NLTE implementation in Turbospectrum - Bertrand Plez - 06/01-2020
Revised 07/04-2020

1 Equations

From Rutten 2003 (Eq. 2.105) the NLTE line source function is:

$$S_\lambda = \frac{j_\lambda}{\alpha_\lambda} = \frac{2hc^2}{\lambda^5} \frac{\psi/\varphi}{\frac{b_l}{b_u} e^{h\nu/kT} - \chi/\varphi} \tag{1}$$

and with complete redistribution ($\chi = \varphi = \psi$),

$$S_\lambda = \frac{2hc^2}{\lambda^5} \frac{1}{\frac{b_l}{b_u} e^{h\nu/kT} - 1} = B_\lambda \frac{e^{h\nu/kT} - 1}{\frac{b_l}{b_u} e^{h\nu/kT} - 1} \tag{2}$$

The bound-bound extinction is then (complete redistribution):

$$\boxed{\alpha_\lambda = \alpha^*_\lambda b_l \frac{1 - \frac{b_u}{b_l} e^{-h\nu/kT}}{1 - e^{-h\nu/kT}}} \tag{3}$$

From which the emissivity:

$$j_\lambda = \alpha_\lambda S_\lambda = \alpha^*_\lambda b_l \frac{1 - \frac{b_u}{b_l} e^{-h\nu/kT}}{1 - e^{-h\nu/kT}} B_\lambda \frac{e^{h\nu/kT} - 1}{\frac{b_l}{b_u} e^{h\nu/kT} - 1} \tag{4}$$

$$j_\lambda = \alpha^*_\lambda b_l B_\lambda \frac{1 - \frac{b_u}{b_l} e^{-h\nu/kT}}{\frac{b_l}{b_u} - e^{-h\nu/kT}} \tag{5}$$

and finally:

$$\boxed{j_\lambda = \alpha^*_\lambda b_u B_\lambda} \tag{6}$$

The total source function at a given wavelength, if more than one line contributes is:

$$\boxed{S_\lambda = \frac{\sum j_\lambda}{\sum \alpha_\lambda}} \tag{7}$$

Similarly for the continuum:

$$\boxed{\alpha^{bf}_\lambda = \alpha^{bf*}_\lambda b_i \frac{1 - \frac{b_c}{b_i} e^{-h\nu/kT}}{1 - e^{-h\nu/kT}}} \tag{8}$$

$$\boxed{j^{bf}_\lambda = \alpha^{bf*}_\lambda b_c B_\lambda} \tag{9}$$

and

$$\boxed{\alpha^{ff}_\lambda = \alpha^{ff*}_\lambda b_c} \tag{10}$$

$$\boxed{j^{ff}_\lambda = \alpha^{ff*}_\lambda b_c B_\lambda} \tag{11}$$

$$S^{ff}_\lambda = B_\lambda \tag{12}$$

2 Philosophy of the implementation in Turbospectrum

In TS the implementation can be done as follows:

- The identification of the upper and lower level is read from the line list, using two more variables at the end of the statement (eqwidt.f version, as bsyn.f does not have the obseqw and eqwerror):

  » read(lunit,*) xlb,chie,gfelog,fdamp,gu,raddmp,levlo,levup,obseqw,eqwerror

  These identifications are those of the model atom.

- The departure coefficients are read from the MULTI output file with identical level id:s.

- The absorption coefficient is corrected in bsyn.f (and eqwidt.f, if we plan to use it). Variable is called plez(). See loop: do 111 ...continue.

- The extinction coefficient is incremented: abso() only. The scattering part absos() is kept to zero. Warning: abso() is normalized by the standard opacity, and is in cm^2/gram of stellar matter. It should either be transformed to α (cm^{-1}), or the emissivity should be given in cm^2/gram as well.

- A new array is created and incremented: emissivity().

- After all lines have been scanned, and their contribution added in abso() and emissivity(), bsynb.f is called.

- In bsynb.f the continuum absocont() and absoscont() are computed, as well as the Planck function, then traneq.f is called to solve the transport equation for the continuum. SOME THINKING NEEDED HERE. Do we want to do it this way? How do we add the continuum contribution for the NLTE species?

- Then same thing is done including lines. In MULTI, the continuum scattering part of the source function is treated properly. We should keep it this way in TS, i.e. keep σ_{cont} and κ_{cont} separate. The source function can then be iterated as usual in TS. CHECK that this gives the same result as MULTI. The source function for the lines needs to be changed from bplan() to emissivity/alpha, i.e. emissivity/abso()*ross().

The implementation must keep compatibility with the older line list format, and with LTE calculations. For this to work a flag NLTE = .true./.false. is implemented (all departure coefficients set to one).

One uncertain issue is the treatment of bf and ff transitions for NLTE species. We do not implement it for the moment.

3 Actual implementation

3.1 babsma

Nothing was changed in the babsma.f part (computation of the continuum opacities). This can be changed later if we decide to implement NLTE corrections for some of the continuous opacities (photoionisation, or hydrogen opacities).

3.2 bsyn

In bsyn.f, a new switch is implemented (NLTE = .false. or .true.), implemented in the run-script by, e.g.[1] :

```
'NLTE :'          '.false.'
```

If set to .true. this switch results in a special treatment of the line list, the calculation of the line opacity, and the source function. By default it is set to .false., ensuring back compatibility with previous LTE versions.

