

NLTE2D: Toulouse 2D non-LTE radiative transfer code

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

Ludovick Léger

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

April 20, 2022

Web access:

[https://idoc.ias.universite-paris-saclay.fr/MEDOC/Radiative transfer codes](https://idoc.ias.universite-paris-saclay.fr/MEDOC/Radiative%20transfer%20codes)

Contents

1	Introduction	2
2	Description of NLTE2D code	2
3	Algorithms	3
4	Description of subroutines and input/output files	3
4.1	Input files	3
4.2	Output files	4
4.3	Visualization files	5
4.4	Description of major variables used in the main program nlte2D.f90	5
4.5	Description of subroutines	6
5	Running NLTE2D	8
6	Results / Test case / CPU time	9
7	Important note about the usage of codes	11
8	Acknowledgements	11
	References	13

1 Introduction

The NLTE2D code was written in Fortran 90 by L. Léger and F. Paletou (Léger et al., 2007) and describes non-LTE radiative transfer (2D) in solar prominences, using complete frequency redistribution. The problem consists in solving a set of equations **including the ionization equilibrium for hydrogen**, the statistical equilibrium of level populations, non-LTE radiative transfer in radiatively allowed bound-bound transitions, and **the Lyman continuum** using several iterative methods (MALI, GS/SOR and multi-grid).

The implementation of MALI (Multilevel Accelerated Lambda Iteration), GS (Gauss-Seidel) and successive over-relaxation (SOR) iterative schemes in 2D, together with a multi-grid algorithm, is thoroughly described in the frame of the short characteristics method for the computation of the formal solution of the radiative transfer equation in cartesian geometry (Léger et al., 2007).

The model of atmosphere considered here is: 8000 K (temperature of the isothermal slab), 0.5 dyn cm^{-2} (gas pressure). See further the test case at section 6.

For this model, the formation of hydrogen lines is first considered. Thus, we obtain the electron density and the emergent intensities for hydrogen lines and the Lyman continuum.

For the present release, the main additions are: the self-consistent calculation of the electron density 2D distribution, and the explicit transfer of the Lyman continuum (related details will be explicitated in the next version of this document; see also Heinzel 1995).

NLTE2D code is available from MEDOC website :

[https://idoc.ias.universite-paris-saclay.fr/MEDOC/Radiative transfer codes/NLTE2D](https://idoc.ias.universite-paris-saclay.fr/MEDOC/Radiative%20transfer%20codes/NLTE2D)

This code is the continuation of [Toulouse 2D numerical radiative codes](#), available from [MEDOC](#) website.

Important: this documentation will be detailed further in another release. In the meantime, please feel free to contact F. Paletou, for any assistance.

2 Description of NLTE2D code

Nature of the physical problem: Resolution of 2D non-LTE radiative transfer problem considering freestanding and illuminated slabs (transfer of hydrogen lines and the Lyman continuum)

Method of solution: MALI (Multilevel Accelerated Lambda Iteration), Gauss-Seidel and successive over-relaxation iterative schemes in 2D, together with a multi-grid algorithm (and short characteristics method for the formal solution in 2D cartesian geometry)

Other relevant information: Self-consistent computation of the electron density, and transfer of the Lyman continuum

Authors: L. Léger & F. Paletou

Program available from:

[https://idoc.ias.universite-paris-saclay.fr/MEDOC/Radiative transfer codes/NLTE2D](https://idoc.ias.universite-paris-saclay.fr/MEDOC/Radiative_transfer_codes/NLTE2D)

Computer(s) on which program has been tested: IAS server

Operating System(s) for which version of program has been tested: Linux (Debian 10)

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

Status: Stable

Accessibility: open (**MEDOC**)

Nb. of code lines in combined program and test deck: 7211

Typical running time: ~ 14 minutes for the test case (see input file) at section 6 and for a PC with 12 processors at 3.10 GHz each

References:

- Leger (2008)
- Heinzel (1995)
- Léger et al. (2007)
- Paletou and Léger (2007)
- Paletou (1995)
- Bjørngen et al. (2019) : see in particular Appendix A.

