilibrium equations . . . 65
2.3.2 Description of H3CRD program . . . 68
2.3.3 Algorithm . . . 69
2.3.4 Atomic structure of hydrogen . . . 69
2.3.5 Description of subroutines . . . 71
2.3.5.1 Set of variables used in module "param_mod" (param.f90 file) 72
2.3.5.2 Description of subroutines in module "general_mod" (general.f90 file) . . . 73
2.3.5.3 Description of subroutines in module "mali_mod" (mali.f90 file) . . . 74
2.3.6 Running H3CRD program . . . 77
2.4 Local operator with background continuum (with continuum absorption). Application to a semi-infinite atmosphere) : planned . . . 78

II Non-LTE radiative transfer in 2D : planned **79**

Acknowledgements **81**

Bibliography **84**

Part I

Non-LTE radiative transfer in 1D

1 Case of a two-level atom

Contents

1.1 Semi-infinite atmosphere 7
1.2 Solar filament 29
1.3 Solar prominence 48
1.4 Conclusion 58

We aim to solve 1D non-LTE radiative transfer equation (RTE) for a two-level atom in the case of a solar : semi-infinite atmosphere, filament and prominence. In paragraph 1.1 (semi-infinite atmosphere), we discretize transfer equation in order to write it in a matrix form. Λ-iteration method as well as Accelerated Lambda Iteration (ALI) method are explained and applied to solve transfer equation in the case of a semi-infinite atmosphere. In paragraph 1.2 (filament), we apply ALI method to solve transfer equation for a filament and for a realistic atom (hydrogen, Lyman Alpha line $L\alpha$). In paragraph 1.3, ALI method is applied to a prominence under the same conditions as a filament.

1.1 Semi-infinite atmosphere

The semi-infinite atmosphere considered here concerns the photosphere and the chromosphere. It is represented by a layer of thickness τ, divided in N_d sublayers. Thickness of each atmosphere sublayer τ_d $(d = 1, ..., N_d)$ is in cm. The layers are spherical but they are considered as plane-parallel slabs. In the case of a two-level atom (a fundamental and a line), there is only one line and therefore only one grid of reduced frequencies x_i, $i = 1, ..., N_{freq}$. Complete frequency redistribution (CRD) is used here. We consider Voigt profile for line. Here, $B = 1$ (Planck function), $\epsilon = 10^{-4}$ (extinction coefficient or probability of collisional destruction of photons). The boundary conditions (BC) are :
- BC (top) : 0 (there is no incident intensity),
- BC (bottom) : $B = 1$ in the whole layer.

We consider 2 grids :
- a grid of optical depths $\tau = (\tau_d)$, $d = 1, ..., N_d$,

- a reduced frequency grid $\left(x_i = \dfrac{\nu_i - \nu_0}{\Delta\nu_D}\right)$, $i = 1, ..., N_{freq}$. ν_i is the line frequency, ν_0 is the line center frequency and $\Delta\nu_D = \dfrac{\lambda}{c}\left(\dfrac{2kT}{m}\right)^{1/2}$ is the Doppler width. λ is the line wavelength, c is the velocity of light, k is the Boltzmann constant, T is the atmospheric temperature and m is the mass of the considered atom.

1.1.1 Semi-infinite atmosphere modeling

Figure 1.1 illustrates this modeling. τ is ranged by increasing value (dimensionless number).

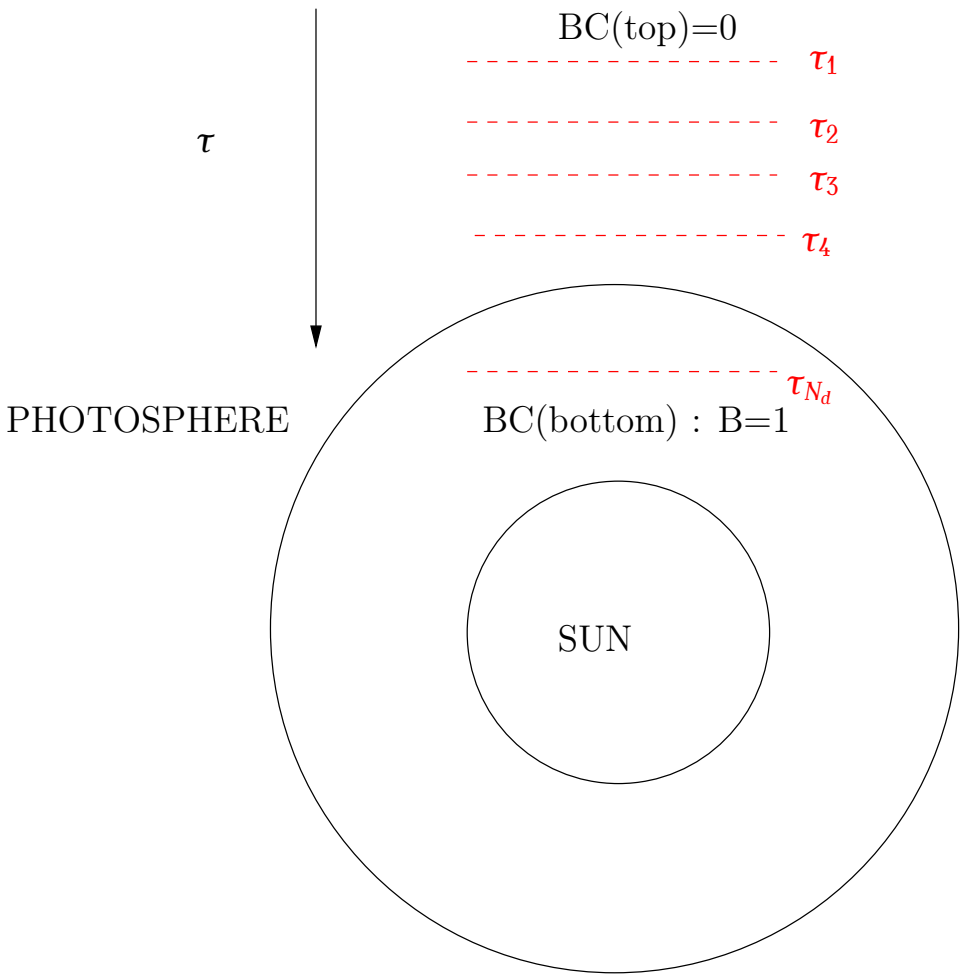

Figure 1.1: Solar semi-infinite atmosphere model (figure is not to scale).

In the next sections, we solve numerically radiative transfer equation (RTE) for a semi-infinite atmosphere. We present two methods : Λ-iteration method (convergence of the solution is slow) and ALI method (Accelerated Lambda Iteration) based on the previous method (we

accelerate Λ-iteration method). We first discretize radiative transfer equation in order to write it in matrix form.

1.1.2 Discretization of radiative transfer equation and boundary conditions

The radiative transfer equation is ((2.22) from Jefferies (1968)) :

$$\mu \frac{dI_{\mu\nu}}{d\tau_\nu} = I_{\mu\nu} - S, \tag{1.1.1}$$

where $I_{\mu\nu} \equiv I(\mu, \nu, \tau)$ is the specific intensity, $\mu = \cos\theta$, θ is the angle between the light ray and the normal to the solar surface ($\mu > 0$). ν is the frequency, τ is the generic optical depth (in center of the line), τ_ν is the optical depth at a given frequency ν, S is the source function. In CRD, S depends only on τ.
The source function for a two-level atom is :

$$S(\tau) = (1 - \epsilon)\, \overline{J}(\tau) + \epsilon\, B(\tau), \tag{1.1.2}$$

where B is the Planck function, ϵ is the extinction coefficient (or probability of collisional destruction of photons).

$$\overline{J}(\tau) = \int_0^{+\infty} J_\nu\, \phi_\nu\, d\nu, \tag{1.1.3}$$

where ϕ_ν is the line profile (Voigt profile).
The mean intensity (on directions) is :

$$J_\nu(\tau) = \frac{1}{4\pi} \oint I_{\mu\nu}\, d\Omega = \int_{-1}^{1} I_{\mu\nu}\, d\mu \tag{1.1.4}$$

Remark 1.1.1 *$\overline{J}$, J_ν, S and $I_{\mu\nu}$ have same unit.*

τ_ν and τ are linked by the following formula :

$$\tau_\nu = \phi_\nu\, \tau \tag{1.1.5}$$

Remark 1.1.2 *By applying the change of variable τ_ν by τ (1.1.5), the RTE (1.1.1) is (formula 3 from Olson et al. (1986)) :*

$$\mu \frac{dI_{\mu\nu}}{d\tau} = \phi_\nu\, [I_{\mu\nu} - S] \tag{1.1.6}$$

Equation (1.1.1) can be written as :

$$\mu^2 \frac{d^2 u}{d\tau_\nu^2} = u - S, \tag{1.1.7}$$

where $u = \frac{1}{2}(I_{\mu\nu} + I_{-\mu\nu})$.

Proof of (1.1.7):

Let's consider :

$$u = \frac{1}{2}(I_{\mu\nu} + I_{-\mu\nu}) \tag{1.1.8}$$

and

$$v = \frac{1}{2}(I_{\mu\nu} - I_{-\mu\nu}) \tag{1.1.9}$$

According to (1.1.1) :

$$\mu \frac{dI_{\mu\nu}}{d\tau_\nu} = I_{\mu\nu} - S,$$

so that

$$-\mu \frac{dI_{-\mu\nu}}{d\tau_\nu} = I_{-\mu\nu} - S \tag{1.1.10}$$

Summing equations (1.1.1) and (1.1.10), we obtain :

$$\mu \frac{dv}{d\tau_\nu} = u - S \tag{1.1.11}$$

Subtracting (1.1.1) from (1.1.10), we obtain :

$$\mu \frac{du}{d\tau_\nu} = v. \tag{1.1.12}$$

Therefore $\frac{dv}{d\tau_\nu} = \mu \frac{d^2u}{d\tau_\nu^2}$. Subtituting (1.1.12) into (1.1.11), we obtain (1.1.7). ■

We discretize equation (1.1.7) : $\mu^2 \frac{d^2u}{d\tau_\nu^2} = u - S$
$\forall \nu, \forall \mu, \forall d = 2, 3, ..., N_{d-1}$, let $\Delta\tau_{\nu,d} = (\tau_{d+1} - \tau_d)$ $\phi_\nu := \tau_{\nu,d+1} - \tau_{\nu,d}$.

The following discretization scheme is considered :

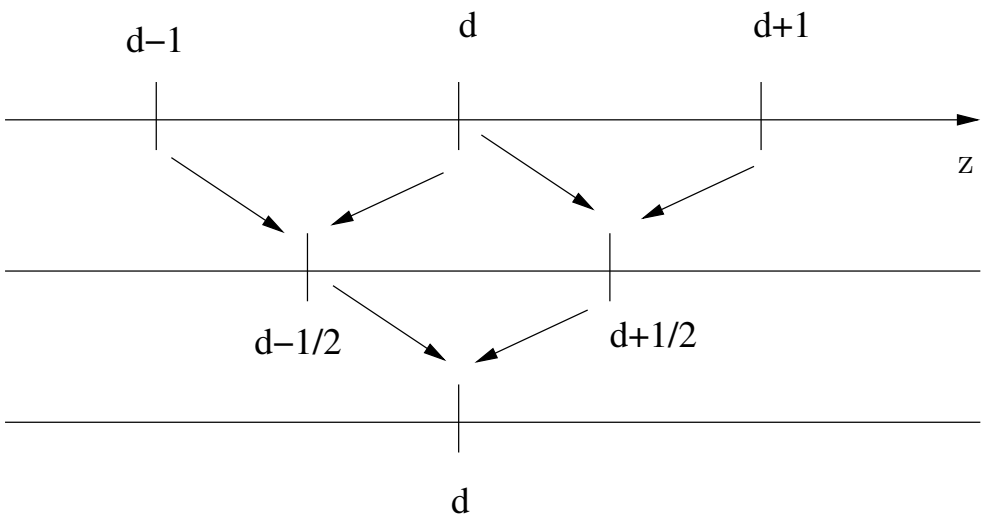

Figure 1.2: Discretization scheme of equation (1.1.7).

The derivative of u with respect to τ_ν at point $d-1/2$ is :

$$\left[\frac{du}{d\tau_\nu}\right]_{d-1/2} = \frac{u_d - u_{d-1}}{(\tau_d - \tau_{d-1})\phi_\nu} = \frac{u_d - u_{d-1}}{\Delta\tau_{\nu,d-1}}$$

Similarly, the derivative of u with respect to τ_ν at point $d+1/2$ is :

$$\left[\frac{du}{d\tau_\nu}\right]_{d+1/2} = \frac{u_{d+1} - u_d}{\Delta\tau_{\nu,d}}$$

And the second derivative of u with respect to τ_ν at point d is :

$$\left[\frac{d^2u}{d\tau_\nu^2}\right]_d = \frac{\left[\frac{du}{d\tau_\nu}\right]_{d+1/2} - \left[\frac{du}{d\tau_\nu}\right]_{d-1/2}}{(\tau_{d+1/2} - \tau_{d-1/2})\phi_\nu}$$

Let

$$\begin{aligned}(\tau_{d+1/2} - \tau_{d-1/2})\phi_\nu &= \frac{1}{2}(\tau_{d+1} - \tau_{d-1})\phi_\nu \\ &= \frac{1}{2}(\tau_{d+1} - \tau_d + \tau_d - \tau_{d-1})\phi_\nu \\ &= \frac{1}{2}(\Delta\tau_{\nu,d} + \Delta\tau_{\nu,d-1})\end{aligned}$$

Hence

$$\left[\frac{d^2u}{d\tau_\nu^2}\right]_d = 2\,\frac{\frac{u_{d+1}-u_d}{\Delta\tau_{\nu,d}} - \frac{u_d-u_{d-1}}{\Delta\tau_{\nu,d-1}}}{\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1}}$$

$$= \frac{2\,u_{d+1}}{\Delta\tau_{\nu,d}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})} - \frac{2\,u_d}{\Delta\tau_{\nu,d}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})}$$

$$- \frac{2\,u_d}{\Delta\tau_{\nu,d-1}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})} + \frac{2\,u_{d-1}}{\Delta\tau_{\nu,d-1}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})}$$

$$= \frac{2\,u_{d-1}}{\Delta\tau_{\nu,d-1}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})} + \frac{2\,u_{d+1}}{\Delta\tau_{\nu,d}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})}$$

$$-\,2\,u_d\left(\frac{1}{\Delta\tau_{\nu,d}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})} + \frac{1}{\Delta\tau_{\nu,d-1}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})}\right).$$

Let

$$\frac{1}{\Delta\tau_{\nu,d}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})} + \frac{1}{\Delta\tau_{\nu,d-1}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})} = \frac{1}{\Delta\tau_{\nu,d}\,\Delta\tau_{\nu,d-1}}$$

Then,

$$\left[\frac{d^2u}{d\tau_\nu^2}\right]_d = \frac{2\,u_{d-1}}{\Delta\tau_{\nu,d-1}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})} - \frac{2\,u_d}{\Delta\tau_{\nu,d}\,\Delta\tau_{\nu,d-1}} + \frac{2\,u_{d+1}}{\Delta\tau_{\nu,d}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})}$$

The discretization of RTE (1.1.7) leads to :

$$\frac{2\,\mu^2\,u_{d-1}}{\Delta\tau_{\nu,d-1}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})} - \frac{2\,\mu^2\,u_d}{\Delta\tau_{\nu,d}\,\Delta\tau_{\nu,d-1}} + \frac{2\,\mu^2\,u_{d+1}}{\Delta\tau_{\nu,d}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})} = u_d - S_d,$$

$\forall\nu, \forall\mu, \forall d = 2, \ldots, N_{d-1}$

$$\Longleftrightarrow S_d =$$

$$\frac{-2\,\mu^2\,u_{d-1}}{\Delta\tau_{\nu,d-1}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})} + \left(1 + \frac{2\,\mu^2}{\Delta\tau_{\nu,d}\,\Delta\tau_{\nu,d-1}}\right)u_d - \frac{2\,\mu^2\,u_{d+1}}{\Delta\tau_{\nu,d}(\Delta\tau_{\nu,d}+\Delta\tau_{\nu,d-1})}$$

$$\Longleftrightarrow -A_d.u_{d-1} + B_d.u_d - C_d.u_{d+1} = S_d \quad \forall\mu, \forall\nu,\ \forall 2 \le d \le N_{d-1}, d \in \mathbb{N},$$

with

$$A_d = \frac{2\mu^2}{\Delta\tau_{\nu,d-1}(\Delta\tau_{\nu,d} + \Delta\tau_{\nu,d-1})},\ B_d = 1+\frac{2\mu^2}{\Delta\tau_{\nu,d}\ \Delta\tau_{\nu,d-1}},\ C_d = \frac{2\mu^2}{\Delta\tau_{\nu,d}(\Delta\tau_{\nu,d} + \Delta\tau_{\nu,d-1})},$$

and $\Delta\tau_{\nu,d} = (\tau_{d+1} - \tau_d)\ \phi_\nu$.

Here, $A_d > 0$, $B_d > 0$ and $C_d > 0$.(Appendix A from Rybicki and Hummer (1991)) ■

We write the boundary conditions (BC) for $d = 1$ (coefficients B_1 and C_1) and $d = N_d$ (coefficients A_{N_d} and B_{N_d}) :

let $\Delta\tau_{\nu,1} = (\tau_2 - \tau_1)\phi_\nu$ and $\Delta\tau_{\nu,N_d} = (\tau_{N_d} - \tau_{N_d-1})\phi_\nu$.

We consider

$$\begin{cases} I_\nu^-(\mu) = I_{sup} = 0\ (\equiv I_{-\mu\nu}),\ \text{pour } \tau_\nu = 0,\ \text{upper BC} \\ I_\nu^+(\mu) = I_{inf} = B = 1\ (\equiv I_{\mu\nu}),\ \text{pour } \tau_\nu = \tau_{\nu_{max}},\ \text{lower BC} \end{cases}$$

According to (1.1.8), we have :

$$2u = I_{\mu\nu} + I_{-\mu\nu} \Longleftrightarrow I_{\mu\nu} = 2u - I_{-\mu\nu} \tag{1.1.13}$$

According to (1.1.9), we have :

$$I_{-\mu\nu} = I_{\mu\nu} - 2v \Longleftrightarrow I_{-\mu\nu} = 2u - I_{-\mu\nu} - 2v \Longleftrightarrow I_{-\mu\nu} = u - v. \tag{1.1.14}$$

For $\tau_\nu = 0$, equation (1.1.14) can be written as :

$$u(\tau_\nu = 0, -\mu, \nu) - v(\tau_\nu = 0, -\mu, \nu) = I_{sup} \tag{1.1.15}$$

(1.1.14) $\Longleftrightarrow v = u - I_{-\mu\nu}$, then (1.1.12) $\Longleftrightarrow \mu\ \dfrac{du}{d\tau_\nu} = u - I_{-\mu\nu}$.

For $\tau_\nu = 0$, we have :

$$u(\tau_\nu = 0, -\mu, \nu) - \mu\ \frac{du}{d\tau_\nu}(\tau_\nu = 0, -\mu, \nu) = I_{sup} \tag{1.1.16}$$

We discretize equation (1.1.16) :

$$\left[\frac{du}{d\tau_\nu}(\tau_\nu = 0, -\mu, \nu)\right]_{d=1} = \frac{u_2 - u_1}{(\tau_2 - \tau_1)\phi_\nu} = \frac{u_2 - u_1}{\Delta\tau_{\nu,1}}.$$

Then, equation (1.1.16) can be written as :

$$u_1 - \mu\ \frac{u_2 - u_1}{\Delta\tau_{\nu,1}} = I_{sup} \Longleftrightarrow u_1\left(1 + \frac{\mu}{\Delta\tau_{\nu,1}}\right) - \mu\ \frac{u_2}{\Delta\tau_{\nu,1}} = I_{sup},$$

with

$$B_1 = 1 + \frac{\mu}{\Delta\tau_{\nu,1}},\ C_1 = \frac{\mu}{\Delta\tau_{\nu,1}} \text{ and } S_1 = I_{sup}.$$

■

As (1.1.13) $\Longleftrightarrow I_{\mu\nu} = 2u - I_{-\mu\nu}$ and (1.1.14) $\Longleftrightarrow I_{-\mu\nu} = u - v$, then

(1.1.13) $\Longleftrightarrow I_{\mu\nu} = u + v \Longleftrightarrow v = I_{\mu\nu} - u$.

