Turbospectrum for NLTE; v20.0

Bertrand Plez, Montpellier University, France,
Jeffrey Gerber, Purdue University, Indiana, USA,
Ekaterina Magg, MPIA Heidelberg, Germany,
and Maria Bergemann, MPIA Heidelberg, Germany

May 30, 2022

1 Credits

If you use this code and the accompanying data (linelist, NLTE data, 1D, average 3D atmospheres), you should cite the papers described in the file `public-ref-NLTE-TS.txt` that is provided on Github. This file contains all references described below and can be directly included in a LATEX document.

- We provide two Python wrappers that can be used to generate NLTE synthetic spectrum in any point in the parameter space of FGKM stars. These wrappers should be credited to Gerber (2022) and Magg et al. (2022).
- NLTE model atoms should be credited to the following studies: H (Mashonkina et al. 2008), O (Bergemann et al. 2021), Na (Larsen et al. 2022), Mg (Bergemann et al. 2017), Si (Bergemann et al. 2013; Magg et al. 2022), Ca (Mashonkina et al. 2017; Semenova et al. 2020), Ti (Bergemann 2011), Mn (Bergemann et al. 2019), Fe (Bergemann et al. 2012b; Semenova et al. 2020), Co (Bergemann et al. 2010; Yakovleva et al. 2020), Ni (Bergemann et al. 2021; Voronov et al. 2022), Sr (Bergemann et al. 2012a, Gallagher et al. in prep.) Ba (Gallagher et al. 2020)
- The line list is taken from Heiter et al. (2021) and has been updated with new atomic data for C,N,O, Si, Mg, as described in Magg et al. (2022).

2 Introduction

This document is meant as a short introduction to the v20.0 of Turbospectrum, which is the first to allow NLTE calculations.

Turbospectrum (TS) was developed over the years (Alvarez & Plez 1998; Plez 2012), and the latest LTE version can be found at `https://github.com/bertrandplez/Turbospectrum2019`. The new NLTE TS v.20 (`https://github.com/bertrandplez/Turbospectrum2020`) can also be used in LTE and is back-compatible with v19 (line lists and model formats). More information about NLTE TS v.20 can be found in the paper by Gerber (2022).

In short, Turbospectrum (TS) takes a 1D model atmosphere, one or several line lists, and computes the emergent spectrum (flux and/or intensities at various angles), with a prescribed chemical composition, which should be consistent with the original model composition, although this is not checked by the code. Various parameters can be adjusted: microturbulence velocity, individual abundances, isotopic ratios, computation for a single chunk of spectrum with a constant wavelength step, or for a number of smaller windows, e.g., around lines of interest. Calculations can be done for plane-parallel or spherically symmetric models.

It operates in two step (babsma.f, and bsyn.f, for historical reason):

1. The model atmosphere is read, and the prescribed chemical composition is used to compute the chemical equilibrium, and the continuous opacities (absorption and scattering) for the range of interest. This is saved in a file. The primary thermodynamic variables are T and the electron pressure P_e, and all other are computed from the chemical equilibrium. The space variable is usually the optical depth scale τ_{500} at 500 nm. Some input model format allow however different choices, the density ρ, or a geometrical depth scale, etc (see Sec. 6.1).
2. The continuous opacity file is read, and line opacity is added to it to allow computation of the spectrum. Depending on the setup this is done in LTE or NLTE (continuous opacity is however always in LTE in this version 20.0). NLTE demands more input data (see below).

3 Compiling the code

The source code is in the source/ folder, and the compilation is made inside exec/, by typing make. The makefile is for Intel Fortran ifort. The gfort version in exec-gf currently does not work, as many older Fortran features are not supported anymore, and routines have not yet been updated. Intel Fortran can now be installed at no cost. For historical reasons, there are a few files controlling the dimension of some arrays. Most should not be touched, except (i) the number of layers of the model, and (ii) the number of wavelengths for the spectrum, in case any of them becomes too small. In that case the code will stop with a message like "NDP too small" or "too many wavelengths".