3 Algorithms

All iterative schemes and numerical methods implemented in NLTE2D code are explained in detail in the above mentioned references.

4 Description of subroutines and input/output files

4.1 Input files

The file to modify is “input” which contains input parameters. So, it is not necessary to enter them by hand when executing the NLTE2D code. The definition of each parameter is given below.

- n : quadrature order
- nbgril : number of 2D geometric grids

- `ymax`, `dely`, `ny(nbgril)` : definition of the finest 2D y-grid. The number of points corresponds to the number of points on the finest grid (desired grid)
- `zmax`, `delz`, `nz(nbgril)` : definition of the finest 2D z-grid.
- `ta` : slab temperature
- `pg` : gas pressure
- `nlev` : number of energy levels
- `lec_grille` : possibility to read the input grid, only if you work with a single grid (y/n)
- `eps_lu` : relative error (criterion for stopping iterative process when running the code for one geometric grid)
- `sor_lu` : parameter for using SOR on the loosest grid (y/n)
- `ali` : parameter for using MALI only in the case where there is only one geometric grid (y/n)
- `itr_finale` : number of MALI iterations before Gauss-Seidel
- `presmooth` : number of presmooth iterations
- `postsmooth` : number of postsmooth iterations.

If `lec_grille='y'`, input 2D grid “grille2d.input” (y-grid and z-grid) is read. Otherwise, we create a grid (grille2d.res).

`probaa.dat`, `probab.dat`, `probag.dat`, `probra.dat`, `prolya.dat`, `prolyb.dat` are the input files (intensity for $Ly\alpha$, $Ly\beta$, $Ly\gamma$, $Ly\delta$, ...) for `xgrids.f90`.

4.2 Output files

Output files for `nlte2d.f90` are :

- `grille2d.res` : geometric grids (if `lec_grille='n'`)
- `disque.res` : disc profiles along y (all lines)
- `opacites.res` : line opacities (`tauy`, `tauz`)
- `halpha.res` : opacity for $H\alpha$ line
- `intensite.res` : emerging intensities (limb then disk) for Lyman continuum
- `populations.res` : population of level 1, 2, 3 of hydrogen, electron and hydrogen densities
- `toto.res` : `Rik` (radiative rate) and `Cik` (collision rate) per grid
- `profil.res` : upper and lower level, absorption profile, frequency weight, incident radiation for each transition

- limbe.res : limb profiles along z (all lines)

Output file for seta.90 is quadrature.res (angular quadrature for the formal solver).

4.3 Visualization files

Two python files, graphe.py and LyCont2D.py, are used to visualize :

- electron density (populations.res)
- emergent Lyman continuum intensity vs. frequency (from head) at respective ny and nz grid points (intensity.res).

4.4 Description of major variables used in the main program nlte2D.f90

- nlev : total number of (bb) energy levels including continuum
- ntran : number of transitions (Lyman continuum is: ntran+1)
- nbb : number of bound-bound transitions (nlev-1)
- nbgril : number of 2D geometric grids (for the multi-grid process, otherwise: 1)
- nfrq : number of frequencies in a line
- tab2d(nbgril) : grid array of size nbgril
- ny(nbgril),nz(nbgril) : number of points in y and z per grid
- jeff(ny(1),nz(1),nbb,nbb) : J^{eff} array (see MALI formalism)
- lastar(ny(1),nz(1),nbb,nbb) : Λ^* matrix
- tab2d(nbgril)%y(ny(nbgril)),tab2d(nbgril)%z(nz(nbgril)) : spatial coordinates in y and z (output : grilles2d.res)
- tab_profil(ntran+1) : line profile array
- grille2d(nbgril)%stran(ntran+1) : transition array per grid
- tab2d(nbgril)%wz(nz(nbgril)) : dilution factor for each grid; see [Paletou \(1996\)](#)
- grille2d(nbgril)%pop(ny(nbgril),nz(nbgril),nlev) : level populations for each grid
- grille2d(nbgril)%stran(ntran)%source(ny(nbgril),nz(nbgril),nfrq) : total source function in frequency
- grille2d(nbgril)%stran(ntran)%sl(ny(nbgril),nz(nbgril)) : line source function
- nelec : electron density

4.5 Description of subroutines

Fortran files of NLTE2D code are shortly described below. The link between each subroutine is mentioned at the beginning of each file by the keyword "use". Subroutines written in bold are additions to the previous [Toulouse 2D numerical radiative codes](#).