Hence (1.1.12) $\Longleftrightarrow \mu \dfrac{du}{d\tau_\nu} = I_{\mu\nu} - u \Longleftrightarrow I_{\mu\nu} = u + \mu \dfrac{du}{d\tau_\nu}$

For $\tau_\nu = \tau_{\nu_{max}}$, we have :

$$u(\tau_\nu = \tau_{\nu_{max}}, \mu, \nu) + \mu \frac{du}{d\tau_\nu}(\tau_\nu = \tau_{\nu_{max}}, \mu, \nu) = I(\tau_\nu = \tau_{\nu_{max}}, \mu, \nu) = I_{inf}(\mu) \tag{1.1.17}$$

We discretize equation (1.1.17) :

$$\left[\frac{du}{d\tau_\nu}(\tau_\nu = \tau_{\nu_{max}}, \mu, \nu)\right]_{d=N_d} = \frac{u_{N_d} - u_{N_d-1}}{(\tau_{N_d} - \tau_{N_d-1})\phi_\nu} = \frac{u_{N_d} - u_{N_d-1}}{\Delta\tau_{\nu,N_d-1}}$$

Then discretized equation (1.1.17) can be written as :

$$u_{N_d} + \mu \frac{u_{N_d} - u_{N_d-1}}{\Delta\tau_{\nu,N_d-1}} = I_{inf}$$

$$\Longleftrightarrow \frac{-\mu}{\Delta\tau_{\nu,N_d-1}} u_{N_d-1} + \left(1 + \frac{\mu}{\Delta\tau_{\nu,N_d-1}}\right) u_{N_d} = I_{inf},$$

with

$$B_{N_d} = 1 + \frac{\mu}{\Delta\tau_{\nu,N_d-1}},\ A_{N_d} = \frac{\mu}{\Delta\tau_{\nu,N_d-1}} \text{ and } S_{N_d} = I_{inf}.$$

■

In summary, solving RTE (1.1.1)-(1.1.7) is equivalent to solving the following linear system :

$$T.u = S, \tag{1.1.18}$$

where T is a tridiagonal matrix of size $N_d \times N_d$ whose terms are :

$$T = \begin{pmatrix} B_1 & -C_1 & & \\ -A_2 & \ddots & \ddots & \\ & \ddots & \ddots & -C_{N_d-1} \\ & & -A_{N_d} & B_{N_d} \end{pmatrix} \tag{1.1.19}$$

with

- $\forall \mu,\ \forall \nu,\ \forall 2 \leq d \leq N_d - 1$, we have :

$$-A_d\ u_{d-1} + B_d\ u_d - C_d\ u_{d+1} = S_d,$$

$$A_d = \frac{2\mu^2}{\Delta\tau_{\nu,d-1}(\Delta\tau_{\nu,d} + \Delta\tau_{\nu,d-1})},\ B_d = 1 + \frac{2\mu^2}{\Delta\tau_{\nu,d}\ \Delta\tau_{\nu,d-1}},$$
$$C_d = \frac{2\mu^2}{\Delta\tau_{\nu,d}(\Delta\tau_{\nu,d} + \Delta\tau_{\nu,d-1})}$$

- $B_1 = 1 + \dfrac{\mu}{\Delta\tau_{\nu,1}},\ C_1 = \dfrac{\mu}{\Delta\tau_{\nu,1}}$

- $A_{N_d} = \dfrac{\mu}{\Delta\tau_{\nu,N_d-1}},\ B_{N_d} = 1 + \dfrac{\mu}{\Delta\tau_{\nu,N_d-1}}$

The second member S is :

$$S = \begin{pmatrix} I_{sup} \\ S_2 \\ \vdots \\ S_{N_d-1} \\ I_{inf} \end{pmatrix} \tag{1.1.20}$$

In the case of a semi-infinite atmosphère, $I_{inf} = B$ and $I_{sup} = 0$.

In the following sections, we introduce two methods to solve system (1.1.18) : Λ-iteration method and ALI (Accelerated Lambda iteration) method.

1.1.3 Λ-iteration method

$\forall \mu,\ \forall \nu$, we solve linear system (1.1.18) by Gaussian elimination (formulas A4 and A5 of appendix A from Rybicki and Hummer (1991)) :

the following quantities are introduced : $\forall \mu,\ \forall \nu$,

$$\begin{aligned} D_d &= (B_d - A_d\ D_{d-1})^{-1}\ C_d,\quad \forall 2 \leq d \leq N_d - 1 \\ D_1 &= B_1^{-1}\ C_1 \\ Z_d &= (B_d - A_d\ D_{d-1})^{-1}(S_d + A_d\ Z_{d-1}),\quad \forall 2 \leq d \leq N_d \\ Z_1 &= B_1^{-1}\ S_1 \end{aligned} \tag{1.1.21}$$

The following Gaussian elimination scheme is considered :

$$\begin{cases} u_d = D_d\ u_{d+1} + Z_d,\quad \forall 1 \leq d \leq N_d \\ u_{N_d+1} = 0 \end{cases} \tag{1.1.22}$$

The Λ-iteration algorithm for solving RTE (1.1.1) or (1.1.7) is :

1. We initialize source function S to a starting value, for example to Planck function $B = 1$

2. For each frequency ν and direction μ :

   - we compute coefficients of the tridiagonal matrix T for each optical depth τ_ν
   - we compute solution u by Gaussian elimination (1.1.22)
   - we compute $J_\nu(\tau) = \int_{-1}^{1} I_{\mu\nu}\, d\mu = 2\int_0^1 u\, d\mu = \sum_{i=1}^{N_\mu} u_i.\alpha_i$, where $(\alpha_i)_i$ are the standard integration weights relative to direction μ
   - we compute $\overline{J}(\tau) = \int_0^\infty J_\nu\, \phi_\nu\, d\nu = \sum_{j=1}^{N_{freq}} J_\nu(j).W(j)$, where W is the integration weight in frequency and must be proportional to ϕ_ν and normalized

3. $S_1 = I_{sup}$ (BC)

4. $S_{N_d} = I_{inf}$ (BC)

5. $S(\tau) = (1-\epsilon)\overline{J}(\tau) + \epsilon B$

We repeat steps 2 to 5 several times (N_{iter}) for convergence.

For $N_{iter} = 200$, Λ-iteration method still does not converge to the exact solution (we obtain the same result as Paletou (2001), Figure 1) which starts from $S(\tau = 0) = B\sqrt{\epsilon}$ (formula (65) from Hummer and Rybicki (1967)).

Figure 1.3 represents source function S as a function of optical depth τ for a Doppler profile $\left(\phi_\nu = e^{-x^2},\ x = \dfrac{\nu - \nu_0}{\Delta\nu_D}\right)$, $B = 1$, $\epsilon = 10^{-4}$.

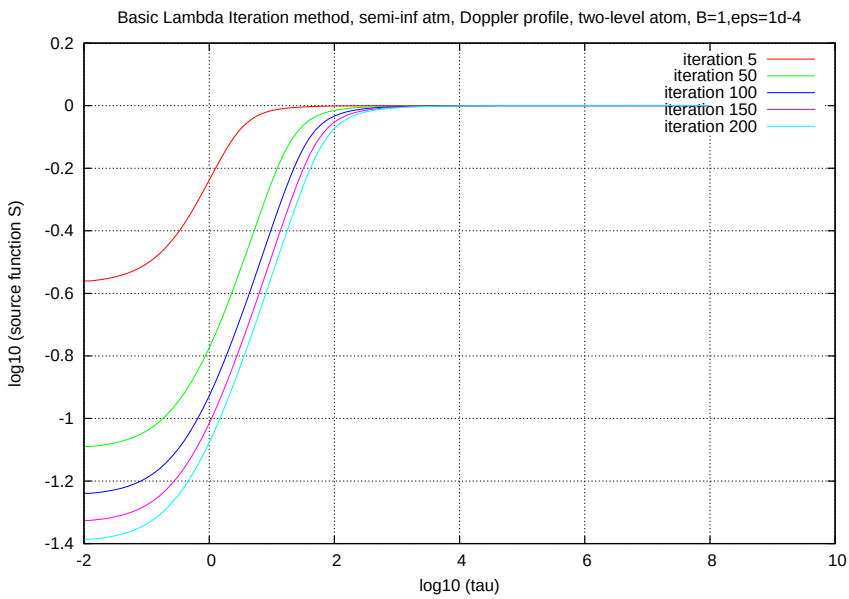

Figure 1.3: Source function as a function of optical depth (logarithmic scale), using Λ-iteration method for a Doppler profile (case of a semi-infinite atmosphere). Here, $B = 1$, $\epsilon = 10^{-4}$. After 200 iterations, the method still does not converge to the exact solution.

In the following section, we introduce ALI (Accelerated Lambda iteration) method which converges faster to the exact solution.

1.1.4 ALI method (Accelerated Lambda Iteration)

We consider the following iterative scheme :

$$\begin{cases} \Lambda = \Lambda^* + (\Lambda - \Lambda^*) \\ S^{n+1} = S^n + \Delta S \end{cases} \tag{1.1.23}$$

where n is the iteration index for ALI convergence, Λ^* is the exact diagonal of the full operator Λ, and S is the source function. We will detail further ΔS.
By definition,

$$\bar{J} = \Lambda\ S, \tag{1.1.24}$$

with

$$\Lambda = \sum_{\nu} \sum_{\mu} \Lambda_{\mu\nu} . \gamma_{\mu\nu} \tag{1.1.25}$$

The full operator Λ is also called the "big Lambda operator". The operator $\Lambda_{\mu\nu}$ is called the "small Lambda operator", is given by (1.1.18) and is defined by (formula (1.1) from Rybicki and Hummer (1991)) :

$$I_{\mu\nu} = \Lambda_{\mu\nu}\ [S] \tag{1.1.26}$$

Proof of (1.1.25) :

$$\bar{J} = \sum_{\nu} J_\nu . \alpha_\nu, \quad \alpha_\nu \text{ integration weights in frequency}$$

$$= \sum_{\nu} \sum_{\mu} I_{\mu\nu} . \alpha_\nu . \beta_\mu, \quad \beta_\mu \text{ integration weights relative to the direction}$$

$$= [\sum_{\nu} \sum_{\mu} \Lambda_{\mu\nu} . \gamma_{\mu\nu}] \; S$$

■

We don't compute Λ, we rather compute $\Lambda^* = \sum_{\mu,\nu} \Lambda^*_{\mu,\nu} . \delta_{\mu,\nu}$, where $\delta_{\mu,\nu}$ is the integration weights relative to frequencies ν and to directions μ. $\Lambda^*_{\mu,\nu}$ is a diagonal matrix containing diagonal elements of the matrix T^{-1} (and not T) according to (1.1.18).

Remark 1.1.3 *$\Lambda_{\mu\nu}$, $\Lambda^*_{\mu\nu}$, Λ^* and Λ matrices have same size $N_d \times N_d$.*

ALI iterative scheme is :

$$\begin{aligned} S_k^{n+1} &= S_k^n + \Delta S_k, \text{ avec} \\ \Delta S_k &= \frac{(1-\epsilon)\bar{J}_k^n + \epsilon B_k - S_k^n}{1 - (1-\epsilon)\Lambda^*_{kk}} \end{aligned} \tag{1.1.27}$$

n is the index on iterations (ALI convergence) and k is the index on optical depth.

Proof of (1.1.27) :

We start from formula of the source function (1.1.2) :

$$S = (1-\epsilon)\bar{J} + \epsilon B, \text{ where } \bar{J} = \Lambda\,[S].$$

We write the following iterative scheme (formula (9) from Olson et al. (1986)) :

$$\forall k = 1, ..., N_d \quad S_k^{n+1} = (1-\epsilon)\Lambda\,[S_k^n] + \epsilon\, B_k.$$

As $\Lambda = \Lambda^* + (\Lambda - \Lambda^*)$, then according to the formula (10) from Olson et al. (1986), we obtain :

$$S_k^{n+1} = (1-\epsilon)\Lambda^*\,[S_k^{n+1}] + (1-\epsilon)(\Lambda - \Lambda^*)S_k^n + \epsilon\, B_k \tag{1.1.28}$$

$$
\begin{aligned}
(1.1.28) &\iff S_k^{n+1} - (1-\epsilon)\Lambda^* \; S_k^{n+1} = (1-\epsilon)(\Lambda - \Lambda^*)S_k^n + \epsilon \; B_k \\
&\iff [1-(1-\epsilon)\Lambda^*]S_k^{n+1} = (1-\epsilon)(\Lambda - \Lambda^*)S_k^n + \epsilon \; B_k \\
&\iff S_k^{n+1} = [1-(1-\epsilon)\Lambda^*]^{-1}[(1-\epsilon)(\Lambda - \Lambda^*)S_k^n + \epsilon \; B_k] \text{ (formula (11) from Olson et al. (1986))} \\
&\iff S_k^{n+1} = \frac{(1-\epsilon)(\Lambda - \Lambda^*)S_k^n + \epsilon \; B_k}{1-(1-\epsilon)\Lambda^*} \\
&\iff S_k^{n+1} = \frac{(1-\epsilon)\Lambda \; S_k^n - (1-\epsilon)\Lambda^* \; S_k^n + \epsilon \; B_k}{1-(1-\epsilon)\Lambda^*} \\
&\iff S_k^{n+1} = \frac{(-(1-\epsilon)\Lambda^* \; S_k^n + S_k^n) - S_k^n + (1-\epsilon)\Lambda \; S_k^n + \epsilon \; B_k}{1-(1-\epsilon)\Lambda^*} \\
&\iff S_k^{n+1} = \frac{[1-(1-\epsilon)\Lambda^*]S_k^n - S_k^n + (1-\epsilon)\Lambda \; S_k^n + \epsilon \; B_k}{1-(1-\epsilon)\Lambda^*} \\
&\iff S_k^{n+1} = S_k^n + \frac{(1-\epsilon)\Lambda \; S_k^n + \epsilon \; B_k - S_k^n}{1-(1-\epsilon)\Lambda^*} \\
&\iff S_k^{n+1} = S_k^n + \frac{(1-\epsilon)\overline{J}_k^n + \epsilon \; B_k - S_k^n}{1-(1-\epsilon)\Lambda^*_{kk}} \quad \text{since } \overline{J} = \Lambda \; S \\
&\iff S_k^{n+1} = S_k^n + \Delta S_k^{ali}
\end{aligned}
$$

■

Computation of $\Lambda^*_{\mu\nu}$ (diagonal matrix) :

$$
\Lambda^*_{\mu\nu} = \begin{pmatrix} T_{11}^{-1} & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & & & \vdots \\ \vdots & & T_{ii}^{-1} & & \\ \vdots & & & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & T_{N_dN_d}^{-1} \end{pmatrix}
$$

where $T = \Lambda_{\mu\nu}$ defined by (1.1.19).
Let $\lambda = T^{-1}$. We compute diagonal elements λ_{ii} without inverting matrix T. According to appendix B from Rybicki and Hummer (1991), we have : $\forall \; 2 \leq i \leq N_d - 1$

$$
\lambda_{ii} = (1 - D_i \; E_{i+1})^{-1}.(B_i - A_i \; D_{i-1})^{-1}, \tag{1.1.29}
$$

with

$$\begin{cases} D_i = (B_i - A_i\ D_{i-1})^{-1}.C_i \\ D_0 = 0 \\ E_i = (B_i - C_i\ E_{i+1})^{-1}.A_i \\ E_{N_d+1} = 0 \end{cases} \tag{1.1.30}$$

Here,

$$\begin{aligned} \lambda_{11} &= (1 - D_1\ E_2)^{-1}.(B_1 - A_1\ D_0)^{-1} = B_1^{-1}.(1 - D1\ E_2)^{-1} \\ \lambda_{N_d N_d} &= (1 - D_{N_d}\ E_{N_d+1})^{-1}.(B_{N_d} - A_{N_d}\ D_{N_d-1})^{-1} = 1/(B_{N_d} - A_{N_d}.D_{N_d-1}) \end{aligned} \tag{1.1.31}$$

The algorithm by ALI method for solving RTE (1.1.1)-(1.1.7) is :

1. We initialize source function S to a starting value, for example to Planck function $B = 1$

2. For each frequency ν and for each direction μ, we compute the diagonal matrix $\Lambda^*_{\mu\nu}(\tau)$ which is stored as a vector of size N_d. We use formulas (1.1.29-1.1.31) for λ_{ii}

3. We compute $\Lambda^* = \sum_{\mu,\nu} \Lambda^*_{\mu\nu}.\delta_{\mu\nu}$, where $\delta_{\mu\nu}$ is the integration weights in ν (which is proportional to ϕ_ν and normalized) and in μ

4. We loop on $N_{iter} = 200$ iterations, to :

   - compute $\overline{J}$ (and J_ν) by Gaussian elimination using formula (1.1.22)

   - compute $S(\tau)$ using formula (1.1.27) for each optical depth and having previously initialized $S_1 = I_{sup}$ and $S_{N_d} = I_{inf}$

ALI method converges after 100 iterations to the exact solution. Figure 1.4 represents source function S as a function of optical depth τ for a Doppler profile, $B = 1$, $\epsilon = 10^{-4}$. We have the same results as figure 1 from Paletou (2001).

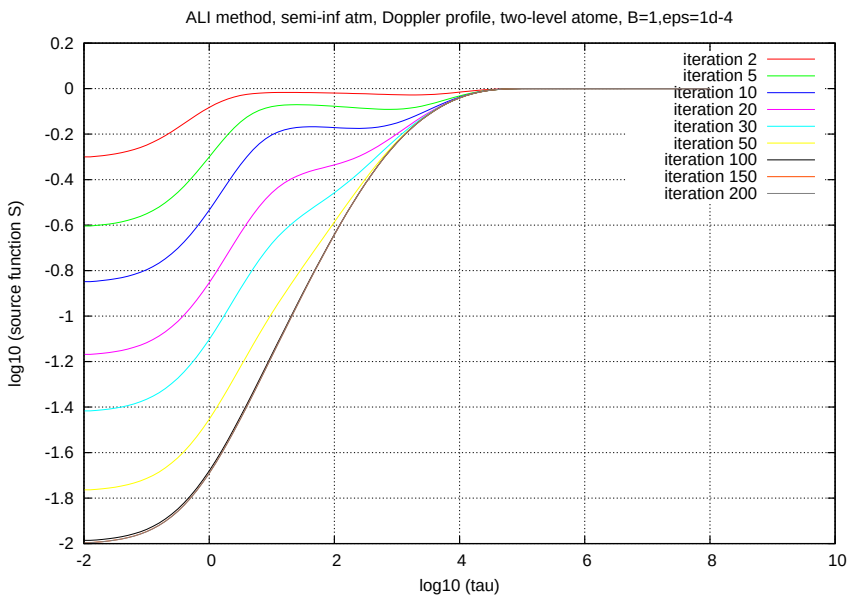

Figure 1.4: Source function as a function of optical depth (logarithmic scale), using ALI method for a Doppler profile (case of a semi-infinite atmosphere). Here, $B = 1$, $\epsilon = 10^{-4}$. Convergence is reached after 100 iterations.

In next section, we accelerate ALI method.

1.1.5 Acceleration of ALI scheme by Ng method

The aim is to accelerate ALI scheme by Ng (1974) method. Ng method (Ng acceleration is a linear combination of successive iterations of source function S) consists in accelerating every three iterations of ALI convergence.

As we have seen before, ALI iterative scheme for the source function S is (see section 1.1.4) :

$$S_k^{n+1} = S_k^n + \Delta S_k \equiv F(S_k^n) \tag{1.1.32}$$

n is the index on the convergence iterations of ALI scheme, k is the index of the optical depth grid τ.
According to section 5 from Olson et al. (1986), we consider the following three-point scheme: Let $Y^n = S_k^n(\tau)$ be a vector on the grid of optical depths. We accelerate the calculation of the source function S.
Equation (1.1.32) can be written as :

$$Y^{n+1} = F(Y^n)$$

According to Olson et al. (1986), we suppose that the iterative scheme converges linearly and we accelerate the convergence by adopting a linear combination of 3 successive iterations :

$$Y^* = (1 - a - b)Y^{n-1} + a\ Y^{n-2} + b\ Y^{n-3} \tag{1.1.33}$$

Then

$$\begin{aligned} F(Y^*) &= (1-a-b)Y^n + a\ Y^{n-1} + b\ Y^{n-2} \\ &= Y^{n+1} = F(Y^n) \end{aligned} \tag{1.1.34}$$

a and b are such that $\sum_{i=1}^{N_d}[y_i^* - F(y_i^*)]^2.W_i$ is minimal. $Y^* = (y_i^*)_{1\leq i\leq N_d}$ and W_i is the weight vector.

This is equivalent to solving the following system of linear equations :

$$\begin{cases} a\ A_1 + b\ B_1 = C_1 \\ a\ A_2 + b\ B_2 = C_2 \end{cases} \tag{1.1.35}$$

The choice of W_i (weight) is important. We choose according to Olson et al. (1986) :

$$W_i = [\bar{J}^n(\tau_i)]^{-1}$$

We can choose $W_i = 1$ but it converges a little slower (see figures 1.5 and 1.6).
The solution of system (1.1.35) is :

$$\begin{cases} a = (C_1\ B_2 - C_2\ B_1)\ /\ (A_1\ B_2 - A_2\ B_1) \\ b = (C_2\ A_1 - C_1\ A_2)\ /\ (A_1\ B_2 - A_2\ B_1) \end{cases} \tag{1.1.36}$$

with

$$\begin{aligned} A_1 &= \sum_{i=1}^{N_d}(y_i^n - 2y_i^{n-1} + y_i^{n-2})^2.W_i \\ B_1 &= \sum_{i=1}^{N_d}(y_i^n - y_i^{n-1} - y_i^{n-2} + y_i^{n-3}).W_i.(y_i^n - 2y_i^{n-1} + y_i^{n-2}) \\ A_2 &= B_1 \\ B_2 &= \sum_{i=1}^{N_d}(y_i^n - y_i^{n-1} - y_i^{n-2} + y_i^{n-3})^2.W_i \\ C_1 &= \sum_{i=1}^{N_d}(y_i^n - 2y_i^{n-1} + y_i^{n-2})\ (y_i^n - y_i^{n-1}).W_i \\ C_2 &= \sum_{i=1}^{N_d}(y_i^n - y_i^{n-1} - y_i^{n-2} + y_i^{n-3})\ (y_i^n - y_i^{n-1}).W_i \end{aligned} \tag{1.1.37}$$

Ng algorithm can be summarized in two steps :

- we start from an initial value of the source function S
- the accelerated vector Y^n is computed after the first four iterations, then after each three normal iterations of ALI scheme.

Accelerated ALI algorithm is :

- Loop on $N_{iter} = 200$ (iterations)
  1. we compute $\overline{J}$ using Gaussian elimination (1.1.22)
  2. we compute S using formula (1.1.32)
  3. by using a counter, subroutine "acceleration"is called after 3 normal ALI iterations
- End loop

Subroutine "acceleration" takes as input : $\overline{J}$, S, $iter = n$ (current iteration). It computes weight $W_i = [\overline{J}]^{-1}$; values A_1, B_1, B_2, C_1, C_2, A_2; values a and b. It computes S_k^{n+1} using formula (1.1.34) :

$$S_k^{n+1} = (1 - a - b)S_k^n + aS_k^{n-1} + bS_k^{n-2} \equiv F(S_k^n)$$

Then, we put this result in S (input).

Accelerated ALI method converges to the exact solution after 20 iterations : figure 1.5 represents source function S as a function of optical depth τ (logarithmic scale) for a Doppler profile, $B = 1$, $\epsilon = 10^{-4}$, using weight $W_i = [\overline{J}^n]^{-1}$ for acceleration. When we take $W_i = 1$, accelerated ALI method converges to the exact solution after 30 iterations (see figure 1.6).