The parameters to be modified (NDP and lpoint) are located in the file "source/spectrum.inc".

4 Control and input parameters

The code needs input data, detailed in Sec. 6, and each run is controlled by a number of parameters (geometry of the model, LTE or NLTE, abundances, ...) These parameters are shown in context in Sec. 5. Parameters are provided using a keyword and its value. See Sec. 5 for the syntax

Keyword	value	comment
LAMBDA_MIN	wavelength in Å	minimum wavelength for synthesis
LAMBDA_MAX	Å	maximum wavelength for synthesis
LAMBDA_STEP	Å	wavelength step for synthesis
METALLICITY	[Fe/H]	global metallicity relative to solar
ALPHA/Fe	[α/Fe]	α-elements abundance[a]
HELIUM	[He/H]	Helium abundance
R-PROCESS	[r/Fe]	r-process fraction enhancement[b]
S-PROCESS	[s/Fe]	s-process fraction enhancement[c]
INDIVIDUAL ABUNDANCES	integer	number of elements with special abundance[d]
MODELINPUT	input model atmosphere	for babsma.f only
MARCS-FILE	T or F	default is true[e]
MODELOPAC	continuous opacity file	output for babsma.f, input for bsyn.f
XIFIX	T or F	true if constant v_{mic} to be read[f]
following parameters for spectral synthesis only in bsyn.f		
RESULTFILE	output spectrum	
SEGMENTSFILE	file name	contains non overlapping spectral segments
RESOLUTION	$\lambda/\Delta\lambda$	spectral resolution in segments is applicable[g]
INTENSITY/FLUX	Intensity or Flux	type of spectrum to be calculated
SPHERICAL	T or F	geometry for radiative transfer[h]
ISOTOPES	integer	number of isotopic abundances[i]
NFILES	integer	number of line lists to be read[j]
NLTE	T or F	NLTE or LTE calculation
NLTEINFOFILE	file name	information on N/LTE species and associated files

[a] O, Ne, Mg, Si, S, Ar, Ca, Ti
[b] see makeabund.f
[c] see makeabund.f
[d] if different from 0, this is followed by lines of data with the element atomic number and its abundance (scale H=12)
[e] ascii or binary MARCS model file
[f] if .true. followed by a line with v_{mic} in km/s, for babsma.f only
[g] default is 500 000
[h] followed by 4 lines of data controlling the solver, that should not be changed
[i] if not 0, followed by lines indicating the isotopic abundances. See makeabund.f for default values
[j] followed by lines with file names

There are some more hidden parameters that can be changed, but safe values have been set by default, and they do not appear in the example scripts.

The default solar abundances are from Grevesse et al. (2007)

```
data sunabund_2007 /
     & 12.00, 10.93,  1.05,  1.38,  2.70,  8.39,  7.78,  8.66,  4.56,   |  1 -  9
     &  7.84,  6.17,  7.53,  6.37,  7.51,  5.36,  7.14,  5.50,  6.18,   | 10 - 18
     &  5.08,  6.31,  3.17,  4.90,  4.00,  5.64,  5.39,  7.45,  4.92,   | 19 - 27
     &  6.23,  4.21,  4.60,  2.88,  3.58,  2.29,  3.33,  2.56,  3.25,   | 28 - 36
     &  2.60,  2.92,  2.21,  2.58,  1.42,  1.92, -99.0,  1.84,  1.12,   | 37 - 45
     &  1.66,  0.94,  1.77,  1.60,  2.00,  1.00,  2.19,  1.51,  2.24,   | 46 - 54
     &  1.07,  2.17,  1.13,  1.70,  0.58,  1.45, -99.0,  1.00,  0.52,   | 55 - 63
     &  1.11,  0.28,  1.14,  0.51,  0.93,  0.00,  1.08,  0.06,  0.88,   | 64 - 72
     & -0.17,  1.11,  0.23,  1.25,  1.38,  1.64,  1.01,  1.13,  0.90,   | 73 - 81
     &  2.00,  0.65, -99.0, -99.0, -99.0, -99.0, -99.0, -99.0,  0.06,   | 82 - 90
     & -99.0, -0.52 /                                                   | 91 - 92
*
```