3.2.1 line list

In order for the NLTE information to be read from the line list, a keyword must be included in the free comment line that is included after the line providing the element, ionisation stage and number of lines in the line list. This comment just needs to include somewhere the word NLTE. Then each line is read according to:

```
          read(lunit,*) xlb,chie,gfelog,fdamp,gu,raddmp,levlo,levup,
     &                  comment_line,
     &                  ilevlo,ilevup,idlevlo,idlevup
```

With the last 4 items (2 integers and 2 character*20) being the lower and upper level number, and identifications. Only the number is used in the following.

3.2.2 departure coefficients

If the species is treated in NLTE, the model atom is read, and departure coefficients are set for all levels (new routines read_modelatom.f and read_departure.f). if a species is to be treated in LTE, the departure coefficients are all set to 1.0.

[1] This is different from the PURE-LTE logical switch which, if set in both babsma and bsyn, includes all scattering into the absorption coefficient, resulting in a pure planckian source function.

3.2.3 line absorption and source function

For all lines of LTE and NLTE species the absorption coefficient is calculated as in Eq. 3, and the emissivity as in Eq. 6. This is done at each wavelength, where the absorption for all contributing lines is added to the continuum absorption. The same is done for the emissivity. For the continuum contributions, Eq. 8, 9, 10, and 11 are used, with all departure coefficients set to 1. When all contributions have been added, the source function is calculated using Eq. 7.

3.3 Solving the radiative transfer equation

In the subroutine bsynb.f, the continuum radiative transfer problem is solved as usual using the Feautrier method, and iterating on the starting Planck source function to converge to the final $S = \frac{\kappa B + \sigma J}{\kappa + \sigma}$ at all wavelengths. Then the line + continuum radiative transfer is solved using the NLTE source function.

In case there is a velocity field, the outgoing intensities and flux are then calculated using the same source function, by a simple integration of S.exp(-tau). This is not strictly correct, as the source function and the opacities should be Doppler-shifted. **There is no iteration on the source function in that case**.

For the total (line + continuum) opacity, the scheme is similar, except that the absorption coefficient is Doppler-shifted. **There is no Doppler-shift implemented so far for the scattering continuum opacities, and the source function is not iterated**.

4 Tests

4.1 Radiative transfer solver

4.1.1 Direct quadrature of source function

This solver (routine Iplus_calc.f) computes the integral $\int_0^{\tau_{\max}} s(t) exp(-t) dt$ on the tau scale for each ray. For optically thick rays it uses the diffusion approximation for the deepest point: $i^+(\tau_{\max}) = S(\tau_{\max}) + \frac{dS}{d\tau}(\tau_{\max})$. For optically thin rays it uses an approximation of the intensity generated in the layer above $\tau_{\min}$: $i^+ = S(\tau_{\min}) * (1 - exp(-\tau_{\min}))$.

This routine gives slightly different results from the ordinary Feautrier routine (traneq.f). It was therefore tested against an analytic solution, for a source function of the form $S(tau) = \sum_0^{n_{max}} a_i \tau^i$, with $a_i = 1$. This was tested for different quadratures (n=1 is the shallowest depth point):

1. propagation of the intensity from the deepest point out to the surface, using the analytic expression for the integral, and an average of the source function in the interval:

$$i_n^+ = I_{n+1}^+ e^{-(\tau_{n+1} - \tau_n)} + 0.5(S_n + S_{n+1}) e^{-(\tau_{n+1} - \tau_n)}$$

2. propagation of the intensity, using the trapezoidal rule for the integration within each interval, i.e. approximating $Se^{-\tau}$ by an affine function within each τ interval:

$$i_n^+ = i_{n+1}^+ e^{-(\tau_{n+1}-\tau_n)} + (S_{n+1}e^{-(\tau_{n+1}-\tau_n)} + S_n)(\tau_{n+1}+\tau_n)/2.$$

3. trapezoidal rule for the whole integral:

$$i^+(1) = \sum_1^{n_{max}} (S_n e^{-\tau_n} + S_{n+1}e^{-\tau_{n+1}})(\tau_{n+1}-\tau_n)/2$$

Rectangle rule for the whole integral gave larger errors. Various degree 4 polynomials for the source function were tested, including non-monotonous. With a sampling of 0.1 in τ, and a range between 10^{-6} and 100, method 2 gave better than 1% results, with method 3 always close. Increasing the sampling to 0.2 and 0.3 increases the error to 3.5 and 8% respectively. Method 1 shows errors in excess of 1% in all cases, increasing above 10 or even 100% with the lowest sampling. We adopt method 2. Note that the Feautrier plane-parallel and spherical formal solvers do not always agree. Their performances degrade when the optical depth sampling is less tight. The PP version does not work in the optically thin case. The spherical version and our adopted method agree within less than a percent most of the time, and occasionally a little more.

4.1.2 Static case

Using departure coefficients of unity recovers the Planck function for the line source function, and a spectrum identical to the LTE case within rounding errors.

The use of a MULTI output for a Ca model atom (106 levels) in a MARCS solar model atmosphere, leads to LTE and NLTE line profile that are identical to the MULTI calculation.

Need to check for a case with strong scattering in the continuum, e.g. Ca II H and K lines in a metal-poor star.

4.1.3 Non-zero velocities case

4.2 NLTE case with departure coefficients set to 1

4.3 Static NLTE case

TEST if iteration of source function must be done or not.

4.4 non-zero velocity NLTE case