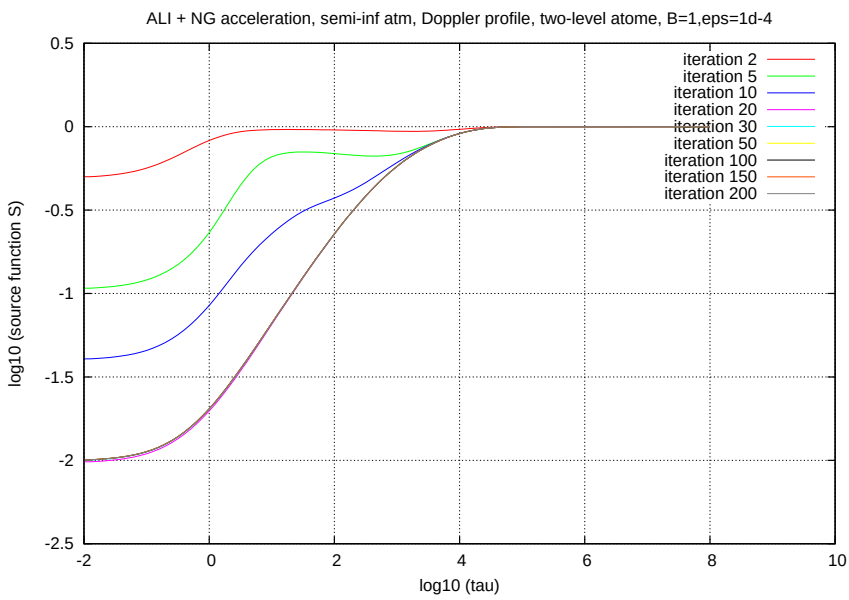

Figure 1.5: Source function S as a function of optical depth on a logarithmic scale, by accelerated ALI method for a Doppler profile (case of a semi-infinite atmosphere). Here, $B = 1$, $\epsilon = 10^{-4}$, $W_i = [\overline{J}^n]^{-1}$. Convergence is reached after 20 iterations.

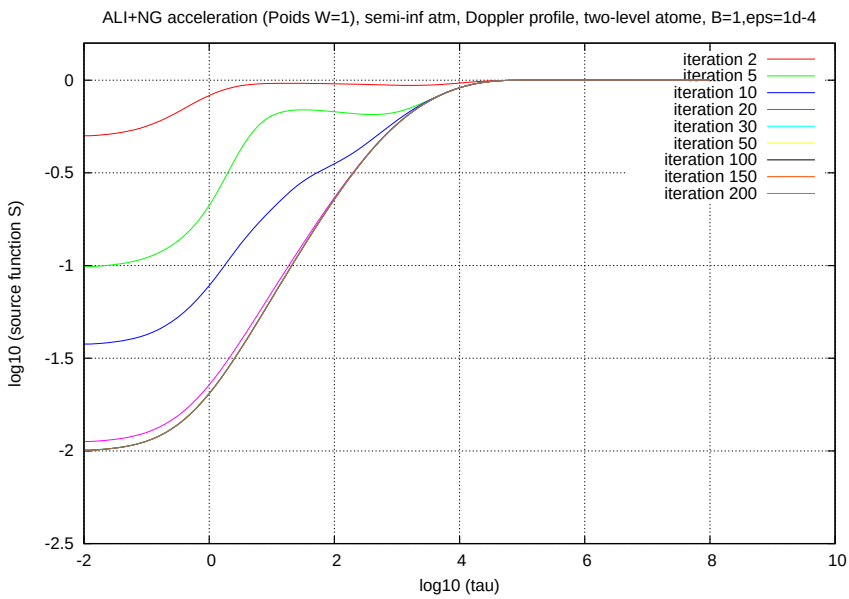

Figure 1.6: Source function S as a function of optical depth on a logarithmic scale, by accelerated ALI method for a Doppler profile (case of a semi-infinite atmosphere). Here, $B = 1$, $\epsilon = 10^{-4}$, $W_i = 1$. Convergence is reached after 30 iterations.

1.1.6 Description of numerical code

Nature of the physical problem : NLTE radiative transfer (1D) in semi-infinite atmosphere for a two-level atom (a fundamental and a fictitious line), without internal velocity field

Method of solution : improved Feautrier method (Rybicki and Hummer, 1991) combined with :
- Λ-iteration method
- ALI (Accelerated Lambda Iteration) method
- acceleration of ALI scheme using Ng method

Other relevant information : we use complete frequency redistribution (CRD)

Authors : M. Chane-Yook & P. Gouttebroze

Program available from :
https://idoc.ias.u-psud.fr/MEDOC/Radiative transfer codes/Tools for radiative transfer

Computer(s) on which program hab been tested : PC with 4 Intel processors (2.67GHz)

Operating System(s) for which version of program has been tested : Linux

Programming language used : Fortran 90/95 (with **gfortran** compiler)

Status : stable

Accessibility : open (MEDOC)

No. of code lines in combined program and test deck : 594

Typical running time : < 1 min for 20 iterations of accelerated ALI cycle

References :
- G. B. Rybicki & D. G. Hummer, "An accelerated lambda iteration method for multilevel radiative transfer. I- Non-overlapping lines with background continuum", A&A, 245, 171-181, 1991
- F. Paletou, "Transfert de rayonnement : méthodes itératives", C. R. Acad. Sci. Paris, t.2, Série IV, 885-898, 2001

In next section, we describe subroutines used in the program.

1.1.7 Description of subroutines

The main program "lambda_it.f90" calls 3 subroutines :

- **grilles** : implementation of frequency grids, of optical thicknesses grid, direction grid and Voigt profile

- **methode_lambda_it_simple(S,J_bar,J_nu)** : Λ-iteration method

- **methode_ali(S_ali,J_bar_ali,J_nu_ali)** : ALI method + Ng acceleration

The main program "lambda_it.f90" uses several modules whose files are (see below for the set of variables used) :

- **param.f90** : contains global variables as well as constants defined as B, ϵ, n_{iter}, ...

- **general.f90** : contains several subroutines like definition of grids, computation of tridiagonal matrix coefficients (1.1.19), computation of integration weights, Voigt function, Gaussian elimination formulas (1.1.22), computation of J_ν and $\overline{J}$

- **lambda_it_simple.f90** : deals with Λ-iteration method

- **ali.f90** : contains several subroutines like ALI scheme, computation of diagonal elements of diagonal matrix $\Lambda^*_{\mu\nu}$ (1.1.29)-(1.1.31), computation of Λ^*, computation of source function S, computation of $\overline{J}$, formulas for Ng acceleration

1.1.7.1 Set of variables used in the program

⋆ **Module param_mod (param.f90 file)** :

- pi : π value
- coeff_extinction=10^{-4} : ϵ value
- BB=1 : Planck function value
- niter=200 : number of iterations for convergence of accelerated ALI and Λ-iteration schemes
- T : electron temperature
- I_sup=0, I_inf=1 : upper and lower boundary conditions
- a_voigt=10^{-3} : parameter a for Voigt function. When $a = 10^{-3}$, we have Doppler function
- nfr=15 : size of grid of reduced frequencies XFR
- nxmod=101 : size of optical thickness grid xmod
- nmu=4 : size of direction grid $\mu = \cos\theta$
- xmod : array of size nxmod corresponding to the generic optical depth $\tau = (\tau_d)$
- xfr : array of size nfr corresponding to reduced frequency x_i
- mu : array of size nmu corresponding to values of $\mu = \cos\theta$ (direction)
- J_bar_ali,J_bar : arrays of size nxmod corresponding to $\overline{J}$, respectively for Λ-iteration method and for ALI method
- J_nu_ali,J_nu : arrays of size (nfr,nxmod) corresponding to J_ν, respectively for Λ-iteration method and for ALI method
- S_ali,S : arrays of size (nxmod,niter) corresponding to source function S, respectively for Λ-iteration method and for ALI method
- u_ali,u : arrays of size (nxmod,nfr,nmu) corresponding to solution of RTE, respectively for Λ-iteration method and for ALI method

- lambda_etoile : array of size nxmod corresponding to matrix Λ^*
- lambda_etoile_mu_nu : array of size (nxmod,nfr,nmu) corresponding to matrix $\Lambda^*_{\mu\nu}$
- tau_nu : array of size (nxmod,nfr) corresponding to the optical depth $\tau_\nu = \phi_\nu\ \tau$
- phi_nu : array of size nfr corresponding to the line profile (Voigt profile) ϕ_ν (not normalized)
- M : array of size nfr corresponding to integration weights (intermediate) in frequency
- W : array of size nfr corresponding to integration weights in frequency, normalized and proportional to profile ϕ_ν

⋆ **Module lambda_it_simple_mod (lambda_it_simple.f90 file)** :

- a,b,c,d : arrays of size nxmod corresponding to tridiagonal matrix coefficients A, B, et C (1.1.19) and to coefficient D of Gaussian elimination (1.1.21)

⋆ **Module ali_mod (ali.f90 file)** :

- a,b,c,d,e : arrays of size nxmod corresponding to tridiagonal matrix coefficients A, B, et C (1.1.19), to coefficient D of Gaussian elimination (1.1.21) and to coefficient E of diagonal matrix $\Lambda^*_{\mu\nu}$ (1.1.30)

1.1.7.2 Subroutine description in each module

Module general_mod (general.f90 file)

Module **general_mod** contains subroutines used for Λ-iteration and ALI methods.

Subroutine grilles : this subroutine provides different grids for modeling a semi-infinite atmosphere. More specifically,
- we consider a generic optical thickness grid τ in cm defined by array xmod of size nxmod
- we consider a grid of reduced frequencies (x_i, i=1,..,nfr) defined by array XFR of size nfr
- we choose Voigt profile as line profile ϕ_ν (at each frequency), non-normalized. The profile is represented by array phi_nu
- the optical depth grid at a given frequency (τ_ν) is equal to τ multiplied by ϕ_ν (1.1.5)
- we consider a grid of directions $\mu = \cos\theta$ (array mu). μ is non-zero

- we calculate weights relative to frequency (used for computation of $\overline{J}$) : since profile ϕ_ν is not normalized, we compute weight W normalized and proportional to ϕ_ν. For this, we first compute weights M in frequency using subroutine *trapez*, which is multiplied by ϕ_ν and which is normalized. Thus we obtain W.

Subroutine trapez : this subroutine calculates integration weights.

Subroutine coeff_mat_tridiagonale : a, b, c are respectively coefficients A_d, B_d, C_d, $1 \leq d \leq N_d = nxmod$ of tridiagonal matrix T (1.1.19).

Function VOIGT, Function W4: This is the VOIGT function described in Humlicek (1982) and Hui (1978).

Subroutine elimination_gauss : This subroutine implements formulas for Gaussian elimination. Vector EE of this subroutine is the second member (1.1.20) of linear system (1.1.18). The quantities d and z are D_d and Z_d (1.1.21). Variables a, b and c are the coefficients of tridiagonal matrix T (1.1.19). u is the solution of RTE (1.1.7) given by formula (1.1.22).

Subroutine calcul_J_nu : This subroutine computes J_ν (1.1.4) using the following formula (see Λ-iteration algorithm in section 1.1.3) :
$J_\nu(\tau,\nu) = \int_{-1}^{1} I_{\mu\nu}\ d\mu = 2\int_0^1 u\ d\mu = \sum_{i=1}^{N_\mu} u_i.\alpha_i$, where $(\alpha_i)_i$ are integration weights. We choose $(\alpha_i)_i = 1/nmu$.

Subroutine calcul_J_bar : We compute $\overline{J}$ using the following formula (see Λ-iteration algorithm in section 1.1.3) :
$\overline{J}(\tau) = \int_0^\infty J_\nu\ \phi_\nu\ d\nu = \sum_{j=1}^{N_{freq}} J_\nu(j).W(j)$, where W is integration weight in frequency and is proportional to ϕ_ν and normalized. W is computed in subroutine *grilles*.

Module lambda_it_simple_mod (lambda_it_simple.f90 file)

This module contains subroutine *methode_lambda_it_simple* which uses the algorithm of Λ-iteration explained in section 1.1.3. As we consider boundary conditions of order 1, we make the following 2 initializations:
- the source function S at the first point of the grid τ_ν is equal to the source function at the second point of the grid
- the source function S at the last point of the grid τ_ν is equal to the source function at the previous point.

Module ali_mod (ali.f90 file)

This module contains subroutines needed to solve RTE, using ALI method.

Subroutine methode_ali : this subroutine uses ALI algorithm described in section 1.1.4 as well as acceleration of ALI scheme described in 1.1.5.

Subroutine elements_diag_lambda_etoile_mu_nu : we compute diagonal matrix $\Lambda^*_{\mu\nu}$ given by formulas (1.1.29)-(1.1.31).

Subroutine operateur_lambda_etoile : we compute operator Λ^* given by the following formula :

$\Lambda^* = \sum_{\mu,\nu} \Lambda^*_{\mu\nu}.\delta_{\mu\nu}$, where $\delta_{\mu\nu}$ is the integration weight in ν (which is proportional to ϕ_ν and normalized) and in μ. We choose for integration weight in μ : $1/nmu$. Integration weight in frequency used here is the one (W) computed in subroutine *grilles*.

Subroutine calcul_S_ali : we compute source function S in an iterative way, given by formula (1.1.27). As for Λ-iteration scheme, we consider boundary conditions of order 1, thus we make the same initializations, namely :
- the source function S at the first point of the grid τ_ν is equal to the source function at the second point of the grid
- the source function S at the last point of the grid τ_ν is equal to the source function at the previous point.

Subroutine calcul_J_bar_ali : we compute $\overline{J}$ in the same way as in subroutine *methode_lambda_it_simple*.

Subroutine acceleration_ali : we implement algorithm of acceleration of ALI scheme (section 1.1.5), in particular coefficients A_1, B_1, A_2, B_2, C_1, C_2 given by (1.1.37). The quantities a_acc and b_acc are coefficients a and b of formulas (1.1.36).

1.1.8 Running numerical program

- make clean
- make
- ./lambda_it

Result files (fort.112, fort.113, fort.92) are read by **gnuplot** software (instructions are in subroutines *methode_ali* and *methode_lambda_it_simple*).

1.2 Solar filament

In this section, we solve non-LTE radiative transfer equation (RTE) using accelerated ALI method in the case of a solar filament, in 1D, with CRD (complete frequency redistribution)

and Voigt profile. For this, we consider a realistic and finite layer (horizontal layer, no symmetry here) and a realistic two-level atom (hydrogen atom and Lyman Alpha line $L\alpha$).

In this case, coefficient a of Voigt profile, ϵ and B are not constant values. We need to calculate them. Moreover, it is necessary to introduce population equations.

1.2.1 Filament modeling

In the modeling (see figure 1.7), a filament is represented by a horizontal plane-parallel slab with optical depth τ, located above the surface of the sun.

In the case of a filament, the probability of collisional destruction ϵ (or extinction coefficient) is computed from transition rates by electronic collision (C_{12}, C_{21}) and Einstein coefficients (A_{21}, B_{12}, B_{21}, $A_{12} = 0$).

Altitude of the filament is used to calculate dilution factors, which are used to compute incident intensities from solar intensities (input) and therefore to calculate lower BC (boundary condition) I_{inf} (here $I_{sup} = 0$).

1.2.2 Two-level atom and statistical equilibrium

The transitions considered here are Bound-Free and Bound-Bound, which are represented in figure 1.8. Bound-Free transitions are between a bound state i and a continuum, producing a free electron with energy ϵ. It starts from excited states limit, i.e. $\epsilon = 0$. Bound-Bound transitions are from level i to level j.

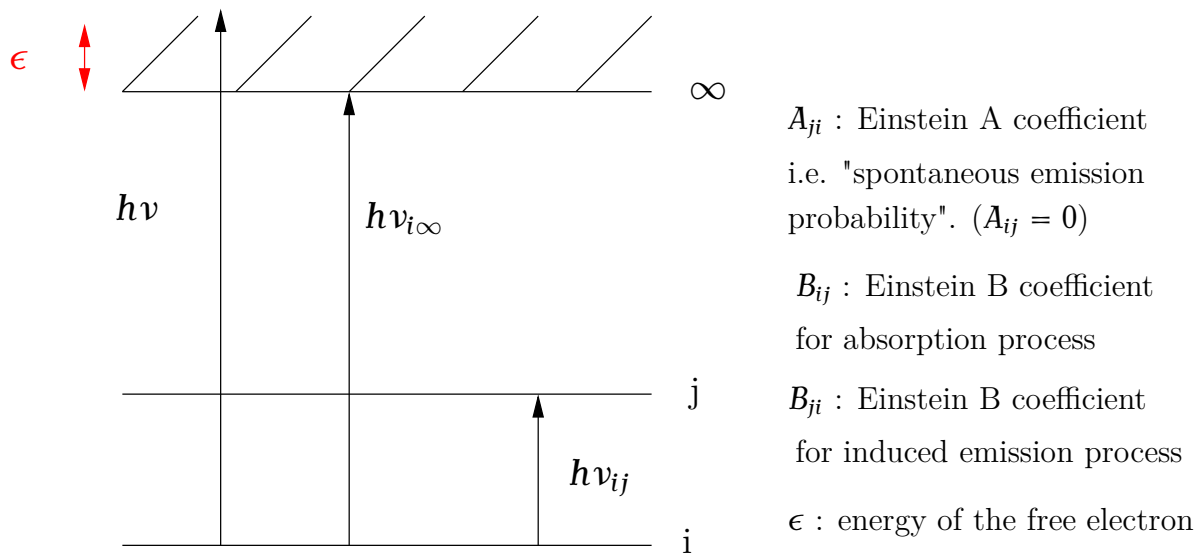

Figure 1.8: Transition types considered.

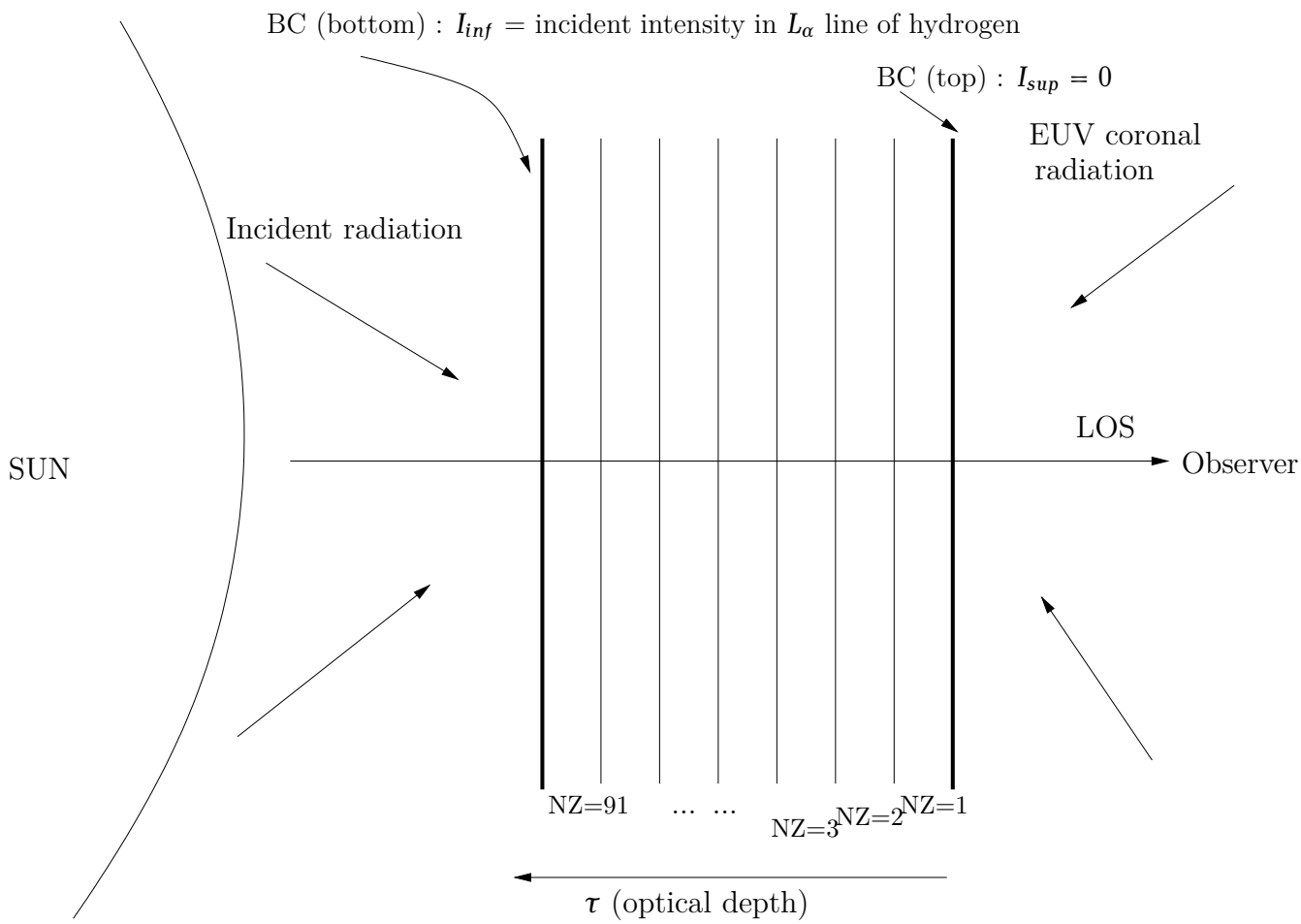

Figure 1.7: Solar filament model (figure is not to scale).