- nlte2d.f90 : main program
- Opacities :
 - abshyd.f90, abssil.f90, abscar.f90, absmag.f90, absalu.f90, absfer.f90 : continuous absorption at LTE by hydrogen, silicon carbon, magnesium, aluminum, iron
 - avray.f90 : rayleigh scattering per neutral hydrogen atom
 - avhm.f90 : H^- opacity per neutral hydrogen atom and unit electron pressure
 - avh2p.f90 : H^{2+} opacity per neutral hydrogen atom and per H^+
 - calabs.f90 : computes the continuum absorption coefficient κ
- Atomic data used in transfer :
 - boltzex.f90 and saha.f90 : compute density of populations according to the Boltzmann-Saha distribution
 - aems.f90 : Einstein's coefficient for the spontaneous emission of hydrogen atom
 - planckf.f90 : computes Planck functions
 - eincoef.f90 : computes all Einstein coefficients, per transition
 - **sigmapi.f90** : computes photoionization cross section for H atom (pay attention to Gaunt factor)
 - gaunt.f90 : computes Gaunt factors
 - colex.f90 : computes collision rate (bound-bound transitions)
 - h1bb.f90 : computes collisional rate coefficients
 - **colion.f90** : computes collision rate (bound-free transitions)
 - **h1bf.f90** : computes collisional ionization coefficients
 - **lycont.f90** : computes source function, absorption coefficient and frequencies for the transfer of Lyman continuum
 - hdata.f90 : computes Einstein's coefficients A_{ij} and transition frequency for a given number of allowed transitions
- Transfert / Statistical equilibrium equations :
 - dopwidth.f90 : computes a Doppler width (microturbulence may be included)
 - lubksb.f90, ludcmp.f90 : standard procedures for the so-called LU-decomposition
 - seta.f90 : defines 2D angular quadrature for the formal solver ([Auer and Paletou \(1994\)](#)), see also Mihalas et al. 1978 for a more precise reference to 'sets' A and B)

- setgeo : computes (once) all sets of points and interpolation weights required by the formal solver rt2d; it is important that the 2D grid is further swept according to the order defined during this first pass (ibid. and details in section 3 of [Léger et al. \(2007\)](#) - critical for the Gauss-Seidel implementation, hereafter)
 - rt2d.f90 : 2D formal solver. Main input : evaldiag (TRUE : computation of the diagonal of lambda operator. FALSE : computation of Jbar). Output : phij array (diagonal of the lambda operator or Jbar), jayx array (diagonal of psistar or jayx for Lyman continuum)
 - rt2dgsml.simple.f90 : 2D Gauss-Seidel formal solver
 - **gsmeqstat.f90** : solves preconditioned statistical equilibrium equations ([Rybicki and Hummer \(1992\)](#)) with local operator with background continuum ([Rybicki and Hummer \(1991\)](#)) and computation of electron density (nelec appears explicitly). Inputs : matrix of Jeff coefficients for each transition at a given position, Λ^* matrix (lastar array) for each transition at a given position, second member (second array) for the resolution of statistical equilibrium equations at a given position. Output : population matrix (popu array) at a given position
 - ratio.f90 : computes r for a grid going from 0 to xmax, in nx points, with a first step dx. The grid in x is such that $x_i = x_{i-1} + r \cdot (x_{i-1} - x_{i-2})$. It is used for logarithmic grids, not only spatial, but also in frequency for the wings of large extension lines (a first part in the 'doppler core' at ~ 0.4 dopplewidth and not constant, then logarithmic beyond, for sweep the wings extended due to the profile of Voigt)
 - nmgm2d.f90, prolong2d.f90, restrict2d.f90 : multi-grid method (will be documented later, see [Leger \(2008\)](#))
 - **rf2d.f90** : specific radiative transfer of Lyman continuum
 - xgrids.f90 : computes absorption profile, incident radiations, reduced frequency grid and weight frequency for each line. Input files : probaa.dat, probab.dat, probag.dat, probra.dat, prolya.dat, prolyb.dat : intensity for $Ly\alpha$, $Ly\beta$, $Ly\gamma$, $Ly\delta$, ...
 - cubquad.f90 : computes weights for a cubic Hermite quadrature
 - init_voigt.f90 and voigt.f90 : compute Voigt profile (pay attention to normalization)
 - radbf.f90 : computes incident radiation for each continuum of hydrogen
 - matray.f90 : solves linear equations for populations (electron density known a priori)
 - formal.f90 : computes the emerging (observable) radiations once the scheme has converged
 - testmonotonie.f90 : checks the monotony of a function
- Modules (common variables used in subroutines and main program) :
 - atomicmodel.f90
 - constantes.f90