5 Eaxample script for running the code

Here we separate the script for calculating continuous opacities and the spectrum, but both are integrated in a single example script in the COM/ folder where from the code is run. It is possible to make a single babsma run, covering the full wavelength range, and then run bsyn a number of times, with various wavelength ranges (that should be contained in the continuous opacity file range), various abundances, in LTE and NLTE, including different line lists, etc.

In the case of MARCS models (Gustafsson et al. 2008)[1] the .OPA file can be used, which allows to skip the babsma.f part and run directly bsyn.f.

[1] https://marcs.astro.uu.se

5.1 Running the first part: continuous opacities with babsma.f

```
#!/bin/csh -f
#
# This script is a demo script for NLTE TS (continuous opacity part, babsma.f).
#
set MODEL = atmos.sun_marcs_t5777_4.44_0.00_vmic1_new
set Feabu = 7.50
set lam_min    = '4800.'
set lam_max    = '6800.'
set deltalam   = '0.006'
set METALLIC   = '     0.000'
set TURBVEL = 1.0

../exec/babsma_lu <<EOF
############
# wavelength range for the continuous opacity calculations. Should encompass the full
# range asked for in the following spectrum calculation (bsyn_lu)
# the step is set to 1A in babsma if smaller than 1A here.
#
'LAMBDA_MIN:'  '${lam_min}'
'LAMBDA_MAX:'  '${lam_max}'
'LAMBDA_STEP:' '${deltalam}'
############
# model atmosphere. Various formats allowed. Only MARCS can be binary or ascii.
# Other types of models are ascii
#
'MODELINPUT:' 'TEST-data/${MODEL}'
'MARCS-FILE:' '.false.'
############
# output continuous opacity file providing continuous abs and scatt at all atmospheric depths
# for a set of wavelengths defined by lambda_min/max/step. If the step is < 1A, it is set to
# 1A by default
#
'MODELOPAC:' 'contopac/${MODEL}opac'
############
# Chemical composition. First overall metallicity, then alpha/Fe, Helium/H,
# and r- and s-process
# The latter are scaled according to their solar-system fraction (see makeabund.f)
# finally individual abundances can be provided by first giving how many of them are
# changed and then for each of them their atomic number followed by the absolute
# abundance on the same line.
#
'METALLICITY:'    '${METALLIC}'
'ALPHA/Fe   :'    '0.00'
'HELIUM     :'    '0.00'
'R-PROCESS  :'    '0.00'
'S-PROCESS  :'    '0.00'
'INDIVIDUAL ABUNDANCES:'   '1'
26  $Feabu
############
# if xifix true, fixed microturbulence is read from next line (km/s)
# otherwise the value(s) are read from the model atmosphere.
#
'XIFIX:' 'F'
$TURBVEL
EOF
```