Let's consider level j in figure 1.8. We introduce population equations (or equations of statistical equilibrium) for this level which give the equilibrium between population and depopulation processes of level j from other levels i :

$$\sum_{j\neq i}^{n} N_i \; P_{ij} = N_j \sum_{j\neq i}^{n} P_{ji} \tag{1.2.1}$$

N_i (respectively N_j) is the (density of) population of level i (respectively of level j).
The transition rate of level j (depopulation of j) is :

$$P_{ji} = A_{ji} + B_{ji}\bar{J}_{ij} + C_{ji} \tag{1.2.2}$$

where C_{ji} is the collisional deexcitation rate and is proportional to the electron density. A_{ji} and B_{ji} are Einstein coefficients.
The transition rate of level i (population of j) is :

$$P_{ij} = B_{ij}\bar{J}_{ij} + C_{ij}$$

where C_{ij} is the excitation rate by collisions. Here, $A_{ij} = 0$.
The right-hand side term in population equations (1.2.1) represents all processes which depopulate level j and the left-hand side term corresponds to all levels which populate level j. To close the system of population equations, we use the following conservation equation for the considered element (i.e. hydrogen atom) :

$$\sum_{j=1}^{n} N_j = N_T \tag{1.2.3}$$

where N_T is the total population of the atomic element.

In the case of a two-level atom ($n = 2$), population equations are reduced to :

$$\begin{aligned} N_1 \; P_{12} &= N_2 \; P_{21} \\ \Longleftrightarrow N_1 \; (B_{12} \; \overline{J} + C_{12}) &= N_2 \; (A_{21} + B_{21} \; \overline{J} + C_{21}) \end{aligned} \tag{1.2.4}$$

where

$$P_{12} = B_{12} \; \overline{J} + C_{12}$$

and

$$P_{21} = A_{21} + B_{21} \; \overline{J} + C_{21}$$

Here, $\overline{J}_{12} = \overline{J}$ since we have only one line ($L\alpha$).

We establish below the formulas necessary for implementation of the numerical code in the case of a solar filament (ϵ, B_{12}, B_{21}, C_{12}, C_{21}, ...).

- Absorption coefficient integrated on frequency (or total absorption coefficient) :

$$\overline{\kappa} = \frac{h\nu_0}{4\pi} N_1 \; B_{12}, \tag{1.2.5}$$

ν_0 being line center frequency ($L\alpha$). This formula (1.2.5) is valid when the profile is normalized, i.e. $\int \phi_\nu \, d\nu = 1$, ν being the frequency.

- Spontaneous emission :

$$\overline{e}_S = \frac{h\nu_0}{4\pi} N_2 \, A_{21} \tag{1.2.6}$$

- Induced emission :

$$\overline{e}_I = \frac{h\nu_0}{4\pi} N_2 \, B_{21} \, \overline{J} \tag{1.2.7}$$

- Relations between Einstein coefficients :

According to Jefferies (1968) :

$$B_{21} = \frac{c^2}{2h\nu_0^3} \, A_{21} \tag{1.2.8}$$

The relation between Einstein coefficients is :

$$g_1 \, B_{12} = g_2 \, B_{21}$$

$$\begin{aligned} &\Longleftrightarrow \frac{g_1}{g_2} \, B_{12} = \frac{c^2}{2h\nu_0^3} \, A_{21} \\ &\Longleftrightarrow B_{12} = \frac{g_2}{g_1} \frac{c^2}{2h\nu_0^3} \, A_{21} \end{aligned} \tag{1.2.9}$$

According to (1.2.5), the total absorption coefficient is :

$$\overline{\kappa} = \frac{h\nu_0}{4\pi} \, N_1 \, \frac{g_2}{g_1} \, \frac{c^2}{2h\nu_0^3} \, A_{21} \tag{1.2.10}$$

Then, the induced emission is :

$$\overline{e}_I = \frac{h\nu_0}{4\pi} \, N_2 \, \frac{c^2}{2h\nu_0^3} \, A_{21} \, \overline{J} \tag{1.2.11}$$

- Global source function : induced emission is treated as a negative absorption.

Two cases :

- Case 1 : simple case where $\overline{e}_S \propto N_2$, $\overline{\kappa} \propto N_1$ et $\overline{e}_I \propto N_2$

$$S = \frac{\overline{e}_S}{\overline{\kappa} - \overline{e}_I / \overline{J}} \tag{1.2.12}$$

- Case 2 : more complicated for statistical equilibrium

$$S = \frac{\overline{e}_S + \overline{e}_I}{\overline{\kappa}}$$

We consider case 1. According to (1.2.6), (1.2.10) and (1.2.11), we have :

$$S = \frac{N_2}{\frac{c^2}{2h\nu_0^3}\left(N_1\,\frac{g_2}{g_1} - N_2\right)} = \frac{2h\nu_0^3}{c^2}\,\frac{1}{\frac{N_1}{N_2}\,\frac{g_2}{g_1} - 1} \tag{1.2.13}$$

- Relation between C_{12} (excitation rate by collisions) and C_{21} (deexcitation rate by collisions) :

$$\begin{aligned} C_{12} &= \left(\frac{N_2}{N_1}\right)_{ETL} \times C_{21} \\ C_{12} &= \frac{g_2}{g_1}\,exp\left(-\frac{h\nu_0}{kT}\right) C_{21}, \quad \text{according to Saha law} \end{aligned} \tag{1.2.14}$$

T is electron temperature.

We eliminate B_{12}, B_{21} and C_{12} in population equations (1.2.4) :

$$\begin{aligned} (1.2.4) \iff N_1\Big[\frac{g_2}{g_1}\,\frac{c^2}{2h\nu_0^3}\,A_{21}\,\overline{J} + \frac{g_2}{g_1}\,exp\left(-\frac{h\nu_0}{kT}\right) C_{21}\Big] \\ = N_2\Big[A_{21} + \frac{c^2}{2h\nu_0^3}\,A_{21}\,\overline{J} + c_{21}\Big] \\ \iff \frac{N_1}{N_2}\,\frac{g_2}{g_1} - 1 = \frac{A_{21} + C_{21}\big[1 - exp\left(-\frac{h\nu_0}{kT}\right)\big]}{\frac{c^2}{2h\nu_0^3}\,A_{21}\,\overline{J} + C_{21}\,exp\left(-\frac{h\nu_0}{kT}\right)} \end{aligned}$$

Then, the source function is :

$$(1.2.13) \iff S = \frac{2h\nu_0^3}{c^2} \times \frac{C_{21}\,exp\left(-\frac{h\nu_0}{kT}\right) + \frac{c^2}{2h\nu_0^3}\,A_{21}\,\overline{J}}{C_{21}\big[1 - exp\left(-\frac{h\nu_0}{kT}\right)\big] + A_{21}} \tag{1.2.15}$$

Let

$$\epsilon_1 = \frac{C_{21}}{A_{21}}\Big[1 - exp\left(-\frac{h\nu_0}{kT}\right)\Big] \tag{1.2.16}$$

Hence

$$(1.2.16) \iff C_{21}\,exp\left(-\frac{h\nu_0}{kT}\right) = \frac{\epsilon_1\,A_{21}}{exp\left(-\frac{h\nu_0}{kT}\right) - 1} \tag{1.2.17}$$

Then, according to (1.2.16) and (1.2.17), we have :

$$(1.2.15) \iff S = \frac{2h\nu_0^3}{c^2} \times \frac{\frac{\epsilon_1 A_{21}}{exp\left(\frac{h\nu_0}{kT}\right)-1} + \frac{c^2}{2h\nu_0^3} A_{21}\,\bar{J}}{A_{21}\epsilon_1 + A_{21}}$$
$$\iff S = \frac{2h\nu_0^3}{c^2}\,\frac{\epsilon_1}{1+\epsilon_1}\,\frac{1}{exp(\frac{h\nu_0}{kT})-1} + \frac{\bar{J}}{1+\epsilon_1}$$

- Planck function B :

  In the previous expression of the source function S, we denote by :

$$B = \frac{2h\nu_0^3}{c^2}\,\frac{1}{exp(\frac{h\nu_0}{kT})-1} \tag{1.2.18}$$

  Then, the source function is : $S = \dfrac{\epsilon_1\, B + \bar{J}}{1+\epsilon_1}$.

- Probability of collisional destruction ϵ :

  Let :

$$\epsilon = \frac{\epsilon_1}{1+\epsilon_1} \tag{1.2.19}$$

  So we obtain the formula (1.1.2) of the source function $S = \epsilon\, B + (1-\epsilon)\,\bar{J}$, under the assumption $\int \phi_\nu\, d\nu = 1$.

Remark 1.2.1 *In practice, we compute ϵ and B. Then, we use accelerated ALI scheme to calculate S. Finally, we compute populations of levels N_1 et N_2.*

- Computation of N_2 : we start from (1.2.13)

$$S = \frac{2h\nu_0^3}{c^2}\,\frac{1}{\frac{N_1}{N_2}\,\frac{g_2}{g_1} - 1}$$
$$\iff \frac{N_1}{N_2} = \frac{g_1}{g_2}\left(1 + \frac{2h\nu_0^3}{c^2\,S}\right)$$

  We solve the following system :

$$\begin{cases} \dfrac{N_1}{N_2} = \dfrac{g_1}{g_2}\left(1 + \dfrac{2h\nu_0^3}{c^2\,S}\right) := t \\ N_1 = N_H - N_2 \end{cases} \tag{1.2.20}$$

So, the first equation of the system (1.2.20) gives :

$$N_2 = \frac{N_H}{1+t} \tag{1.2.21}$$

1.2.3 Implementation

1.2.3.1 Construction of the atmosphere and boundary conditions

1. Initial values : $T = 8000\ K$ (electron temperature), thickness=altitude=10000 km, $\nu_0 = \dfrac{c}{1215 \times 10^{-8}}\ s^{-1}$ or Hz ($L\alpha$ center frequency).

2. Position grid in cm : plane-parallel slab representing the filament is divided into $NZ = 91$. Let Z be altitude or position of the mesh in cm.

3. In the sake of simplicity and to focus on numerical method, we run PROM7 code for the following atmosphere model (filament) :

   $$T = 8000\ K,\ p = 0.1\ dyn.cm^{-2},\ V_T = 5\ km/s,\ \text{thickness=altitude} = 10000\ km,$$

   in order to obtain electron density (N_e) and hydrogen density (N_H), which are used in the computation of ϵ, B, $\overline{\kappa}$, ...

4. $N_H = N_1 + N_2$, $N_2 << N_1$

   - N_1 is used to calculate absorption coefficient $\overline{\kappa}$ (1.2.10). In the sake of simplicity, we suppose $N_1 = N_H$ but in fact, N_e and N_1 are calculated from pressure.
   - N_2 is used to compute coefficient a of VOIGT function.

5. Statistical weights of levels 1 and 2 (hydrogen) : $g_1 = 2$, $g_2 = 8$.

6. Computation of A_{21} using analytical formula from Johnson (1972) which is implemented in the numerical code as follows : $A_{21} = AEMS(1,2)$, where $AEMS$ is a function.

7. Computation of B_{21} and B_{12} from A_{21}, using formulas (1.2.8) and (1.2.9).

8. Computation of C_{12} and C_{21} : C_{12} is calculated from N_e and formula from Johnson (1972), which is represented in the numerical code by "function $CECH$". C_{12} is defined by :

   $$C_{12} = N_e\ CECH(1,2,T)\ exp\left(\frac{h\nu_0}{kT}\right)$$

   C_{21} is deduced from C_{12} by formula (1.2.14).

9. Computation of $\overline{\kappa}$ from formula (1.2.10), with $N_1 = N_H$.

10. Doppler width : $\Delta\nu_D = \dfrac{\nu_0}{c}\sqrt{V_T^2 + \dfrac{2kT}{m_H}}$, with m_H mass of hydrogen.

11. Computation of extinction coefficient ϵ according to formulas (1.2.16) and (1.2.19).

12. Optical depth grid τ : is calculated from grid of positions Z and from $\overline{\kappa}$. τ is given by the following formula :

$$\tau = \int_0^Z \overline{\kappa}(z')\, dz' \Longleftrightarrow \tau(Z) = \tau(Z-1) + \overline{\kappa}(Z)\,[Z(i) - Z(i-1)],$$

where i is the index in position grid Z.

13. Reduced frequency grid : $x = \dfrac{\nu - \nu_0}{\Delta\nu_D} = \dfrac{\Delta\nu}{\Delta\nu_D}$,
where $\Delta\nu = \nu - \nu_0$ is the relative frequency, and $\nu = \nu_0 + x\ \Delta\nu_D$ is the absolute frequency. Let xfr be this grid of reduced frequencies, of size nfr.

14. VOIGT profile : we use reduced frequencies defined by array xfr. VOIGT profile ϕ_ν is represented in the numerical code by "function VOIGT" (Humlicek, 1982; Hui, 1978):

$$\phi_\nu = VOIGT(a, xfr),$$

where a is VOIGT coefficient, defined below. ϕ_ν is a non-normalized profile whose :

- integral with respect to reduced frequencies is equal to $\sqrt{\pi}$,
- integral with respect to relative frequencies is equal $\sqrt{\pi}\ \Delta\nu_D$.

15. Computation of coefficient a (VOIGT) :

$$a = DFRCO + DFRNA,$$

where $DFRCO$ is the collisional broadening, computated by subroutine $ELCOH1$ in PROM7 but adapted to $L\alpha$ line. $DFRNA$ is the natural broadening, defined for $L\alpha$ line by $\dfrac{A_{21}}{4\pi}$.

16. Non-normalized profile with respect to reduced frequencies :

$$\phi_x = VOIGT(a, xfr)$$

17. Normalized profile with respect to relative frequency ($\Delta\nu = \nu - \nu_0$) :
ϕ_ν_normalized $= \dfrac{\phi_x}{aire \times \Delta\nu_D}$, with $area = \int \phi_x\ dx = \sqrt{\pi}$, x is reduced frequency

18. Optical depth at a given frequency ν : τ_ν is defined by

$$\tau_\nu = \int_0^\tau \phi_\nu_\text{normalized}(\tau')\, d\tau'$$
$$\Longleftrightarrow \tau_\nu(z) = \tau_\nu(z-1) + \frac{1}{2}\,[\phi_\nu_\text{normalized}(z-1) + \phi_\nu_\text{normalized}(z)] \times [\tau(z) - \tau(z-1)]$$

19. Integration weight W in relative frequency :
    - subroutine $TRAPEZ$ in numerical code computes integration weights M applied to xfr (reduced frequency).
    - we multiply M by the normalized profile ϕ_ν_normalized and we sum.
    - $W = \dfrac{M.\phi_\nu_\text{normalized}}{sum}$: normalized weight and proportional to normalized profile.

    Remark 1.2.2 *This integration weight W in used in the computation of $\overline{J} = \int J_\nu\, d\nu$.*

Remark 1.2.3 ⋆ *Reduced frequency x is only used when VOIGT function is called.*

⋆ *Most of the time, we work with relative frequencies : $\Delta\nu = \nu - \nu_0 = x\Delta\nu_D$.*

⋆ *We work with normalized profile with respect to relative frequency : ϕ_ν_normalized.*

20. Computation of dilution factor $FADIR$ for $L\alpha$ line (hydrogen) :
    $FADIR$ is obtained by subroutine $INTALT$ in PROM7, adapted to $L\alpha$. There is no limb-darkening in $L\alpha$ line.

21. Computation of Planck function B using formula (1.2.18).

22. Computation of lower BC (boundary condition) I_{inf} using subroutine $SOLINH$ in PROM7 adapted to $L\alpha$ line.
    - ⋆ We read input file of incident solar intensities "intinc.dat" of $L\alpha$ ($NFINT = 20$ values) : fequency in Å (1st column), intensity in $erg/cm^2/s/sr/$Å (2nd column).
    - ⋆ We multiply by $\dfrac{10^{-8}\ \nu_0^2}{c}$ frequencies read in order to obtain them in Hz.
    - ⋆ We multiply solar intensities by 10^{-8} to obtain them in $erg/cm^2/s/sr/Hz$.
    - ⋆ These solar intensities are then interpolated at relative frequencies ($xfr\ \Delta\nu_D$).
    - ⋆ Finally, we multiply them by $2 \times FADIR$ to obtain I_{inf}.

Remark 1.2.4 *C_{12}, C_{21}, B_{12}, B_{21}, A_{21} do not change during the radiative transfer process. They are fixed by the atmosphere model. The Planck function B is associated with $L\alpha$ line, is calculated once at the beginning.*

1.2.3.2 Method for solving Non-LTE RTE using accelerated ALI scheme

The algorithm using accelerated ALI method for solving RTE (1.1.1) or (1.1.7) is :

1. We calculate source function S by accelerated ALI method described in sections 1.1.4 and 1.1.5, from τ, B and ϵ.

2. When S has converged, we compute level populations N_1 and N_2.

Remark 1.2.5 *It is possible to compute N_1 and N_2 at each iteration of convergence as for S in order to prepare the numerical code for a multilevel atom (MALI). For MALI scheme, we don't calculate S but we iterate on level populations. We make convergence of all entities at the same time .*

Figures 1.9 and 1.10 represent source function as a function of optical depth (logarithmic scale), for a Doppler profile, respectively without acceleration (convergence is reached after 100 iterations) and with Ng acceleration (convergence is reached after 50 iterations).

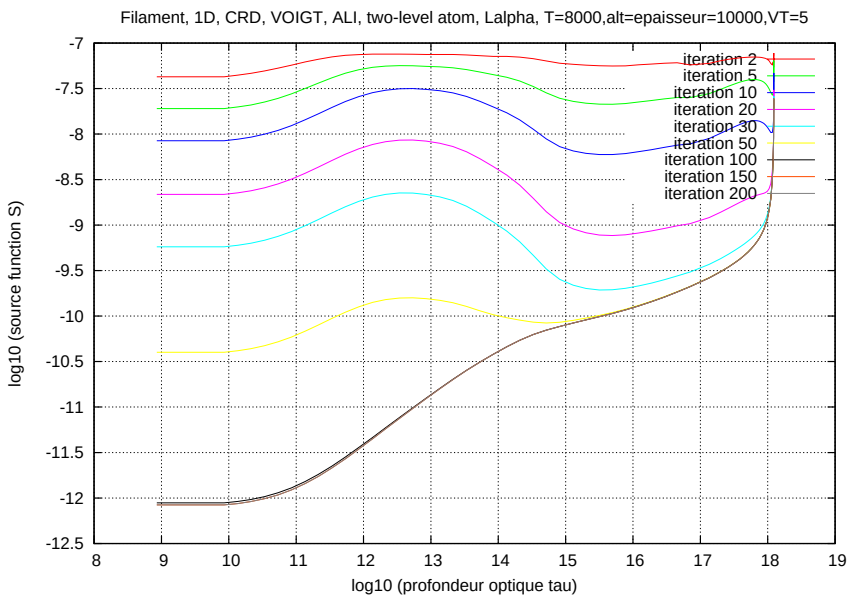

Figure 1.9: Source fonction as a function of optical depth (logarithmic scale), using ALI method, for a Doppler profile (case of a filament). Convergence is reached after 100 itérations.

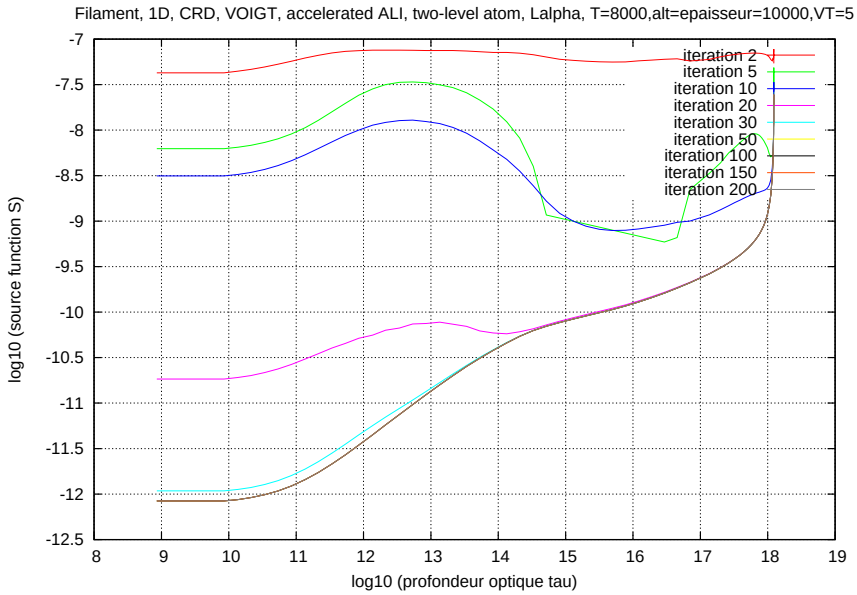

Figure 1.10: Source fonction as a function of optical depth (logarithmic scale), using accelerated ALI method, for a Doppler profile (case of a filament). Convergence is reached after 50 itérations.

1.2.4 Description of numerical code

Nature of the physical problem : NLTE radiative transfer (1D) in a filament for a two-level hydrogen atom (a fundamental and Lα line), without internal velocity field

Method of solution : improved Feautrier method (Rybicki and Hummer, 1991) combined with :
- ALI (Accelerated Lambda Iteration) method
- acceleration of ALI scheme using Ng method

Other relevant information : we use complete frequency redistribution (CRD). Electron and hydrogen densities are computed by PROM7 code

Authors : M. Chane-Yook & P. Gouttebroze

Program available from :
https://idoc.ias.u-psud.fr/MEDOC/Radiative transfer codes/Tools for radiative transfer

Computer(s) on which program hab been tested : PC with 4 Intel processors (2.67GHz)

Operating System(s) for which version of program has been tested : Linux

Programming language used : Fortran 90/95 (with **gfortran** compiler)

Status : stable

Accessibility : open (MEDOC)

No. of code lines in combined program and test deck : 1005

Typical running time : < 1 min for 50 iterations of accelerated ALI cycle

References :
- G. B. Rybicki & D. G. Hummer, "An accelerated lambda iteration method for multilevel radiative transfer. I- Non-overlapping lines with background continuum", A&A, 245, 171-181, 1991
- F. Paletou, "Transfert de rayonnement : méthodes itératives", C. R. Acad. Sci. Paris, t.2, Série IV, 885-898, 2001

In next section, we describe subroutines used in the program.

1.2.5 Description of subroutines

Structure of the program is the same as for a semi-infinite atmosphere but with additions of subroutines.
Unlike a semi-infinite atmosphere, in the case of a filament, constant parameters become arrays.