-
- transitions.f90
 - quadrature.f90
 - common.f90

5 Running NLTE2D

- Download the package source file NLTE2D.tgz from MEDOC website:
https://idoc.ias.universite-paris-saclay.fr/MEDOC/Radiative_transfer_codes/NLTE2D
- gfortran compiler is required. Type the following linux commands:
- **tar -xvzf NLTE2D.tgz**
- **cd NLTE2D**
- Make the following perl scripts compil.pl and launch.pl executable :
chmod +x compil.pl launch.pl
- The folder contains the following files: compil.pl, launch.pl, nlte2d.f90 (main program), rt2dgsm.simple.f90, nmgm2d.f90, rf2d.f90, rt2d.f90, hdata.f90, aems.f90, ratio.f90, seta.f90, dopwidth.f90, eincoef.f90, boltzex.f90, saha.f90, xgrids.f90, cubquad.f90, init_voigt.f90, voigt.f90, radbf.f90, planckf.f90, sigmapl.f90, gaunt.f90, colex.f90, h1bb.f90, colion.f90, h1bf.f90, setgeo.f90, lycont.f90, abshyd.f90, absfil.f90, abscar.f90, absmag.f90, absalu.f90, absfer.f90, avray.f90, avhm.f90, avh2p.f90, calabs.f90, gsmeqstat.f90, matray.f90, testmonotonie.f90, ludcmp.f90, lubksb.f90, formal.f90, prolong2d.f90, restrict2d.f90, atomicmodel.f90, transitions.f90, quadrature.f90, constantes.f90, common.f90, grille2d.input, probaa.dat, probab.dat, probag.dat, proba.dat, prolya.dat, prolyb.dat, input (input parameter file), graphe.py (python file to visualize spatial distribution of level 3 of H), LyCont2D.py (python file to visualize spatial distribution of Lyman continuum)
- The file to modify is **input** : it contains input parameters. It is not necessary to enter them by hand when executing the NLTE2D code
- Run the code by typing :
./compil.pl (2 times if the first time the following message is displayed "Fatal error : Cannot open module file xxx.mod")
./launch.pl
- The executable of the code is nlte2d
- The output files of the main program nlte2d.f90 are : grille2d.res, quadrature.res, profils.res, disque.res, opacites.res, halpha.res, intensite.res, populations.res, toto.res, profils.res, limbe.res
- The output file of seta.90 is : quadrature.res

-
- The folder **results** contains the output files corresponding to a test case (corresponding to parameters from « input » file) to be able to check if your results are good
 - To visualize electron density (populations.res file), we use graphe.py file by typing the following command:
python graphe.py
 - To visualize emergent Lyman continuum intensity (intensite.res file), we use LyCont2D.py file by typing the following command:
python LyCont2D.py
 - Before running NLTE2D again, type: **rm nlte2d *.o *.mod**

6 Results / Test case / CPU time

Results obtained by NLTE2D code can be found in [Léger et al. \(2007\)](#) for different cases.