5.2 Running the second part: spectrum calculation with bsyn.f

```
#!/bin/csh -f

#
# This script is a demo script for NLTE TS (bsyn part for spectrum calculation).
#
set MODEL = atmos.sun_marcs_t5777_4.44_0.00_vmic1_new
set Feabu = 7.50

set lam_min    = '4800.'
set lam_max    = '6800.'

set deltalam   = '0.006'
set METALLIC   = '     0.000'
set TURBVEL = 1.0

set SUFFIX     = _${lam_min}-${lam_max}_xit${TURBVEL}-NLTE-windows.spec
set result     = ${MODEL}${SUFFIX}

../exec/bsyn_lu <<EOF
############
# Use NLTE if true. Source function is then computed with departure coefficients
# from departure coefficient file for the atoms in model atom files, if they
# are provided.
#
'NLTE :'          '.true.'
############
# file containing NLTE information
# (species, with LTE/NLTE flags, and associated files)
#
'NLTEINFOFILE:'  'DATA/SPECIES_LTE_NLTE.dat'
#
############
# if present these files will be used to compute the spectrum in a number of
# windows specified in SEGMENTSFILE.
# Comment out if not needed, and the spectrum will be computed from
# LAMBDA_MIN to LAMBDA_MAX
# Segments must NOT overlap
#
'SEGMENTSFILE:'     'TEST-data/uves_giant_Fe-seg.txt'
#
# spectral resolution  to be used in these windows.
# If not specified, a default value of 500000 is used.
# Note that a constant step is set for each window, using this resolution
# if no windows, LAMBDA_STEP is used.
#
'RESOLUTION:'     '300000.'
############
# spectral interval in the case of a single wavelength interval, i.e. no
# segmentsfile.
# min and max lambda for the calculations and constant wavelength step
#
'LAMBDA_MIN:'     '${lam_min}'
'LAMBDA_MAX:'     '${lam_max}'
'LAMBDA_STEP:'    '${deltalam}'
############
# Intensity / Flux
# in the Intensity case, 12 angles are hardcoded, and spectra are computed
# for all of them, in addition to the flux.
#
```

```
'INTENSITY/FLUX:' 'Flux'
###########
# file containing continuous opacity at all model points. Can be computed with babsma.f
# or be a .opa file from the MARCS web site.
#
'MODELOPAC:' 'contopac/${MODEL}opac'
###########
# output file containing spectrum
#
'RESULTFILE :' 'syntspec/${result}'
###########
# Chemical composition. First overall metallicity, then alpha/Fe, Helium/H, and r-
# and s-process
# The latter are scaled according to their solar-system fraction (see makeabund.f)
# finally individual abundances can be provided by first giving how many of them are
# changed and then for each of them their atomic number followed by the absolute
# abundance
# in bsyn.f isotopic fractions can be set, e.g.
# 12.012 0.9
# 12.013 0.1
# to set 90% of 12C and 10% of 13C.
#
'METALLICITY:'    '${METALLIC}'
'ALPHA/Fe   :'    '0.00'
'HELIUM     :'    '0.00'
'R-PROCESS  :'    '0.00'
'S-PROCESS  :'    '0.00'
'INDIVIDUAL ABUNDANCES:'   '1'
26  $Feabu
'ISOTOPES : ' '2'
12.012  0.9
12.013  0.1
###########
# line lists. First how many there are, and then the list of lists
#
'NFILES   :' '2'
TEST-data/nlte_linelist_test.txt
DATA/Hlinedata
###########
# spherical (T) or plane-parallel (F) radiative transfer.
# If spherical, a few more parameters are read
# DO NOT CHANGE THESE PARAMETERS UNLESS YOU REALLY KNOW WHAT YOU ARE DOING.
#
'SPHERICAL:'  'F'
  30
  300.00
  15
  1.30
EOF
```

6 Input data

6.1 Model atmosphere

Various model format can serve as input to basbma.f:

1. MARCS (**?**) binary or ascii models. Binary files are internally used by the MARCS developers, and most users will download ascii formatted models from the MARCS web site[2], the `".mod"` files. In order to use any of these models, bin or ascii, the entry `MARCS-FILE` has to be set to `.true.`.