1.2.5.1 Set of variables used in the program

We describe below variables used mainly in module **param_mod** (param.f90 file) :

- RS=$6.96 \times 10^{1}0$: solar radius in cm
- h=6.6262×10^{-27} : Planck constant in cm^2 g/s
- ryd=2.17875×10^{-11} : Rydberg constant in erg
- bolt=$1.38064852 \times 10^{-6}$: Boltzmann constant in cm^2 g/s^2/K
- cl=2.997925×10^{10} : velocity of light in cm/s
- m_H=uma=1.660×10^{-24} : atomic mass unit of hydrogen
- nfr=20 : size of grid of reduced frequencies XFR

- n_grille=46 : size of grille array (see below)
- NZ=nxmod=91 : size of xmod array (see below). NZ=2× n_grille-1
- nmu=4 : size of direction grid ($\mu = \cos\theta$)
- coeff_extinction : array of size NZ corresponding to ϵ
- niter=200 : number of iterations for the convergence of accelerated ALI scheme
- T_in=8000 : temperature in K (model)
- EPST_in=10000 : filament thickness (model) in km
- VT_in=5×10^5 : microturbulence velocity in cm/s (model)
- altitude_in =10000 : filament altitude in km (model)
- NFINT=20 : size of FRFI and FINT arrays for reading solar incident intensities (input)
- I_sup=0 : upper boundary condition
- a_voigt : array of size NZ corresponding to parameter a of Voigt function. When $a = 10^{-3}$, we obtain Doppler function
- grille : array of size n_grille used to compute position grid
- position,VT : arrays of size NZ representing respectively position grid and microturbulence velocity
- xmod : array of size nxmod representing generic optical depth $\tau = (\tau_d)$
- xfr : array of size nfr representing reduced frequency x_i

- mu : array of size nmu representing values of $\mu = \cos\theta$ (direction)
- I_inf : array of size nfr representing incident intensities interpolated at reduced frequencies
- J_bar_ali : array of size nxmod representing $\overline{J}$
- J_nu_ali : array of size (nfr,nxmod) representing J_ν
- S_ali : array of size (nxmod,niter) representing source fonction S
- N_1, N_2 : arrays of size (nxmod,niter) representing level 1 (fundamental) and level 2 ($L\alpha$ line) populations, respectively
- u_ali : array of size (nxmod,nfr,nmu) representing solution of RTE for ALI method
- lambda_etoile : array of size nxmod representing Λ^* matrix
- lambda_etoile_mu_nu : array of size (nxmod,nfr,nmu) representing $\Lambda^*_{\mu\nu}$ matrix
- tau_nu : array of size (nxmod,nfr) representing optical depth at frequency ν : $\tau_\nu = \phi_\nu\,\tau$
- phi_nu : array of size (nfr,NZ) corresponding to Voigt profile ϕ_ν (non-normalized profile) for $L\alpha$ line
- phi_nu_normalise : array of size (nfr,NZ) corresponding to normalized profile with respect to relative frequencies
- M : array of size nfr representing intermediate integration weight with respect to frequency
- W : array of size nfr representing integration weight with respect to frequency, normalized and proportional to the profile phi_nu_normalise
- Ne, NH : arrays of size nxmod representing electron density and hydrogen density, respectively

- A21, B12, B21 : Einstein coefficients
- C12, C21 : arrays of size nxmod representing collisional excitation rate and collisional deexcitation rate, respectively
- lambda_0 = 1215 : $L\alpha$ center wavelength in Å
- gg1=1, gg2=8 : represent statistical weights of hydrogen levels 1 and 2, respectively
- kappa_bar : array of size NZ representing total absorption coefficient $\overline{\kappa}$
- T : array of size NZ representing temperature in K
- delta_nu_D : array of size NZ representing Doppler width
- DFRNA : refers to natural broadening of $L\alpha$ line
- DFRCO : array of size NZ representing collisional broadening of $L\alpha$ line
- nu_0=cl/(lambda_0$\times 10^{-8}$) : $L\alpha$ center frequency in s^{-1} or Hz
- coeff_Lalpha=10^{-8} : multiplicative coefficient in input file of solar incident intensities for $L\alpha$ line
- FADIR : dilution factor for $L\alpha$ line
- FRFI, FINT : arrays of size NFINT representing respectively frequencies (1st column) and incident intensities (2nd column) of input file "intensite_incidente_L_alpha"
- EXPHN=2.65, EXPS=2/3,PIA02=$\pi/2$, XMEL=9.10956×10^{-28}, CSE=8$\times$bolt/π/XMEL : constant values.

1.2.5.2 Description of subroutines in each module

Module general_mod (general.f90 file)

Module **general_mod** contains subroutines used for ALI method. Here are the modifications and/or additions of subroutines. The other subroutines remain unchanged compared to those of semi-infinite atmosphere.

Subroutine grilles : this subroutine sets up different grids for modeling a filament. More specifically,

1. electron (N_e) and hydrogen (N_H) densities obtained by PROM7 code are read
2. model parameters (temperature, microturbulence velocity) are input
3. we consider a general grid of size n_grille=46 which will be used as a basis to calculate position grid
4. we consider a position grid calculated from previous grid and from filament thickness
5. we compute atomic parameters : A_{21}, B_{21}, B_{12}, C_{21}, C_{12}
6. we compute ϵ
7. we compute Doppler width $\Delta\nu_D$ and total absorption coefficient $\overline{\kappa}$
8. we consider a generic optical depth grid τ in cm represented by xmod array of size nxmod
9. we consider a reduced frequency grid (x_i, i=1,..,nfr) represented by xfr array of size nfr
10. we compute natural broadening DFRNA of $L\alpha$ line, then collisional broadening DFRCO, which are used to calculate coefficient a (a_voigt in the code) of VOIGT
11. we choose Voigt profile as line profile ϕ_ν (at each frequency), which is non-normalized with respect to reduced frequencies. The profile is represented by phi_nu array
12. we compute $\int \phi_\nu \, d\nu$ (area by trapezoidal rule) in order to calculate normalized line profile with respect to relative frequency (array phi_nu_normalise in the code)
13. we consider optical depth grid at a given frequency ν : τ_ν
14. we consider a grid of directions $\mu = \cos\theta$ (array mu). μ must be different to 0
15. computation of weights in relative frequencies M using subroutine TRAPEZ (for the calculation of $\overline{J}$) : since ϕ_ν profile is not normalized, we compute weight W which is normalized and proportional to phi_nu_normalise
16. we compute dilution factor FADIR
17. we calculate lower boundary condition I_inf from incident intensities (subroutine SOLINH)

Subroutine elimination_gauss : This subroutine implements formulas for Gaussian elimination. One must pay attention to initialization of second member of the linear system (1.1.18) : the first component is initialized at $I_{sup} = 0$, the last component is initialized at I_{inf} (incident intensity calculated by SOLINH subroutine) at a given frequency, and the other components are initialized by source function.

Subroutine INTERL : it is a linear interpolation subroutine.

Subroutine calcul_coefficient_extinction : we compute ϵ using formula (1.2.19).

Subroutine integration_trapezes : this subroutine is used to calculate area using trapezoidal rule.

Subroutine SOLINH : this subroutine reads incident intensities input file "intensite_incidente_L_alpha". The 2 columns (there are NFINT=20 values for frequency and intensity) are multiplied by factors in order to have Hz and erg/cm^2/s/sr/Hz as units for frequency and intensity. Then an interpolation is made with respect to relative frequency $\Delta\nu = x\ \Delta\nu_D$ for the intensities (I_inf), which are multiplied by 2 times the dilution factor.

Function BENU : it is Planck function given by formula (1.2.18).

Subroutine INTALT : this subroutine computes dilution factor for $L\alpha$ line.

Function EXPINT : this function is used to compute collisional excitation rate.

Subroutine ELCOH1 : this subroutine computes collisional broadening DFRCO for $L\alpha$ line (hydrogen).

Function AEMS : AJI is obtained by AEMS function (in PROM7 code) which is computed from "Gaunt factors" (Table 1 from Johnson (1972)).

Function CECH : this function computes collisional excitation rates (coefficients) called S_e in the formulas below.
Let n and n' be two levels ($n < n'$). The excitation rate (coefficient) is given by the following formula (Johnson (1972)) :

$$\begin{aligned} S_e(n,n') = {} & (8kT/\pi m)^{1/2}\frac{2n^2}{x}\ \pi\ a_0^2\ y^2 \left(A_{nn'}\left[\left(\frac{1}{y}+\frac{1}{2}\right)E_1(y) - \left(\frac{1}{z}+\frac{1}{2}\right)E_1(z)\right]\right) \\ & + (8kT/\pi m)^{1/2}\frac{2n^2}{x}\ \pi\ a_0^2\ y^2 \left(\left[B_{nn'} - A_{nn'}\ \ln\frac{2n^2}{x}\right]\left[\frac{1}{y}E_2(y) - \frac{1}{z}E_2(z)\right]\right) \end{aligned} \tag{1.2.22}$$

$E_i(z) = \int_1^\infty e^{-zt}\ t^{-i}\ dt$ is called Exponential integral of order i ($i = 0, 1, 2, ...$).
m denotes the electron mass and $a_0 = 0.5292\ 10^{-8}\ cm$ the Bohr radius.
Here and below, E_n denotes level energy of n for hydrogen. We have :

$$
\begin{aligned}
y &= (E_{n'} - E_n)/kT, \\
z &= r_{nn'} + y, \\
x &= 1 - (n/n')^2, \\
B_{nn'} &= \frac{4n^4}{n'^3}\ x^{-2}\ (1 + \frac{4}{3}x^{-1} + b_n\ x^{-2}), \\
b_n &= n^{-1}(4 - 18.63n^{-1} + 36.24n^{-2} - 28.09n^{-3}), \quad n \geq 2, \\
A_{nn'} &= 2n^2\ x^{-1}\ f_{nn'}, \\
f_{nn'} &= \frac{32}{3\sqrt{3}\ \pi}\frac{n}{n'^3}\ x^{-3}\ g(n, x), \\
g(n, x) &= g_0(n) + g_1(n)\ x^{-1} + g_2(n)\ x^{-2}, \\
r_{nn'} &= r_n\ x, \\
r_n &= 1.94n^{-1.57}
\end{aligned}
$$

where g_0, g_1, g_2 are Gaunt factors for pour Bound-Free transitions, given in Table 1.1 (Table 1 from Johnson (1972)) :

	$n = 1$	$n = 2$	$n \geq 3$
$g_0(n)$	1.11330	1.0785	$0.9935 + 0.2328n^{-1} - 0.1296n^{-2}$
$g_1(n)$	-0.4059	-0.2319	$-n^{-1}(0.6282 - 0.5598n^{-1} + 0.5299n^{-2})$
$g_2(n)$	0.07014	0.02947	$n^{-2}(0.3887 - 1.181n^{-1} + 1.470n^{-2})$

Table 1.1: Gaunt factors.

Module ali_mod (fichier ali.f90)

This module contains several subroutines necessary for solving RTE using accelerated ALI method.

Subroutine methode_ali : accelerated ALI scheme remains unchanged compared to the case of a semi-infinite atmosphere except :
- at the beginning, we initialize source function to Planck function defined by BENU function (in the program)
- in the convergence iteration loop, level populations (subroutine calculation_population) are computed after source function computation.

Subroutine calcul_S_ali : in the case of a filament, extinction coefficient ϵ is not a fixed parameter but a vector. It is the same for Planck function.

Subroutine calcul_population : this subroutine was added to calculate populations of level 1 (N_1) and level 2 (N_2) using formulas (1.2.20) and (1.2.21).

1.2.6 Running numerical program

- make clean
- make
- ./lambda_it

Result files (fort.110, fort.101, fort.102, fort.113, fort.120) are read by **gnuplot** free software whose commands can be found in *methode_ali* subroutine.

1.3 Solar prominence

Non-LTE radiative transfer equation (RTE) is solved using accelerated ALI method for a solar prominence, in 1D and CRD (complete frequency redistribution) with Voigt profile. For this, we consider a realistic and finite vertical slab. More precisely a half-slab because it is symmetrical (radiation). We consider moreover a realistic two-level atom (hydrogen atom and $L\alpha$ line).
The only differences from a filament case are:

- half-slab instead of full-slab
- boundary conditions must be rewritten (tridiagonal matrix).

1.3.1 Prominence modeling

Picture 1.11 represents an erupting prominence observed by SDO/AIA (2012/08/30) at 304Å and 171Å wavelengths.

In the modeling (see figure 1.12), a prominence is represented by a plane-parallel slab with thickness e, standing vertically above the solar surface (at altitude h) and irradiated on both sides by the Sun. It is a 1D representation. Each side of this symmetrical model is illuminated by incident radiation from the photosphere, chromosphere and solar corona. This radiation field is very important since it determines the boundary conditions for solving RTE (1.1.1) or (1.1.7). Inside the prominence, the initial condition is defined by 3 physical parameters: electronic temperature, T, pressure, P, microturbulence velocity, V_T. Due to the symmetry of the problem, calculations can be done in a half slab

Formulas for a two-level atom and for statistical equilibrium are the same as for a filament. Implementation and setup of the atmosphere have the same structure as for a filament. Only boundary conditions differ from those of a filament.

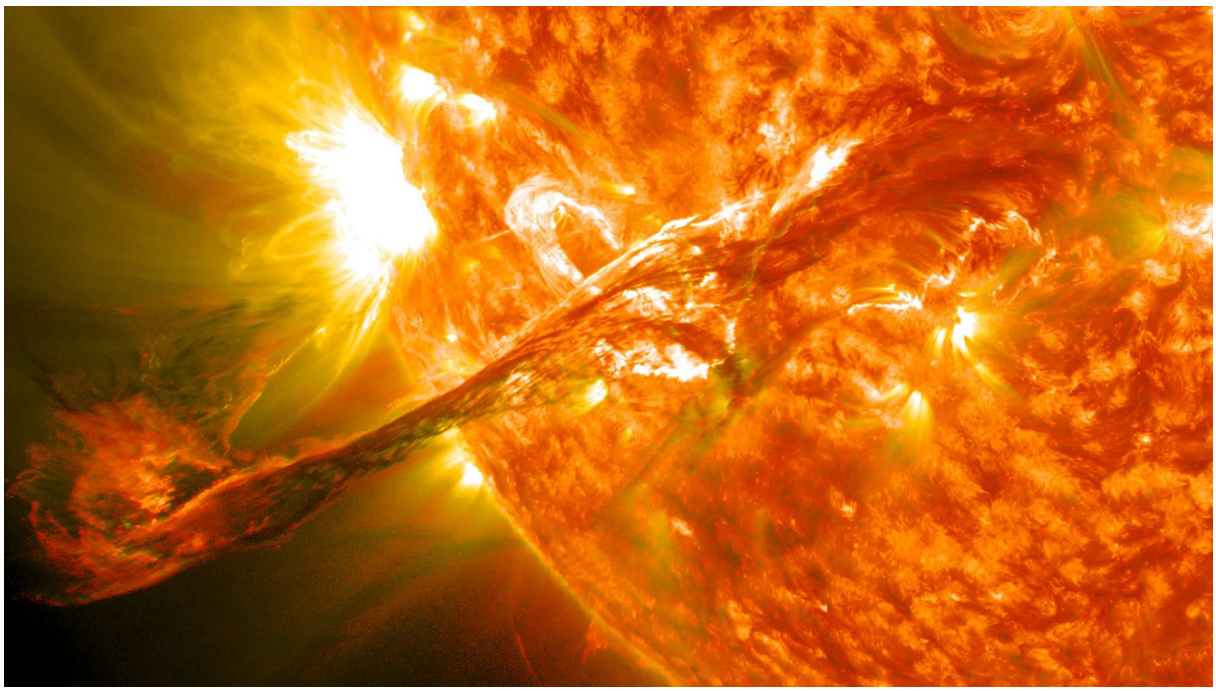

Figure 1.11: Erupting solar prominence observed by SDO/AIA, at 304Å and 171Å wavelengths, 2012/08/30.

1.3.2 Accelerated ALI method and boundary conditions

1.3.2.1 Boundary conditions

In the case of a prominence, instead of I_{inf} we have two boundary conditions :

1. on the surface of the prominence : it is the same boundary condition (I_{inf}) for the case of a filament but divided by 2 because we are only interested in half a slab here. So we have $I_{\text{sup}}(\tau = 0) = \frac{1}{2} \times$incident intensity in $L\alpha$ line of hydrogen. In other words, $I_{\text{sup}}(\text{prominence}) = \frac{1}{2} I_{\text{inf}}(\text{filament})$

2. in the center of the prominence : we start from discretized equations of section 1.1.2. We have :
$$I^+(\tau_{\text{max}}) = I^-(\tau_{\text{max}}), \tag{1.3.1}$$
with $I^-(\tau_{\text{max}}) = I(\tau_{\text{max}}, -\mu, \nu)$ and $I^+(\tau_{\text{max}}) = I(\tau_{\text{max}}, \mu, \nu)$.
We have $u = \frac{1}{2}(I_{\mu\nu} + I_{-\mu\nu})$ and $v = \frac{1}{2}(I_{\mu\nu} - I_{-\mu\nu})$ according to (1.1.8) and (1.1.9).
Then :
$$\begin{aligned} & u(\tau_{\text{max}}, \mu, \nu) = \frac{1}{2}\left[I(\tau_{\text{max}}, \mu, \nu) + I(\tau_{\text{max}}, -\mu, \nu)\right] \\ \Longleftrightarrow\ & u(\tau_{\text{max}}, \mu, \nu) = \frac{1}{2}\left[I^+(\tau_{\text{max}}) + I^-(\tau_{\text{max}})\right] \\ \Longleftrightarrow\ & u(\tau_{\text{max}}, \mu, \nu) = I^+(\tau_{\text{max}}), \quad \text{according to (1.3.1)} \end{aligned} \tag{1.3.2}$$

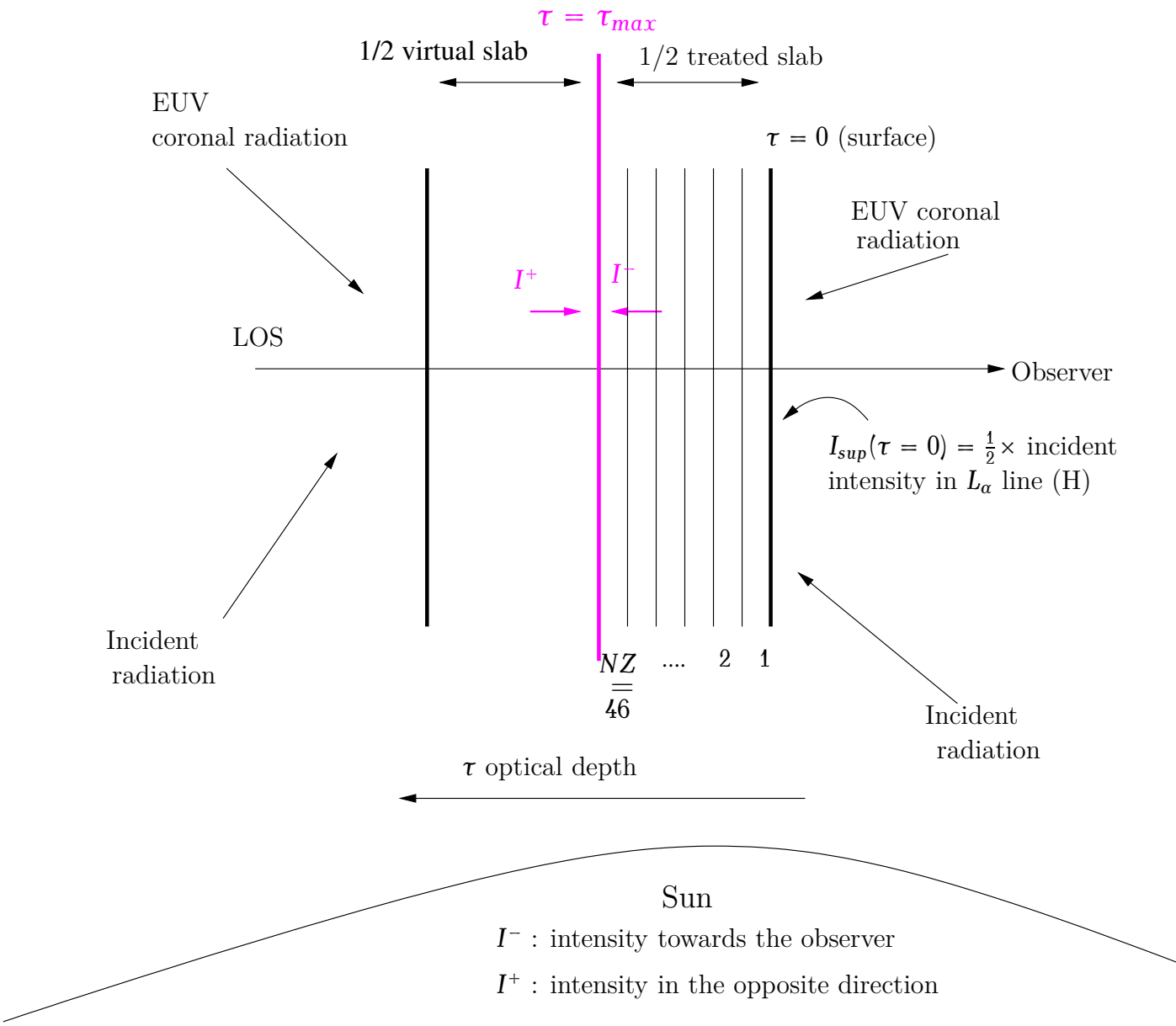

Figure 1.12: Prominence model (figure is not to scale).

And

$$v(\tau_{\max}, \mu, \nu) = 0 \tag{1.3.3}$$

According to (1.1.12), we have $\mu \dfrac{du}{d\tau_\nu} = v$. Then $\dfrac{du}{d\tau_\nu}(\tau_{\max}, \mu, \nu) = 0$ according to (1.3.2) and (1.3.3).