Consider the test case with the following parameters (see input file of the code package NLTE2D.tgz) :

$n = 8$, $nbgril = 2$, $y_{max} = 5.10^3$, $dely = 10^{-4}$, $ny(nbgril) = 63$, $z_{max} = 3.10^4$, $delz = 10^{-4}$,
 $nz(nbgril) = 63$, $ta = 8000$, $pg = 0.5$, $nlev = 6$, $lec_grille = y$, $eps_lu = 10^{-4}$, $sor_lu = n$,
 $ali = n$, $itr_finale = 0$, $presmooth = 3$, $postsmooth = 5$.

Figure 1 (resp. Figure 2) represents electron density (resp. emergent Lyman continuum radiation at limb), for illustrative purpose.

For the test case above, on a PC with 12 processors at 3.10 GHz each, the CPU time is around 14 minutes.

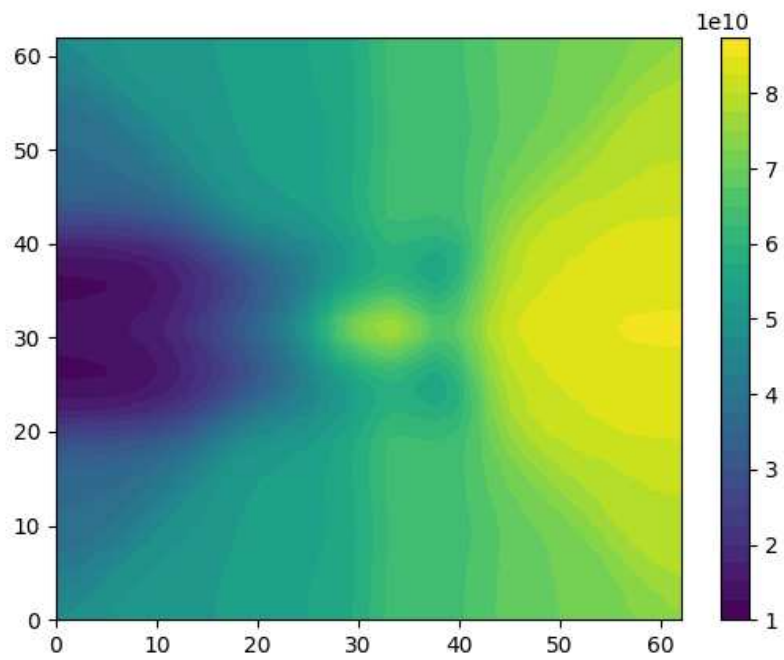


Figure 1: Electron density

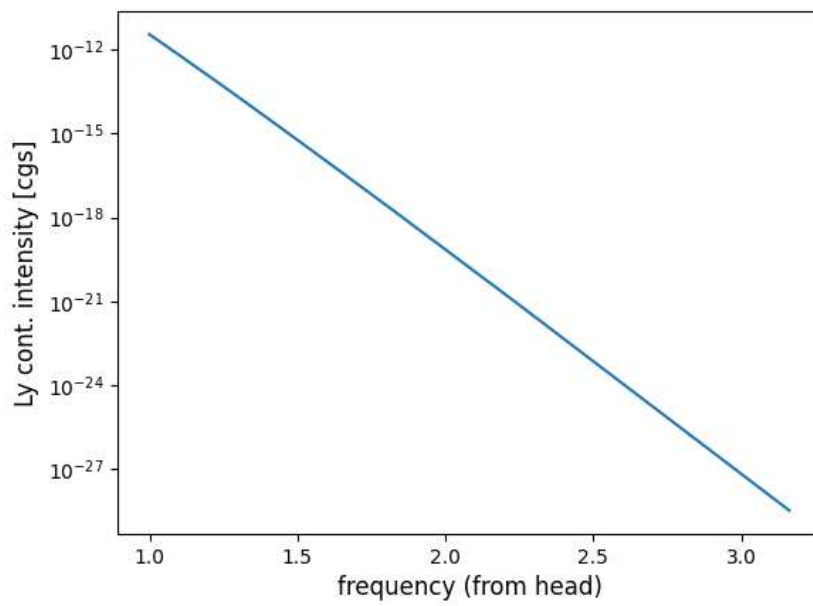


Figure 2: Example of emergent Lyman continuum radiation at limb

7 Important note about the usage of codes

We strongly encourage interested scientists to use, and develop further our codes. However, we shall ask them to comply with the following rules:

1. the first refereed-journal publication of any new user should include both the names of L. Léger and F. Paletou among the co-authors (with affiliation: Université Paul Sabatier, Observatoire Midi-Pyrénées, Cnrs, Cnes, IRAP, F-31400 Toulouse);
2. the MEDOC service at IAS (Orsay, France) which distributes these ressources should also be properly acknowledged;
3. further publications should be mentioned to frederic.paletou@univ-tlse3.fr

8 Acknowledgements

We thank MEDOC director Eric Buchlin.

References

- L. H. Auer and F. Paletou. Two-dimensional radiative transfer with partial frequency redistribution I. General method. *Astron. Astrophys.*, 285:675–686, May 1994. <https://ui.adsabs.harvard.edu/abs/1994A&A...285..675A>.
- Johan P. Bjørgen, Jorrit Leenaarts, Matthias Rempel, Mark C. M. Cheung, Sanja Danilovic, Jaime de la Cruz Rodríguez, and Andrii V. Sukhorukov. Three-dimensional modeling of chromospheric spectral lines in a simulated active region. *Astron. Astrophys.*, 631:A33, November 2019. doi: 10.1051/0004-6361/201834919. <https://ui.adsabs.harvard.edu/abs/2019A&A...631A..33B>.
- P. Heinzel. Multilevel NLTE radiative transfer in isolated atmospheric structures: implementation of the MALI-technique. *Astron. Astrophys.*, 299:563, July 1995. <https://ui.adsabs.harvard.edu/abs/1995A&A...299..563H>.
- Ivan Hubeny and Dimitri Mihalas. *Theory of Stellar Atmosphere*. NJ: Princeton University Press, 2014.
- Julien Lambert, Frédéric Paletou, Eric Josselin, and Jean-Michel 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. <https://ui.adsabs.harvard.edu/abs/2016EJPh...37a5603L>.
- 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. <https://tel.archives-ouvertes.fr/tel-00332781>.
- L. Léger, L. Chevallier, and F. Paletou. Fast 2D non-LTE radiative modelling of prominences. Numerical methods and benchmark results. *Astron. Astrophys.*, 470(1):1–9, July 2007. doi: 10.1051/0004-6361:20066730. <https://ui.adsabs.harvard.edu/abs/2007A&A...470....1L>.
- F. Paletou. Two-dimensional multilevel radiative transfer with standard partial frequency redistribution in isolated solar atmospheric structures. *Astron. Astrophys.*, 302:587, October 1995. <https://ui.adsabs.harvard.edu/abs/1995A&A...302..587P>.
- F. Paletou. A note on improved computations of solar prominences: 2D radiative models. *Astron. Astrophys.*, 311:708–709, July 1996. <https://ui.adsabs.harvard.edu/abs/1996A&A...311..708P>.
- Frédéric Paletou and Ludovick Léger. Fast multilevel radiative transfer. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 103:57–66, January 2007. doi: 10.1016/j.jqsrt.2006.07.003. <https://ui.adsabs.harvard.edu/abs/2007JQSRT.103...57P>.
- 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. <https://ui.adsabs.harvard.edu/abs/1991A&A...245..171R>.

G. B. Rybicki and D. G. Hummer. An accelerated lambda iteration method for multilevel radiative transfer. II. Overlapping transitions with full continuum. *Astron. Astrophys.*, 262:209–215, August 1992. <https://ui.adsabs.harvard.edu/abs/1992A&A...262..209R>.