[2] https://marcs.astro.uu.se

2. ATLAS (Kurucz 1970; Castelli & Kurucz 2003) models[3]. The first line of the file starts with `TEFF`. Example beginning of file:

```
TEFF   5772.  GRAVITY 4.43800 LTE
TITLE  [0.0] VTURB=2  L/H=1.25 NOVER NEW ODF
 OPACITY IFOP 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0
 CONVECTION ON   1.25 TURBULENCE OFF  0.00  0.00  0.00  0.00
ABUNDANCE SCALE   1.00000 ABUNDANCE CHANGE 1 0.92075 2 0.07837
 ABUNDANCE CHANGE  3 -10.99  4 -10.66  5  -9.34  6  -3.65  7  -4.26  8  -3.38
 ABUNDANCE CHANGE  9  -7.48 10  -4.20 11  -5.87 12  -4.51 13  -5.67 14  -4.53
 ABUNDANCE CHANGE 15  -6.68 16  -4.90 17  -6.54 18  -5.86 19  -6.96 20  -5.73
 ABUNDANCE CHANGE 21  -8.99 22  -7.14 23  -8.04 24  -6.40 25  -6.65 26  -4.59
```

3. MULTI (Carlsson 1986) formatted model. The first line of the model is an ascii chain containing the name of the model. Next line starts with either `MASS` (mass-scale model), or with `TAU` (τ-scale model, assuming optical depth at 500nm). Example beginning of file:

```
t5777g44m00_av3DSTAGGER_columnMass_m1dFMT
MASS SCALE
*LOGG
4.44
* NDEP
101
*
  -1.994545       4171.080       4.2364801E+10  0.0000000E+00   1.000000
  -1.944072       4186.400       4.7309701E+10  0.0000000E+00   1.000000
```

4. For all following cases the model file starts with a line like:

```
'blabla' 42 5000. 0.0 0 0
```

where blabla is the model name, or some code, 42 is the number of layers, 5000. is the wavelength for the optical-depth scale, and the last 2 numbers should not be different from0 (otherwise some re-scaling of the model will be made. Possibilities are (NB: units are cgs, except v_{micro} in km/s):

   (a) blabla = a name. A plane-parallel model, containing $\log\tau, T, \log P_e, \log P_{gas}, v_{micro}$. Example:

```
'5200g3.0z-3.80a0.40t1.5.d'      48     5000.    3.00   0   0.
   -0.32000E+01     4176.  -0.242211E+01   0.302086E+01   1.5
   -0.31000E+01     4188.  -0.235561E+01   0.310106E+01   1.5
```

   (b) blabla(1:3) = sph. A spherical model, containing $\log\tau, T, \log P_e, \log P_{gas}$ (dyn/cm^2)/kurucz ,v_{micro} (km/s), r (cm). Example:

```
'sphINTERPOL'  56   5000.  1.03 0 0.00
 -2.9611  3062.73  -5.5103   1.4424   2.0000    0.370778E+13
 -2.8389  3100.07  -5.3779   1.5471   2.0000    0.370243E+13
 -2.7261  3143.44  -5.2431   1.6466   2.0000    0.369728E+13
```

   (c) blabla(1:7) = Stagger. Average <3D> Stagger models (Magic et al. 2013)[4]. The second line contains $T_{\rm eff}$, $\log g$, [Fe/H], number of layers. Following lines contain geometrical depth (cm, increasing inward), T, ρ (g/cm^3).

```
'Stagger'  123  5000. 4.43770 0 0.0
5771.0  4.43769  +0.00  123
 -9.22739e+07      3646.12  1.69219e-10
 -9.07849e+07      3654.31  2.02030e-10
 -8.93067e+07      3681.49  2.40101e-10
```

   (d) blabla(1:4) = alva. Model without optical depth scale: T, P_{gas} (dyn/cm^2), r (cm, increasing outward). The microturbulence velocity v_{micro} (km/s) can be optionally added as a fourth column. Example (without microturbulence):

[3] http://kurucz.harvard.edu/grids.html, https://wwwuser.oats.inaf.it/castelli/
[4] https://staggergrid.wordpress.com

```
'alva'   87     5000.0    0.      0    0.
  1727.3  0.29131092E-12  0.91324519E+14
  1730.2  0.32604626E-12  0.90917730E+14
  1732.7  0.36470422E-12  0.90510941E+14
```

(e) blabla(1:6) = KURUCZ. Reformatted ATLAS model containing ρx (i.e. the mass-scale as in MULTI2.3 ,odels,T, P_{gas} (dyn/cm^3),n_e (cm^{-3}),κ_{Ross} (cm^2/g).