The discretization of order 1 gives : :

$$\begin{aligned} & u_{N_d} - u_{N_d-1} = 0 \\ & \Longleftrightarrow -A_{N_d-1}.u_{N_d-1} + B_{N_d}.u_{N_d} - C_{N_d+1}.u_{N_d+1} = E_{N_d}, \end{aligned}$$

with $A_{N_d-1} = 1$, $B_{N_d} = 1$, $E_{N_d} = 0$ (last coefficients of tridiagonal matrix and second member (see subroutine elimination_gauss)).

1.3.2.2 Method for solving Non-LTE RTE using accelerated ALI scheme

Accelerated ALI algorithm for solving RTE, and program structure for a prominence are the same as for a filament.

Remark 1.3.1 *$NZ = 46$ instead of 91 for the grid of positions.*

Figures 1.13 and 1.14 represent source function with respect to optical depth (logarithmic scale), for a Doppler profile, respectively without Ng acceleration (convergence reached after 50 iterations) and with Ng acceleration (convergence reached after 20 iterations). This is the representation of a half-slab.

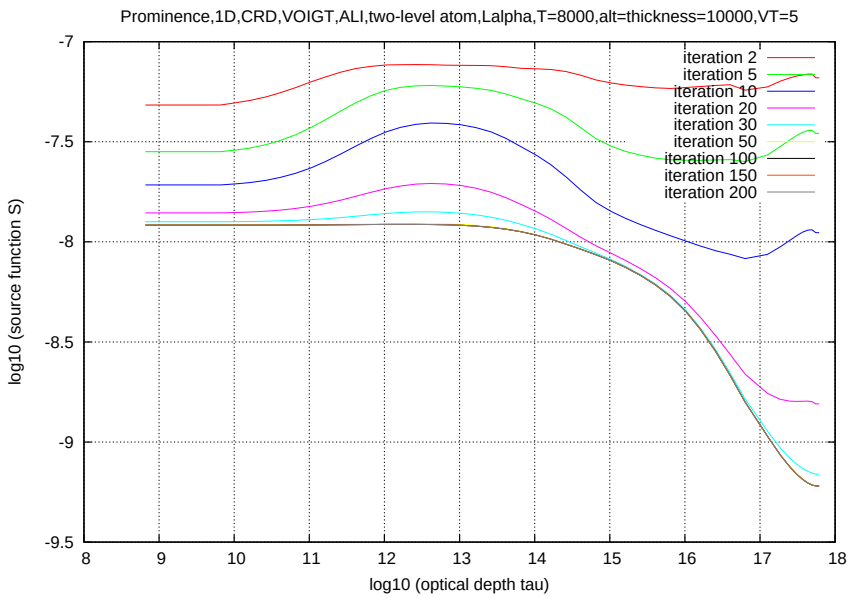

Figure 1.13: Source function with respect to optical depth (logarithmic scale), using ALI method for a Doppler profile (case of a prominence). Convergence is reached after 50 iterations.

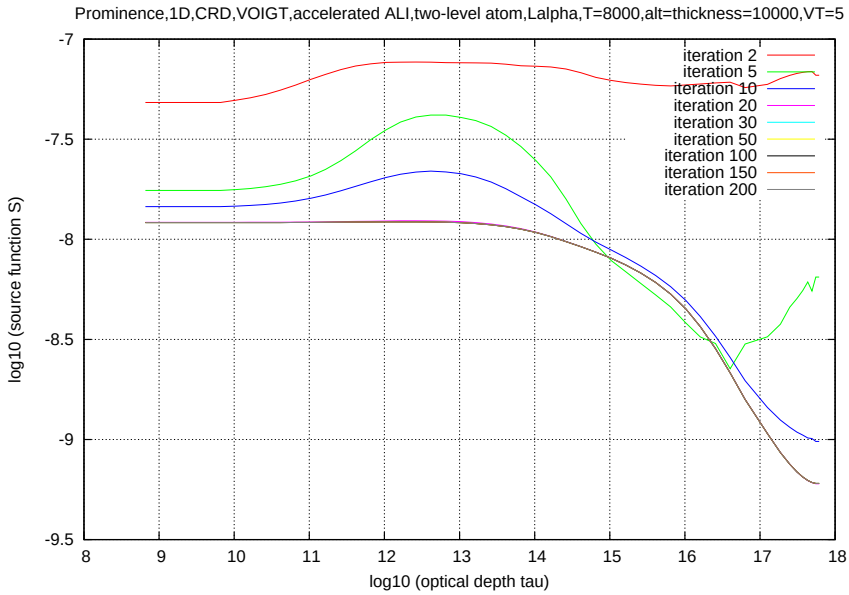

Figure 1.14: Source function with respect to optical depth (logarithmic scale), using accelerated ALI method for a Doppler profile (case of a prominence). Convergence is reached after 20 iterations.

1.3.3 Description of numerical code

Nature of the physical problem : NLTE radiative transfer (1D) in a prominence for a two-level hydrogen atom (a fundamental and Lα line), without internal velocity field

Method of solution : improved Feautrier method (Rybicki and Hummer, 1991) combined with :
- ALI (Accelerated Lambda Iteration) method
- acceleration of ALI scheme using Ng method

Other relevant information : we use complete frequency redistribution (CRD). Electron and hydrogen densities are computed by PROM7 code

Authors : M. Chane-Yook & P. Gouttebroze

Program available from :
https://idoc.ias.u-psud.fr/MEDOC/Radiative transfer codes/Tools for radiative transfer

Computer(s) on which program hab been tested : PC with 4 Intel processors (2.67GHz)

Operating System(s) for which version of program has been tested : Linux

Programming language used : Fortran 90/95 (with **gfortran** compiler)

Status : stable

Accessibility : open (MEDOC)

No. of code lines in combined program and test deck : 1014

Typical running time : < 1 min for 20 iterations of accelerated ALI cycle

References :
- G. B. Rybicki & D. G. Hummer, "An accelerated lambda iteration method for multilevel radiative transfer. I- Non-overlapping lines with background continuum", A&A, 245, 171-181, 1991
- F. Paletou, "Transfert de rayonnement : méthodes itératives", C. R. Acad. Sci. Paris, t.2, Série IV, 885-898, 2001

In next section, we describe subroutines used in the program.

1.3.4 Description of subroutines

Structure of the program is the same as for a solar filament but with modifications for boundary conditions.

1.3.4.1 Set of variables used in the program

We describe variables used in module **param_mod** (param.f90 file) :

- nfr=20 : size of reduced frequency grid XFR
- n_grille=46 : size of grille array (see below)
- NZ=nxmod=n_grille : size of xmod array (see below)
- nmu=4 : size of direction grid $\mu = \cos\theta$
- coeff_extinction : array of size NZ corresponding to ϵ
- niter=200 : number of iterations for the convergence of accelerated ALI scheme
- T_in=8000 : temperature in K (model)
- EPST_in=10000 : prominence thickness (model) in km

- VT_in=5×10^5 : microturbulence velocity in cm/s (model)
- altitude_in =10000 : prominence altitude in km (model)
- NFINT=20 : size of FRFI and FINT arrays for reading solar incident intensities (input)
- a_voigt : array of size NZ corresponding to parameter a of Voigt function. When $a = 10^{-3}$, we obtain Doppler function
- grille : array of size n_grille used to compute position grid
- position,VT : arrays of size NZ representing respectively position grid and microturbulence velocity
- xmod : array of size nxmod representing generic optical depth $\tau = (\tau_d)$
- xfr : array of size nfr representing reduced frequency x_i
- mu : array of size nmu representing values of $\mu = \cos\theta$ (direction)
- I_sup : array of size nfr representing incident intensities interpolated at reduced frequencies
- J_bar_ali : array of size nxmod representing $\overline{J}$
- J_nu_ali : array of size (nfr,nxmod) representing J_ν
- S_ali : array of size (nxmod,niter) representing source fonction S
- N_1, N_2 : arrays of size (nxmod,niter) representing level 1 (fundamental) and level 2 ($L\alpha$ line) populations, respectively
- u_ali : array of size (nxmod,nfr,nmu) representing the solution of RTE for ALI method
- lambda_etoile : array of size nxmod representing Λ^* matrix

- lambda_etoile_mu_nu : array of size (nxmod,nfr,nmu) representing $\Lambda^*_{\mu\nu}$ matrix
- tau_nu : array of size (nxmod,nfr) representing optical depth at frequency ν : $\tau_\nu = \phi_\nu\, \tau$
- phi_nu : array of size (nfr,NZ) corresponding to Voigt profile ϕ_ν (non-normalized profile) for $L\alpha$ line
- phi_nu_normalise : array of size (nfr,NZ) corresponding to normalized profile with respect to relative frequencies
- M : array of size nfr representing intermediate integration weight with respect to frequency
- W : array of size nfr representing integration weight with respect to frequency, normalized and proportional to the profile phi_nu_normalise
- Ne, NH : arrays of size nxmod representing electron density and hydrogen density, respectively
- A21, B12, B21 : Einstein coefficients
- C12, C21 : arrays of size nxmod representing collisional excitation rate and collisional deexcitation rate, respectively
- lambda_0 = 1215 : $L\alpha$ center wavelength in Å
- gg1=1, gg2=8 : represent statistical weights of hydrogen levels 1 and 2, respectively
- kappa_bar : array of size NZ representing total absorption coefficient $\overline{\kappa}$
- T : array of size NZ representing temperature in K
- delta_nu_D : array of size NZ representing Doppler width
- DFRNA : refers to natural broadening of $L\alpha$ line

- DFRCO : array of size NZ representing collisional broadening of $L\alpha$ line
- nu_0=cl/(lambda_0$\times 10^{-8}$) : $L\alpha$ center frequency in s^{-1} or Hz
- nu : array of size nfr representing absolute frequency $\nu = \nu_0 + xfr \times \Delta\nu_D$
- coeff_Lalpha=10^{-8} : multiplicative coefficient in input file of solar incident intensities for $L\alpha$ line
- FADIR : dilution factor for $L\alpha$ line
- FRFI, FINT : arrays of size NFINT representing respectively frequencies (1st column) and incident intensities (2nd column) of input file "intensite_incidente_L_alpha"
- EXPHN=2.65, EXPS=2/3,PIA02=$\pi/2$, XMEL=9.10956×10^{-28}, CSE=8$\times$bolt/π/XMEL : constant values

1.3.4.2 Description of subroutines in each module

Main modifications concern module **general_mod**.

Module general_mod (general.f90 file)

Module **general_mod** contains subroutines used for ALI method.

Subroutine grilles : this subroutine sets up different grids for modeling a prominence. More specifically,

1. electron (N_e) and hydrogen (N_H) densities obtained by PROM7 code are read
2. model parameters (temperature, microturbulence velocity) are input
3. we consider a general grid of size n_grille=46 which will be used as a basis to calculate position grid
4. we consider a position grid calculated from previous grid and from prominence thickness
5. we compute atomic parameters : A_{21}, B_{21}, B_{12}, C_{21}, C_{12}
6. we compute ϵ

7. we compute Doppler width $\Delta\nu_D$ and total absorption coefficient $\overline{\kappa}$
8. we consider a generic optical depth grid τ in cm represented by xmod array of size nxmod
9. we consider a reduced frequency grid (x_i, i=1,..,nfr) represented by xfr array of size nfr
10. we compute absolute frequencies $\nu = \nu_0 + xfr \times \Delta\nu_D$
11. we compute natural broadening DFRNA of $L\alpha$ line, then collisional broadening DFRCO, which are used to calculate coefficient a (a_voigt in the code) of VOIGT
12. we choose Voigt profile as line profile ϕ_ν (at each frequency), which is non-normalized with respect to reduced frequencies. The profile is represented by phi_nu array
13. we compute $\int \phi_\nu \, d\nu$ (area by trapezoidal rule) in order to calculate normalized line profile with respect to relative frequency (array phi_nu_normalise in the code)
14. we consider optical depth grid at a given frequency ν : τ_ν
15. we consider a grid of directions $\mu = \cos\theta$ (array mu). μ must be different to 0
16. computation of weights in relative frequencies M using subroutine TRAPEZ (for the calculation of $\overline{J}$) : since ϕ_ν profile is not normalized, we compute weight W which is normalized and proportional to phi_nu_normalise
17. we compute dilution factor FADIR
18. we compute upper boundary condition I_sup from incident intensities (subroutine SOLINH)

Subroutine elimination_gauss : This subroutine implements formulas for Gaussian elimination. One must pay attention to initialization of second member of the linear system (1.1.18) : the first component is initialized at I_{sup} at a given frequency (incident intensity calculated by SOLINH subroutine), the last component is zero, and the other components are initialized by source function.

Subroutine SOLINH : this subroutine reads incident intensities input file "intensite_incidente_L_alpha". The 2 columns (there are NFINT=20 values for frequency and intensity) are multiplied by factors in order to have Hz and erg/cm^2/s/sr/Hz as units for frequency and intensity. Then an interpolation is made with respect to relative frequency $\Delta\nu = x \; \Delta\nu_D$ for the intensities (I_sup), which are multiplied by dilution factor.

1.3.5 Running numerical program

- make clean
- make
- ./lambda_it

Result files (fort.100, fort.101, fort.102, fort.113) are read by **gnuplot** free software whose commands can be found in *methode_ali* subroutine.

1.4 Conclusion

In this chapter 1, 1D radiative transfer for a two-level atom, in the case of a solar semi-infinite atmosphere, a solar filament and a solar prominence, using complete frequency redistribution (CRD), is treated.

More precisely, transfer equation (1.1.7) is discretized by finite difference method to obtain a linear system (1.1.18) which is solved by Gaussian elimination (1.1.21)-(1.1.22), to determine intensity u. Two iterative schemes that update source function S step by step are implemented : Λ-iteration method and ALI method. Then, ALI scheme is accelerated by Ng method (see section 1.1.5), which consists in accelerating every third iteration of ALI convergence.

In the case of a semi-infinite atmosphere, for $B = 1$, $\epsilon = 10^{-4}$, we obtain the same results (figures 1.3 and 1.4) as Paletou (2001). This validates the results of iterative schemes. Λ-itération method still does not converge to the solution after 200 iterations (and after 1500 iterations). ALI method converges to the solution after 100 iterations and accelerated ALI method converges to the solution after 20 iterations.

Accelerated ALI scheme is then applied to solar filaments and prominences. In these cases, geometry of the problem is different.

In next chapter, we are interested in the case of a multilevel atom.

2

Case of a multilevel atom

Contents

2.1 Multilevel formulation 59
2.2 Iterative method and operator choice 64
2.3 Local operator with no background continuum (no continuum absorption). Application to a semi-infinite atmosphere 65
2.4 Local operator with background continuum (with continuum absorption). Application to a semi-infinite atmosphere) : planned . 78

2.1 Multilevel formulation

In this chapter, we are interested in self-consistent solution of statistical equilibrium equations with non-LTE radiative transfer equations corresponding to each 1D-treated transition for a multilevel atom. We consider the following assumptions :

- non-overlapping lines
- there is no ionization/recombination, more generally no ionization equilibrium : only transitions between 2 levels are considered (bound-bound transitions)
- with no background continuum (with no continuum absorption, see section 2.3), then with background continuum (with continuum absorption, see section 2.4)
- no internal velocity field : the structure is considered as static in the atmosphere but can be easily modified (see boundary conditions)
- complete frequency redistribution (CRD)

The treatment of multilevel lines is equivalent to simultaneously solving statistical equilibrium equations for level populations and each radiative transfer equation (RTE) for each transition.

In this section, we describe radiative transfer equations and population equations for the case of a multilevel atom (Rybicki and Hummer, 1991). In next section, we present the numerical method used (MALI: Multi Accelerated Lambda Iteration).

According to Rybicki and Hummer (1991), the Radiative Transfer Equation (RTE) is :

$$\mu \frac{\partial I_{\mu\nu}}{\partial z} = -\chi_{\mu\nu}.I_{\mu\nu} + \eta_{\mu\nu}, \tag{2.1.1}$$

where $I_{\mu\nu}$ is the specific intensity, $\chi_{\mu\nu}$ is the total opacity and $\eta_{\mu\nu}$ is the total emissivity at frequency ν and for angle θ.

An atom/ion is considered to have several $l, l', ...$ levels. Let n_l be the (density of) population of level l. Each level is characterized by its statistical weight g_l and its energy E_l.

Notations 2.1.1 *For example, if $l' < l$, then we have the following scheme :*
$E_{l'} < E_l \Longleftrightarrow l' < l$ *(see figure 2.1 below).*

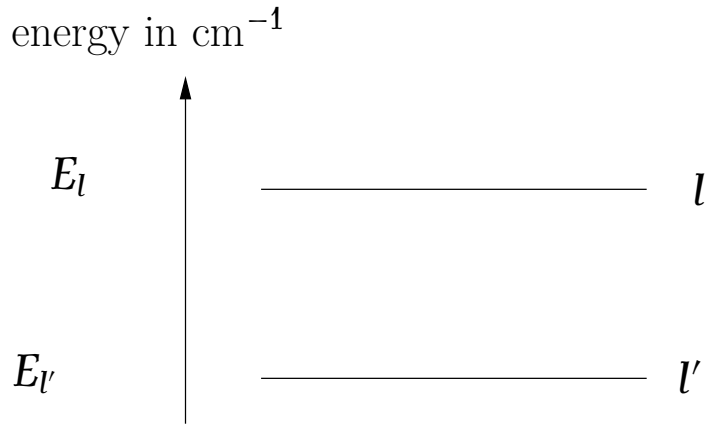

Figure 2.1: Levels l and l' of an atom/ion and their corresponding energy.

Let's consider transition between l and l' levels (see figure 2.1). Radiative properties of ll' line are characterized by its emissivity $\eta_{ll'}$ and its opacity $\chi_{ll'}$ (these quantities may depend on μ in the presence of velocity fields).

For $l > l'$, we have :

$$\begin{aligned} \eta_{ll'}(\mu, \nu) &= \frac{h\nu}{4\pi} n_l \; A_{ll'} \; \varphi_{ll'}(\mu, \nu) \\ \chi_{ll'}(\mu, \nu) &= \frac{h\nu}{4\pi} (n_{l'} \; B_{l'l} - n_l \; B_{ll'}) \; \varphi_{ll'}(\mu, \nu), \end{aligned} \tag{2.1.2}$$

$A_{ll'}$ (spontaneous emission), $B_{ll'}$ (stimulated emission) and $B_{l'l}$ (absorption) are the Einstein coefficients, and $\varphi_{ll'}(\mu, \nu)$ is the normalized line profile function.

Source function of the radiative transition ll' (i.e. line) is $(l > l')$:

$$S_{ll'} = \frac{n_l \, A_{ll'}}{n_{l'} \, B_{l'l} - n_l \, B_{ll'}} \tag{2.1.3}$$

Remark 2.1.1 *In CRD, $S_{ll'}$ is independent of frequency ν. We assume that radiation scattering on massive particles is isotropic, so $S_{ll'}$ is also independent of direction μ.*
In PRD (Partial Frequency Redistribution), $\varphi_{ll'} \neq \varphi_{l'l}$, so the source function depends on frequency ν.

Total emissivity $\eta_{\mu\nu}$ and total absorption coefficient $\chi_{\mu\nu}$ are :

$$\begin{aligned} \eta_{\mu\nu} &= \sum_{l>l'} \eta_{ll'}(\mu, \nu) + \eta_c(\nu) \\ \chi_{\mu\nu} &= \sum_{l>l'} \chi_{ll'}(\mu, \nu) + \chi_c(\nu), \end{aligned} \tag{2.1.4}$$

where $\eta_c(\nu)$ and $\chi_c(\nu)$ are the background emissivity and opacity.

Remark 2.1.2 *Total emissivity $\eta_{\mu\nu}$ and total absorption coefficient $\chi_{\mu\nu}$ are functions of n_i (population of levels).*

The total source function is then given by :

$$S_{\mu\nu} = \frac{\eta_{\mu\nu}}{\chi_{\mu\nu}} \tag{2.1.5}$$

The equations of statistical equilibrium for the ion populations may be written :

$$n_l \sum_{l'} (R_{ll'} + C_{ll'}) = \sum_{l'} n_{l'} \, (R_{l'l} + C_{l'l}), \tag{2.1.6}$$

where $C_{ll'}$ are the collisional rate coefficients and $R_{ll'}$ are the radiative rate coefficients, given by :

$$\begin{aligned} R_{ll'} &= A_{ll'} + B_{ll'} \, \bar{J}_{ll'}, \quad l > l' \ : \ \text{emission} \\ R_{ll'} &= B_{ll'} \, \bar{J}_{ll'}, \quad l < l' \ : \ \text{absorption}, \end{aligned} \tag{2.1.7}$$

where $\bar{J}_{ll'}$ is the integrated mean intensity (integrated on direction $\mu = \cos\theta$ for ll' line), defined by :

$$\bar{J}_{ll'} = \frac{1}{4\pi} \int d\Omega \int \varphi_{ll'}(\mu, \nu) \, I_{\mu\nu} \, d\nu \tag{2.1.8}$$

Thus, we may write population equations (2.1.6) in the convenient form (let's consider level l. Levels l' are the other levels of the atomic model) :

$$\begin{aligned}
&\sum_{l'<l} \left[n_l\ A_{ll'} - (n_{l'}\ B_{l'l} - n_l\ B_{ll'})\ \bar{J}_{ll'}\right] \\
&-\sum_{l'>l} \left[n_{l'}\ A_{l'l} - (n_l\ B_{ll'} - n_{l'}\ B_{l'l})\ \bar{J}_{ll'}\right] \\
&+\sum_{l'\neq l} (n_l\ C_{ll'} - n_{l'}\ C_{l'l}) = 0
\end{aligned} \tag{2.1.9}$$

And we use the following closure conservation equation :

$\sum_l n_l = N_T$, where N_T is the total population of the atomic element.

Proof of (2.1.9) :

Reminder 2.1.1 *Radiative and collisional processes (i.e. transitions between two levels) between two energy levels l and l' of an atom or ion are :*

- *spontaneous emission (of a photon) characterized by rate $A_{ll'}$*
- *stimulated emission characterized by coefficient $B_{ll'}.\bar{J}$*
- *radiative absorption characterized by coefficient $B_{l'l}.\bar{J}$ ($\bar{J}$ depends on radiation)*
- *collisional excitation characterized by coefficient $C_{l'l}$*
- *collisional deexcitation characterized by coefficient $C_{ll'}$.*

These processes are represented in figure 2.2.

We start from equation (2.1.6). Let's consider level l, l' being the other levels of atomic model.
The first left-hand side term in (2.1.6) can be written as (n_e is implicit in $C_{ll'}$) :

$$\begin{aligned}
n_l \sum_{l'} (R_{ll'} + C_{ll'}) &= \sum_{l<l'} n_l\ R_{ll'} + \sum_{l>l'} n_l\ R_{ll'} + \sum_{l'\neq l} n_l\ C_{ll'} \quad \text{according to (2.1.7)} \\
&= \sum_{l<l'} n_l\ B_{ll'}\ \bar{J}_{ll'} + \sum_{l>l'} n_l\ (A_{ll'} + B_{ll'}\ \bar{J}_{ll'}) + \sum_{l'\neq l} n_l\ C_{ll'} \\
&= \sum_{l'>l} n_l\ B_{ll'}\ \bar{J}_{ll'} + \sum_{l'<l} n_l\ A_{ll'} + \sum_{l'<l} n_l\ B_{ll'}\ \bar{J}_{ll'} + \sum_{l'\neq l} n_l\ C_{ll'}
\end{aligned} \tag{2.1.10}$$

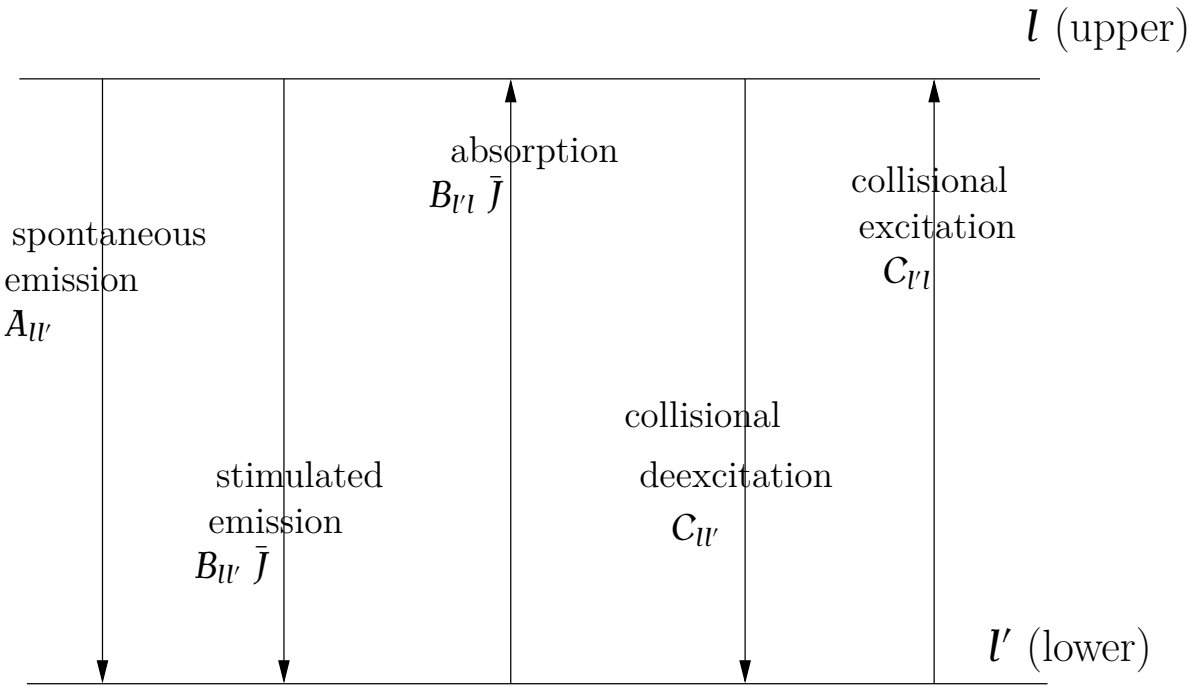

Figure 2.2: Different radiative and collisional processes between two energy levels l and l' ($l' < l$) of an atom or ion.

The second right-hand side term in (2.1.6) can be written as :

$$\sum_{l'} n_{l'}\ (R_{l'l} + C_{l'l}) = \sum_{l'<l} n_{l'}\ R_{l'l} + \sum_{l'>l} n_{l'}\ R_{l'l} + \sum_{l'\neq l} n_{l'}\ C_{l'l} \tag{2.1.11}$$

In CRD we have $\bar{J}_{ll'} = \bar{J}_{l'l}$. According to (2.1.7), the right-hand side term in (2.1.11) can be written as :

$$\sum_{l'<l} n_{l'}\ B_{l'l}\ \bar{J}_{ll'} + \sum_{l'>l} n_{l'}\ A_{l'l} + \sum_{l'>l} n_{l'}\ B_{l'l}\ \bar{J}_{ll'} + \sum_{l',l'\neq l} n_{l'}\ C_{l'l} \tag{2.1.12}$$

Thus, population equations (2.1.6) are (we consider level l, l' being the other levels) :

$$\sum_{l'>l} n_l\ B_{ll'}\ \bar{J}_{ll'} + \sum_{l'<l} n_l\ A_{ll'} + \sum_{l'<l} n_l\ B_{ll'}\ \bar{J}_{ll'} + \sum_{l',l'\neq l} n_l\ C_{ll'} - \sum_{l'<l} n_{l'}\ B_{l'l}\ \bar{J}_{ll'}$$

$$- \sum_{l'>l} n_{l'}\ A_{l'l} - \sum_{l'>l} n_{l'}\ B_{l'l}\ \bar{J}_{ll'} - \sum_{l',l'\neq l} n_{l'}\ C_{l'l} = 0$$

By grouping the different terms together, we obtain (2.1.9). ■

2.2 Iterative method and operator choice

2.2.1 MALI method

The MALI scheme to be introduced here is based on the operator splitting technique given in equation (1.2) from Rybicki and Hummer (1991) :

$$I_{\mu\nu} = \Lambda^*_{\mu\nu}[S_{\mu\nu}] + (\Lambda_{\mu\nu} - \Lambda^*_{\mu\nu})[S^\dagger_{\mu\nu}], \tag{2.2.1}$$

Equation (2.2.1) gives $I_{\mu\nu}$ in terms of the populations. It is important to note that $S^\dagger_{\mu\nu}$ as well as the operators $\Lambda_{\mu\nu}$ and $\Lambda^*_{\mu\nu}$ are constructed from the old populations $n^\dagger_l$ computed at the previous iteration.

The new populations n_l enter only though the source function $S_{\mu\nu}$ which is expressed in terms of them by way of equations (2.1.2), (2.1.4) and (2.1.5).

If this form for $I_{\mu\nu}$ (2.2.1) is substituted into equation (2.1.8) and the resulting form for $\bar{J}_{ll'}$ is substituted into equations (2.1.9), then the statistical equilibrium equations are expressed solely in terms of knows quantities and the new populations n_l (equations 2.3.8).

The MALI iterative scheme is :

1. initial choice for the old populations $n^\dagger_l$
2. set up the equations of populations (2.1.9). Then we solve the preconditionned equations of statistical equilibrium (2.3.8) in order to obtain new populations n_l
3. these new n_l populations from step 2 become old populations for the next iteration, etc. This iterative process continues until convergence is reached.

2.2.2 Choice of approximate lambda operator $\Lambda^*_{\mu\nu}$

We note that in practice the so-called exact lambda operator $\Lambda_{\mu\nu}$ is itself an approximation based on some discretization of the problem (1.1.7) in space (see section 1.1.2 of chapiter 1 "Case of a two-level atom"), and it appears as a matrix operator (inverse of the matrix T (1.1.19)) acting on the values of the source function ((1.1.18) and (1.1.20)) at the chosen discrete spatial grid.

One of the simplest choices for an approximation operator $\Lambda^*_{\mu\nu}$ is to take the diagonal part of this full matrix operator $\Lambda_{\mu\nu}$, this is the choice made in the previous chapter for the case of a two-level atom and also by Olson et al. (1986). $\Lambda^*_{\mu\nu}$ is recalculed every time when the populations n_i are updated (unlike the case of a two-level atom). Other possible choices are: the tridiagonal part of the full operator or an even wider band than the tridiagonal.

The principal advantage of the diagonal approximation is that the equations of statistical equilibrium remain completely local, whereas more sophisticated band approximations, such

as tridiagonal, introduce nonlocalness into these equations, which are then harder to solve, and may be more unstable.

We now derive linear preconditioned equations of statistical equilibrium for a multilevel line problem, using a local operator (i.e. we choose the diagonal approximation $\Lambda^*_{\mu\nu}$) in the following cases :

- with no background continuum (i.e. with no continuum absorption) and application to semi-infinite atmosphere (see section 2.3)
- with background continuum (i.e. with continuum absorption) and application to semi-infinite atmosphere (see section 2.4).

It is important to preserve linearity of original equations of the statistical equilibrium, as long as the electronic density n_e is known a priori.

2.3 Local operator with no background continuum (no continuum absorption). Application to a semi-infinite atmosphere

2.3.1 Statistical equilibrium equations

Since the lines are assumed non-overlapping, in the neighborhood of line ll' we have $S_{\mu\nu} = S_{ll'}$, which is frequency-independent.
Using this in equation (2.2.1) we obtain, for frequencies near the line :

$$I_{\mu\nu} = \Lambda^*_{\mu\nu}\, S_{ll'} + I^{eff}_{\mu\nu}, \tag{2.3.1}$$

where

$$I^{eff}_{\mu\nu} = \Lambda_{\mu\nu}\,[S^{\dagger}_{ll'}] - \Lambda^*_{\mu\nu}\, S^{\dagger}_{ll'} := I^{\dagger}_{\mu\nu} - \Lambda^*_{\mu\nu}\, S^{\dagger}_{ll'} \quad \text{according to (1.1.1)} \tag{2.3.2}$$

Here, $I^{\dagger}_{\mu\nu}$ is the radiation field that one gets from the formal solution with the old populations.

Remark 2.3.1 *As the operator is local here (it is the diagonal of $\Lambda_{\mu\nu}$), the product of a matrix by a vector is a simple multiplication between terms of source function and Λ matrix. That's why brackets [ ] are omitted in the expression of $\Lambda^*_{\mu\nu}[S_{\mu\nu}] = \Lambda^*_{\mu\nu}\, S_{ll'}$.*

Substituting (2.3.1) into expression of the integrated mean intensity $\bar{J}_{ll'}$ (2.1.8), we have :

$$\bar{J}_{ll'} = \bar{\Lambda}^*_{ll'}\, S_{ll'} + \bar{J}^{eff}_{ll'}, \tag{2.3.3}$$

with

$$\bar{\Lambda}^*_{ll'} = \frac{1}{4\pi} \int d\Omega \int \varphi_{ll'} \, \Lambda^*_{\mu\nu} \, d\nu, \tag{2.3.4}$$

and

$$\bar{J}^{eff}_{ll'} = \frac{1}{4\pi} \int d\Omega \int \varphi_{ll'} \, I^{eff}_{\mu\nu} \, d\nu = \bar{J}^{\dagger}_{ll'} - \bar{\Lambda}^*_{ll'} \, S^{\dagger}_{ll'} \tag{2.3.5}$$

are angle and frequency averages of $\Lambda_{\mu\nu}$ and $I^{eff}_{\mu\nu}$, using $\varphi_{ll'}$ as a weighting function.

The quantity $\bar{J}^{\dagger}_{ll'}$ defined below, is the value of the integrated mean intensity obtained by integrating over the old radiation field at previous iteration :

$$\bar{J}^{\dagger}_{ll'} = \frac{1}{4\pi} \int d\Omega \int \varphi_{ll'} \, I^{\dagger}_{\mu\nu} \, d\nu \tag{2.3.6}$$

Proof of (2.3.4)-(2.3.6) :

According to (2.3.1) and (2.3.2), equation (2.1.8) can be written as :

$$\begin{aligned} \bar{J}_{ll'} &= \frac{1}{4\pi} \int d\Omega \int \varphi_{ll'}(\mu,\nu) \, \Lambda^*_{\mu\nu} \, S_{ll'} \, d\nu + \frac{1}{4\pi} \int d\Omega \int \varphi_{ll'}(\mu,\nu) \, I^{eff}_{\mu\nu} \, d\nu \\ &= A + B \end{aligned} \tag{2.3.7}$$

Since $S_{ll'}$ is independent of the frequency ν, the first term A is written as :

$$\begin{aligned} A &= \frac{1}{4\pi} S_{ll'} \int d\Omega \int \varphi_{ll'}(\mu,\nu) \, \Lambda^*_{\mu\nu} \, d\nu \\ &= \frac{1}{4\pi} S_{ll'} \, \bar{\Lambda}^*_{ll'}, \quad \text{avec } \bar{\Lambda}^*_{ll'} = \frac{1}{4\pi} \int d\Omega \int \varphi_{ll'} \, \Lambda^*_{\mu\nu} \, d\nu. \end{aligned}$$

Hence (2.3.4). ■

The second term B is $\bar{J}^{eff}_{ll'}$:

$$\begin{aligned} B &= \frac{1}{4\pi} \int d\Omega \int \varphi_{ll'}(\mu,\nu) \, I^{eff}_{\mu\nu} \, d\nu = \bar{J}^{eff}_{ll'} \quad \text{by definition} \\ &= \frac{1}{4\pi} \int d\Omega \int \varphi_{ll'}(\mu,\nu) \, I^{\dagger}_{\mu\nu} \, d\nu - \frac{1}{4\pi} \int d\Omega \int \varphi_{ll'}(\mu,\nu) \, \Lambda^*_{\mu\nu} \, S^{\dagger}_{ll'} \, d\nu \quad \text{according to (2.3.2)} \\ &= \bar{J}^{\dagger}_{ll'} - \Lambda^*_{ll'} \, S^{\dagger}_{ll'} \end{aligned}$$

Hence (2.3.6) and (2.3.5). ■

Subtituting these results in population equations (2.1.9), with the use of ll' line source function expression (2.1.3), we obtain the preconditioned equations of statistical equilibrium for level l (l' being the other levels) :

$$\begin{aligned}
&\sum_{l'<l} [n_l\ A_{ll'}(1-\bar{\Lambda}^*_{ll'}) - (n_{l'}\ B_{l'l} - n_l\ B_{ll'})\ \bar{J}^{eff}_{ll'}] \\
&-\sum_{l'>l} [n_{l'}\ A_{l'l}(1-\bar{\Lambda}^*_{ll'}) - (n_l\ B_{ll'} - n_{l'}\ B_{l'l})\ \bar{J}^{eff}_{ll'}] \\
&+\sum_{l',l'\neq l} (n_l\ C_{ll'} - n_{l'}\ C_{l'l}) = 0
\end{aligned} \tag{2.3.8}$$

Proof of (2.3.8) :

According to (2.1.3) and (2.3.3), equations of statistical equilibrium (2.1.9) are :

$$\begin{aligned}
&\sum_{l'<l} [n_l\ A_{ll'} - (n_{l'}\ B_{l'l} - n_l\ B_{ll'})\ \bar{J}_{ll'}] \\
&-\sum_{l'>l} [n_{l'}\ A_{l'l} - (n_l\ B_{ll'} - n_{l'}\ B_{l'l})\ \bar{J}_{ll'}] \\
&+\sum_{l',l'\neq l} (n_l\ C_{ll'} - n_{l'}\ C_{l'l}) = 0 \\
&\Longleftrightarrow \sum_{l'<l} U - \sum_{l'>l} V + \sum_{l',l'\neq l} (n_l\ C_{ll'} - n_{l'}\ C_{l'l}) = 0
\end{aligned} \tag{2.3.9}$$

Let $U = n_l\ A_{ll'} - (n_{l'}\ B_{l'l} - n_l\ B_{ll'})\ \bar{J}_{ll'}, \quad l' < l$

$= n_l\ A_{ll'} - (n_{l'}\ B_{l'l} - n_l\ B_{ll'})\ \bar{\Lambda}^*_{ll'}\ S_{ll'} - (n_{l'}\ B_{l'l} - n_l\ B_{ll'})\ \bar{J}^{eff}_{ll'}$ according to (2.3.3)

$= U_1 - (n_{l'}\ B_{l'l} - n_l\ B_{ll'})\ \bar{J}^{eff}_{ll'}$

And $U_1 = n_l\ A_{ll'} - (n_{l'}\ B_{l'l} - n_l\ B_{ll'})\ \bar{\Lambda}^*_{ll'}\ S_{ll'}$

$= n_l\ A_{ll'} - (n_{l'}\ B_{l'l} - n_l\ B_{ll'})\ \bar{\Lambda}^*_{ll'}\ \dfrac{n_l\ A_{ll'}}{n_{l'}\ B_{l'l} - n_l\ B_{ll'}}$ according to (2.1.3)

$= n_l\ A_{ll'} - n_l\ A_{ll'}\ \bar{\Lambda}^*_{ll'}$

$= n_l\ A_{ll'}\ (1 - \bar{\Lambda}^*_{ll'})$

Then,

$$U = n_l\ A_{ll'}\ (1 - \bar{\Lambda}^*_{ll'}) - (n_{l'}\ B_{l'l} - n_l\ B_{ll'})\ \bar{J}^{eff}_{ll'}, \quad l' < l$$

Let

$$V = n_{l'}\ A_{l'l} - (n_l\ B_{ll'} - n_{l'}\ B_{l'l})\ \bar{J}_{ll'}, \quad l' > l$$

$$= n_{l'}\ A_{l'l} - (n_l\ B_{ll'} - n_{l'}\ B_{l'l})\ \bar{\Lambda}^*_{ll'}\ S_{l'l} - (n_l\ B_{ll'} - n_{l'}\ B_{l'l})\ \bar{J}^{eff}_{ll'} \quad \text{according to (2.3.3)}$$

$$= n_{l'}\ A_{l'l} - W - (n_l\ B_{ll'} - n_{l'}\ B_{l'l})\ \bar{J}^{eff}_{ll'}$$

It is sufficient to prove that $W = n_{l'}\ A_{l'l}\ \bar{\Lambda}^*_{ll'}$ for $l' > l$:

$$W = (n_l\ B_{ll'} - n_{l'}\ B_{l'l})\ \bar{\Lambda}^*_{ll'}\ S_{l'l}$$

$$= (n_l\ B_{ll'} - n_{l'}\ B_{l'l})\ \bar{\Lambda}^*_{ll'}\ \frac{n_{l'}\ A_{l'l}}{n_l\ B_{ll'} - n_{l'}\ B_{l'l}} \quad \text{according to (2.1.3)}$$

$$= n_{l'}\ A_{l'l}\ \bar{\Lambda}^*_{ll'}, \quad l' > l$$

Thus,

$$V = n_{l'}\ A_{l'l}\ (1 - \bar{\Lambda}^*_{ll'}) - (n_l\ B_{ll'} - n_{l'}\ B_{l'l})\ \bar{J}^{eff}_{ll'}$$

Hence (2.3.8). ■

Remark 2.3.2 *The result of these substitutions is to leave the form of statistical equilibrium equations the same as before (2.1.9), except that the Einstein A-coefficient (i.e. $A_{ll'}$ or $A_{l'l}$) has been multiplied by the factor $(1 - \bar{\Lambda}^*_{ll'})$ and the integrated mean intensity $\bar{J}_{ll'}$ is now replaced by $\bar{J}^{eff}_{ll'}$.*

These preconditioned statistical equilibrium equations are clearly still linear (if the electron density n_e is known a priori) in the ion/atom populations.

Another desirable feature of these modified equations (2.3.8) is that they automatically guarantee non-negative solutions for the new populations. This property follows the non-negativity of the modified rate coefficients.

Finally, it is important for the preceding argument that the approximate operator $\Lambda^*_{\mu\nu}$ be the diagonal of the exact $\Lambda_{\mu\nu}$ to obtain $(\Lambda_{\mu\nu} - \Lambda^*_{\mu\nu}) > 0$.

2.3.2 Description of H3CRD program

Nature of the physical problem : self-consistent solution in 1D of statistical equilibrium equations with NLTE radiative transfer equations corresponding to each treated transition for hydrogen atom (3 levels), for a semi-infinite atmosphere, using complete frequency redistribution (CRD), without internal velocity field

Method of solution : MALI (Multi Accelerated Lambda Iteration) method and short characteristics method

Other relevant information : ionization equilibrium in not included. All transitions are radiatively permitted. Boundary conditions are monochromatic. We use the (old) benchmark from Avrett (1968).

Author : F. Paletou (IRAP)

Program available from :
https://idoc.ias.u-psud.fr/MEDOC/Radiative transfer codes/Tools for radiative transfer

Computer(s) on which program hab been tested : PC with 4 Intel processors (2.67GHz)

Operating System(s) for which version of program has been tested : Linux

Programming language used : rewritten in Fortran 90 by M. Chane-Yook (**gfortran** compiler)

Status : stable

Accessibility : open (MEDOC)

No. of code lines in combined program and test deck : 781

Typical running time : < 1 min for 50 iterations of MALI cycle

References :

- G. B. Rybicki & D. G. Hummer, "An accelerated lambda iteration method for multilevel radiative transfer. I- Non-overlapping lines with background continuum", A&A, 245, 171-181, 1991

- F. Paletou, "Transfert de rayonnement : méthodes itératives", C. R. Acad. Sci. Paris, t.2, Série IV, 885-898, 2001

- E. H. Avrett, "Resonance lines in Astrophysics", NCAR, 1968.

2.3.3 Algorithm

Figure 2.3 describes the algorithm of H3CRD code.

2.3.4 Atomic structure of hydrogen

We consider here 3 levels for hydrogen (see figure 2.4). More specifically, we consider absorption and emission transitions :

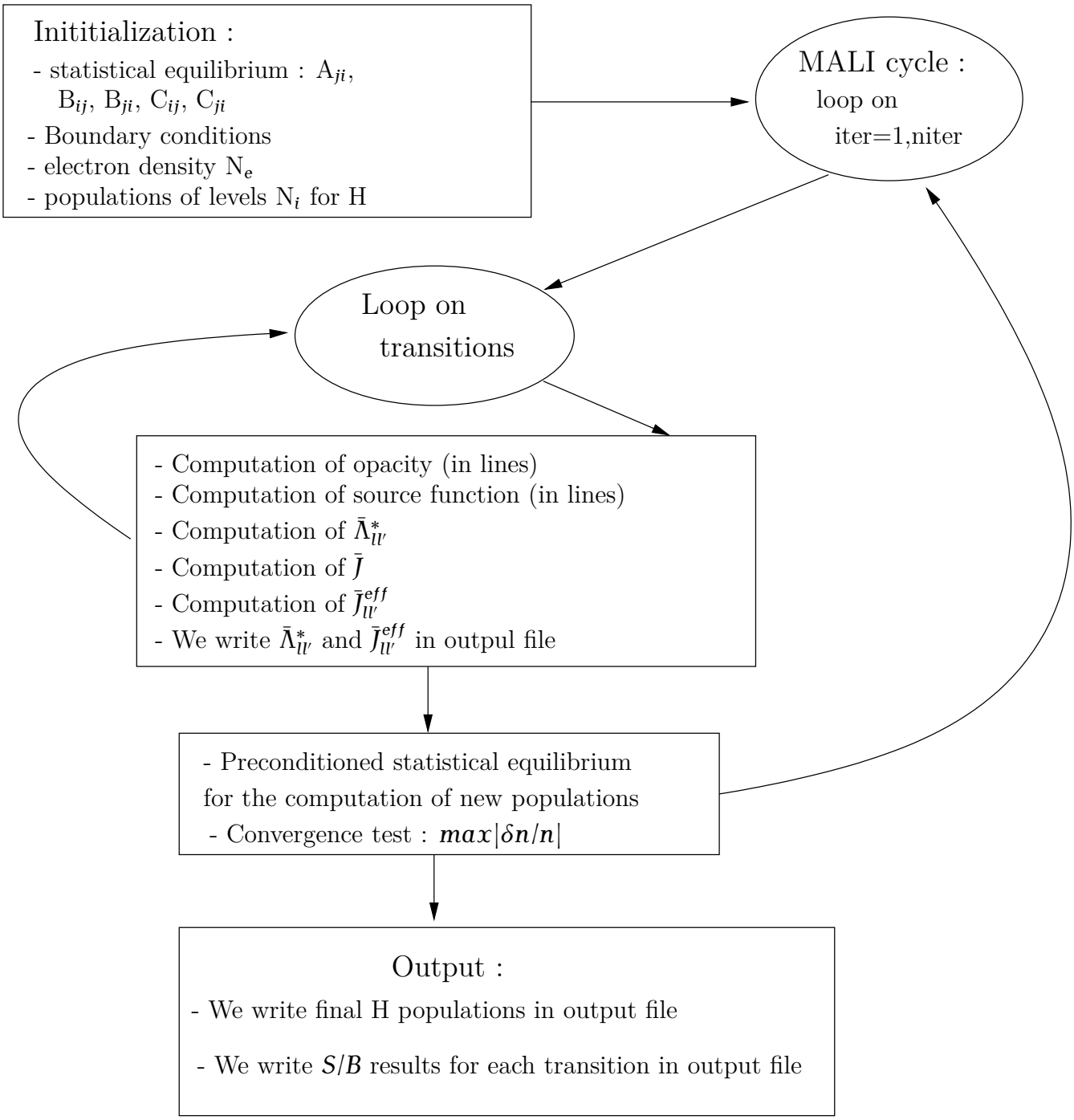

Figure 2.3: Algorithm of H3CRD code.

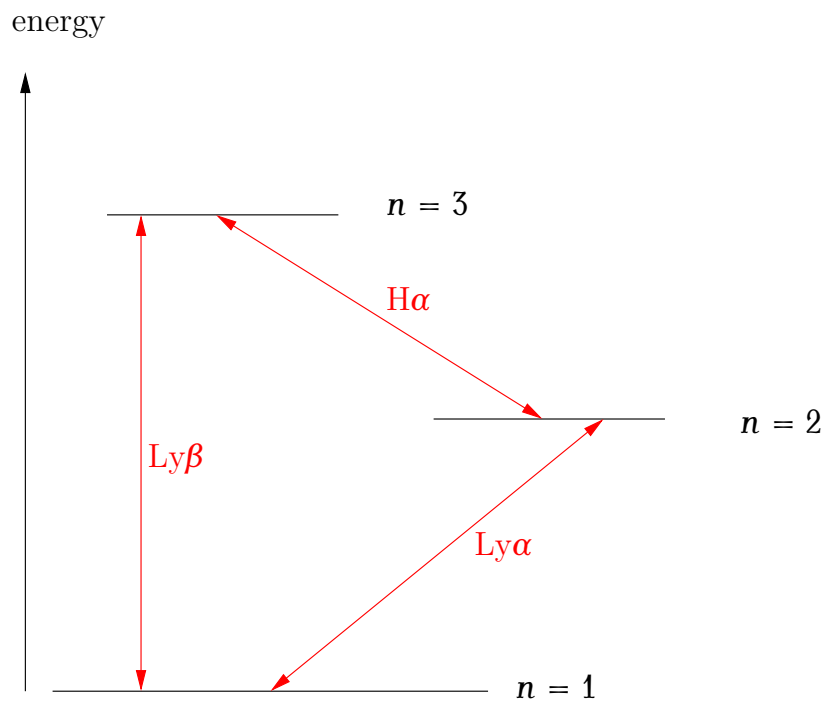

Figure 2.4: Hydrogen energy diagram (figure is not to scale) - 3 radiative transitions (in red).

- 1 to 2 : Lyα
- 1 to 3 : Lyβ
- 3 to 2 : Hα

For other simplified atoms, refer to Avrett (1968).

2.3.5 Description of subroutines

H3CRD code is written in fortran 90. The main program "h3crd.f90" calls 3 subroutines :

- **grilles** : setting up of frequency grid, optical depth grid, direction grid, grid for statistical weights of levels, angular quadrature weight grid, Einstein coefficients A and B, collisional coefficients and Gauss profile (for each line).

- **initialisation_population** : estimation of level 1 population (n_1) from χ_{21}, computation of the other population densities at LTE, calculation of electron density n_e assuming that each optical depth is initialized in the same way, computation of C_{ij} from populations at LTE.

- **cycle_mali** : 1) formal solution of RTE (using short characteristics method). 2) coupling with statistical equilibrium equations (using MALI method).

The main program "h3crd.f90" uses several modules whose files are (see the following section 2.3.5.1 for the set of variables used) :

- **param.f90** : contains global variables as well as constants defined as $nlev$, $nfreq$, $ndir$, ...

- **general.f90** : contains several subroutines such as computation of grids, initialization of populations, computation of B Einstein coefficients, computation of Doppler width, definition of Planck function (lower boundary condition; no illumination on the surface), computation of optical depth, LU decomposition for linear system solution.

- **mali.f90** : contains several subroutines used for MALI cycle : solution of preconditioned statistical equilibrium equations (Rybicki and Hummer, 1991), solution of radiative transfer equations using short characteristics method.

The output file is "fort.1" which contains results S/B (τ) for each transition.

2.3.5.1 Set of variables used in module "param_mod" (param.f90 file)

These are global variables.

- nlev=3 : number of atomic levels
- nfrq=16 : number of frequencies
- ndir=3 : number of directions
- nord=2 : quadrature order
- nd=53 : number of layers for optical depth grid
- pi : value of π
- h : Planck constant in $cm^2.g.s^{-1}$
- ryd : Rydberg constant in erg
- bolt : Boltzmann constant in $cm^2.g^{-1}.s^{-2}.K^{-1}$
- cl : velocity of light in $cm.s^{-1}$
- m_H : mass of hydrogen atom in g
- RS : Sun radius in cm
- ta=5000K : atmospheric temperature
- chil21=1 : absorption coefficient
- g (array of size nlev) : statistical weights of hydrogen levels

- nu (array of size (nlev,nlev)) : transition frequency
- A, B, C (arrays of size (nlev,nlev)) : Einstein A and B coefficients, collision rate coefficients C
- tau0, tau (arrays of size nd) : optical depth grids
- csz, wtdir (arrays of size ndir) : angular quadrature (cosine of the angle between radiation and z-axis) and angular quadrature weights
- wtnu (array of size nfrq) : integration weights with respect to frequency
- phi (array of size nfrq) : line profile
- phij, chi, jeff (arrays of size nd) : $\bar{J}$, line absorption coefficient, $\bar{J}^{eff}_{ll'}$
- sl (array of size (nfrq,nd)) : source function
- n, nprec (arrays of size (nlev,nd)) : population density, population density calculated at LTE
- ne : electron density
- bc0, bcd (arrays of size (nfrq,ndir)) : upper (top of layer, z(1)) and lower (z(nz)) boundary conditions
- lstar (array of size nd) : $\bar{\Lambda}^{*}_{ll'}$
- erls (array of size niter, niter is the number of iterations in MALI cycle. For more details, see subroutine "cycle_mali" in the file "mali.f90") : error
- jjeff, llstar (arrays of size (nlev,nlev,nd)) : intermediate arrays for $\bar{J}^{eff}_{ll'}$ and $\bar{\Lambda}^{*}_{ll'}$

2.3.5.2 Description of subroutines in module "general_mod" (general.f90 file)

Module **general_mod** contains subroutines used in MALI cycle.

Subroutine grilles : this subroutine sets up different grids for modeling a semi-infinite atmosphere. Specifically,
- we consider a grid of optical thickness "tau0" in cm, ordered by increasing value (see figure 1.1)
- we consider statistical weights of hydrogen levels, angular quadrature (cosine of the angle between radiation and z-axis), weights in angular quatdrature, frequencies of the 3 transitions, Einstein A coefficients and C_{ji} (see page 46 from Avrett (1968)),
- we calculate Einstein B coefficients
- we calculate Gaussian line profile. We renormalize profile and frequency quadrature.
- reduced frequency grid $x = \nu - \nu_0/\Delta\nu_D$, where ν is frequency, ν_0 is line center frequency, and $\Delta\nu_D$ is Doppler width

Subroutine initialisation_population : estimation of n_1 (population of level 1) from absorption coefficient $\chi_{21} = 1$ (opacity grid is fixed and becomes geometric grid "tau0"). The other population densities are calculated at LTE. We also compute total electron density assuming that each depth is initialized in the same way. We compute C_{ji} using populations at LTE

Subroutine eincoef : computation of induced emission coefficients B_{ji} and B_{ij}

Function dopwidth : Doppler width calculation

Subroutine boltzex : calculation of population density at LTE

Function planckf : calculation of Planck function

Subroutines ludcmp et lubksb : solution of the system of equations $matsys\ .\ N = SEC$ using LU decomposition (numerical recipes)

2.3.5.3 Description of subroutines in module "mali_mod" (mali.f90 file)

Module **mali_mod** contains subroutines solving in self-consistent way statistical equilibrium equations with non-LTE radiative transfer equations corresponding to each treated transition, using MALI and short characteristics methods.

Subroutine cycle_mali : niter is the number of iterations of MALI cycle, input by the user. Boundary conditions bc0 and bcd are then defined. Hence, MALI cycle starts :
- for each transition, we calculate renormalized absorption coefficient, we calculate $\bar{\Lambda}^*_{ll'}$ (we choose evaldiag=true in subroutine "rt1d"), we calculate $\bar{J}$ (we choose evaldiag=false in subroutine "rt1d"), we calculate J^{eff}. J^{eff} and $\bar{\Lambda}^*_{ll'}$ are stored in order to compute level populations
- we solve preconditioned statistical equilibrium equations (subroutine "malieqstat") and we calculate the error
- we write final populations and S/B (output) for each transition.

Subroutine malieqstat : Matrix $matsys$ of size (nlev,nlev) is built from preconditioned statistical equilibrium equations (2.17) of Rybicki and Hummer (1991). In H3CRD program, terms of $matsys$ matrix are :

- ⋆ $matsys(1,1) = A_{12}(1-\bar{\Lambda}^*_{12k}) + B_{12}\bar{J}^{eff}_{12k} + C_{12} + A_{13}(1-\bar{\Lambda}^*_{13k}) + B_{13}\bar{J}^{eff}_{13k} + C_{13}$ with $A_{12} = 0$ and $A_{13} = 0$

- ⋆ $matsys(1,2) = -A_{21}(1-\bar{\Lambda}^*_{12k}) - B_{21}\bar{J}^{eff}_{12k} - C_{21}$

- ⋆ $matsys(1,3) = -A_{31}(1-\bar{\Lambda}^*_{13k}) - B_{31}\bar{J}^{eff}_{13k} - C_{31}$

- ⋆ $matsys(2,1) = -A_{12}(1-\bar{\Lambda}^*_{12k}) - B_{12}\bar{J}^{eff}_{21k} - C_{12}$ with $A_{12} = 0$

- ⋆ $matsys(2,2) = A_{21}(1-\bar{\Lambda}^*_{21k}) + B_{21}\bar{J}^{eff}_{21k} + C_{21} + A_{23}(1-\bar{\Lambda}^*_{23k}) + B_{23}\bar{J}^{eff}_{23k} + C_{23}$ with $A_{23} = 0$

- ⋆ $matsys(2,3) = -A_{32}(1-\bar{\Lambda}^*_{23k}) - B_{32}\bar{J}^{eff}_{23k} - C_{32}$

- ⋆ $matsys(3,1) = matsys(3,2) = matsys(3,3) = 1$: closure condition. We replace the last equation of (2.3.8) by $n_e = n_1 + n_2 + n_3$ (particular case of hydrogen : $n_p = n_1 + n_2 + n_3 = n_H$ and $n_p = n_e$).

The following linear system is obtained :

$$matsys \; . \; N = SEC, \tag{2.3.10}$$

$$\text{with } N = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} \text{ and } SEC = \begin{pmatrix} 0 \\ 0 \\ n_e \end{pmatrix}$$

We solve system (2.3.10) using subroutines **ludcmp** and **lubksb** (LU decomposition). $matsys$ matrix can be written as :

$$\begin{pmatrix} A_{12}(1-\bar{\Lambda}^*_{12k})+B_{12}\bar{J}^{eff}_{12k}+C_{12}+A_{13}(1-\bar{\Lambda}^*_{13k})+B_{13}\bar{J}^{eff}_{13k}+C_{13} & -A_{21}(1-\bar{\Lambda}^*_{12k})-B_{21}\bar{J}^{eff}_{12k}-C_{21} & -A_{31}(1-\bar{\Lambda}^*_{13k})-B_{31}\bar{J}^{eff}_{13k}-C_{31} \\ -A_{12}(1-\bar{\Lambda}^*_{12k})-B_{12}\bar{J}^{eff}_{21k}-C_{12} & A_{21}(1-\bar{\Lambda}^*_{21k})+B_{21}\bar{J}^{eff}_{21k}+C_{21}+A_{23}(1-\bar{\Lambda}^*_{23k})+B_{23}\bar{J}^{eff}_{23k}+C_{23} & -A_{32}(1-\bar{\Lambda}^*_{23k})-B_{32}\bar{J}^{eff}_{23k}-C_{32} \\ 1 & 1 & 1 \end{pmatrix}$$

Subroutine rt1d : Figure 2.5 summarizes calculation of $\bar{J}$. Subroutine **rt1d** computes either the diagonal of Λ operator (when evaldiag=true) or $\bar{J}$ (when evaldiag=false), using short characteristics method (Leger, 2008; Lambert et al., 2016).

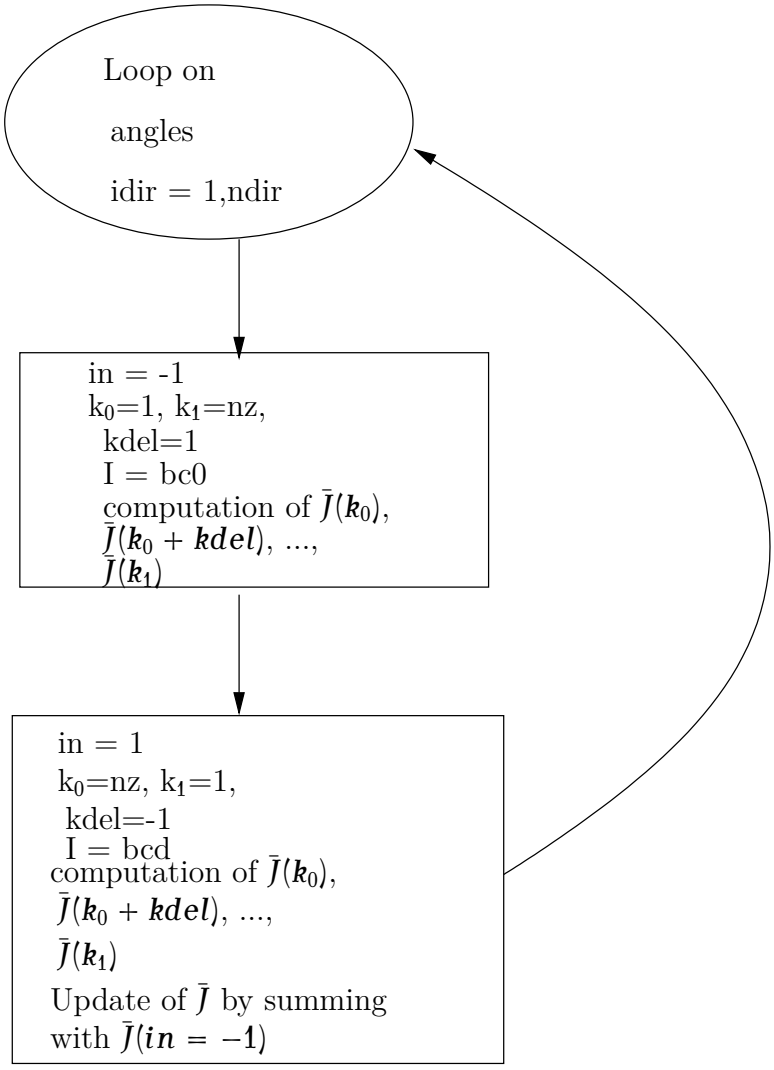

Figure 2.5: Algorithm for computation of $\bar{J}$.

2.3.6 Running H3CRD program

- make clean
- make
- ./h3crd

Result file is fort.1 which contains results of S/B (τ) for each transition.

2.4 Local operator with background continuum (with continuum absorption). Application to a semi-infinite atmosphere) : planned

Part II

Non-LTE radiative transfer in 2D : planned

Acknowledgements

I thank MEDOC director Eric Buchlin. I also thank Pierre Gouttebroze for his help on the case of a two-level atom in 1D. I thank Jacques Dubau for atomic physics part.

14th April 2020
M. C-Y

Bibliography

L. Auer. Improved Boundary Conditions for the Feautrier Method. *Astrophys. J. Lett.*, 150: L53, October 1967. doi: 10.1086/180091.

E. H. Avrett. Resonance lines in Astrophysics. *NCAR*, 1968.

A. Hui. Rapid computation of the Voigt and complex error functions. *Journal of Quantitiative Spectroscopy and Radiative Transfer*, 19:509–516, May 1978. doi: 10.1016/0022-4073(78)90019-5.

J. Humlicek. Optimized computation of the Voigt and complex probability functions. *Journal of Quantitiative Spectroscopy and Radiative Transfer*, 27:437–444, 1982.

D. G. Hummer. Non-coherent scattering-VI. Solutions of the transfer problem with a frequency-dependent source function. *Monthly Notices of the RAS*, 145:95, 1969. doi: 10.1093/mnras/145.1.95.

D. G. Hummer and G. Rybicki. Computational Methods for Non-LTE Line_transfer Problems. *Methods in Computational Physics*, 7:53–126, 1967.

J. T. Jefferies. *Spectral line formation*. Blaisdell, 1968.

L. C. Johnson. Approximations for Collisional and Radiative Transition Rates in Atomic Hydrogen. *Astrophys. J.*, 174:227, May 1972. doi: 10.1086/151486.

J. Lambert, F. Paletou, E. Josselin, and J-M. Glorian. Numerical radiative transfer with state-of-the-art iterative methods made easy. *European Journal of Physics*, 37(1):015603, January 2016. doi: 10.1088/0143-0807/37/1/015603.

L. Leger. *Transfert de rayonnement hors-ETL multidimensionnel. Application au spectre de l'hélium dans les protubérances solaires*. PhD thesis, Université de Toulouse III - Paul Sabatier, 2008.

K.-C. Ng. Hypernetted chain solutions for the classical one-component plasma up to Gamma equals 7000. *Journal of Chemical Physics*, 61:2680–2689, October 1974. doi: 10.1063/1.1682399.

G. L. Olson and P. B. Kunasz. Short characteristic solution of the non-LTE transfer problem by operator perturbation. I. The one-dimensional planar slab. *Journal of Quantitiative Spectroscopy and Radiative Transfer*, 38:325–336, 1987. doi: 10.1016/0022-4073(87)90027-6.

G. L. Olson, L. H. Auer, and J. R. Buchler. A rapidly convergent iterative solution of the non-LTE line radiation transfer problem. *Journal of Quantitiative Spectroscopy and Radiative Transfer*, 35:431–442, June 1986. doi: 10.1016/0022-4073(86)90030-0.

F. Paletou. A note on improved computations of solar prominences: 2D radiative models. *Astron. Astrophys.*, 311:708–709, July 1996.

F. Paletou. Transfert de rayonnement : méthodes itératives. *C. R. Acad. Sci. Paris*, 2:885–898, 2001.

G. B. Rybicki. Recent Advances in Computational Methods. In L. Crivellari, I. Hubeny, and D. G. Hummer, editors, *NATO Advanced Science Institutes (ASI) Series C*, volume 341 of *NATO Advanced Science Institutes (ASI) Series C*, page 1, 1991.

G. B. Rybicki and D. G. Hummer. An accelerated lambda iteration method for multilevel radiative transfer. I - Non-overlapping lines with background continuum. *Astron. Astrophys.*, 245:171–181, May 1991.