(f) There are a couple more specific format for some special models.

(g) Other formats can easily be implemented.

It is possible to shortcut the continuous opacity calculation run (babsma.f) when using MARCS models, if the corresponding opacity file (`.opa` from the MARCS web site) is used as input to the synthetic spectrum calculation run (bsyn.f).

6.2 Linelists

There is a specific line list for hydrogen, called Hlinedata, which is supplied with the code and it is also suitable for NLTE calculations for H. For all other elements (atoms and molecules) one or several line lists can be provided.

The standard atomic NLTE linelist, based on the Gaia-ESO linelist (Heiter et al. 2021) is supplied with the code and can be downloaded from https://keeper.mpdl.mpg.de/d/a8aa94a4f4ed4bb3b984/. This linelist can also be used in LTE calculations and is backward compatible with v19.0 of Turbospectrum. The linelist contains NLTE label identifications for O, Na, Mg, Al, Si, Ca, Ti, Mn, Fe, Co, Ni, Ba, Sr, in the last 6 columns of the corresponding line data.

The content and format is explained below (verbatim from the DOC/Readme-Linelist_format file).

```
Atomic case:
Example for neutral iron: header, and first 2 lines in the list.
The header contains first a code (26.000 and 1 here) that specifies the element (26),
the isotope (000, being no specific isotope), and the ion (1), and then the number of
lines that are listed (14280).
Next line is the element in character chain. 'NLTE' is actually not used.

'  26.000            '    1     14280
'Fe I    NLTE'
  4200.087  3.884 -1.130   -7.420    7.0  5.01E+07  0.000  'p' 'd'   0.0    1.0
    'Fe I LS:3p6.3d6.(5D).4s.4p.(3P*) z3D* LS:3p6.3d7.(4F).4d f3F'
    82 0  'z3D3*' 'none'  'c' 'x'
  4200.463  4.154 -3.374   -7.290    9.0  2.45E+08  0.000  'p' 'd'   0.0    1.0
    'Fe I LS:3p6.3d7.(4F).4p y5D* LS:3p6.3d6.(5D).4s.\ (6D).5d 7F'
    90 484  'y5D3*' '6s5F'  'c' 'a'

col 1: lambda(A)
col 2: Elow(eV)
col 3: loggf
col 4: fdamp (see below)
col 5: gup
col 6: gamma_rad (if =0, gf-value is used to compute gamma_rad)
col 7: gamma_Stark (may be omitted)
col 8: s,p,d,f etc for upper level (or X), see fdamp
col 9: same for lower level
col 10: equivalent width, when needed (abundance determination in eqwidt run)
col 11: error in eqw
col 12: (in quotes) some text describing levels or whatever you like to include
The following columns are only used in NLTE run:
col 13: lower level number in NLTE model atom
col 14: same for upper level
col 15: lower level id
col 16: same for upper level
col 17: not used (may be omitted)
col 18: not used (may be omitted)
```

```
and for fdamp:
================================

... in the fourth column of the line list (called fdamp):
* 1) use ABO theory (Anstee, Barklem, O'Mara) for collisional damping with H,
*    with data taken from line list: fdamp contains sigma.alpha.
*    This number is available starting with VALD3 version of the VALD database.
*    See : http://www.astro.uu.se/~barklem/howto.html
* 2) if (1) not available check if something can be computed in the anstee.f
*    routine using generic broadening recipes from the ABO formalism
* 3) if (2) not available, check in linelist for a gamma6 at 10000K
* 4) if nothing else worked, compute Unsoeld approximation, using a fudge factor
*    read from column 4.

Example for
case 1:
  6103.649  1.848  0.361  836.274    6.0  1.05E+08 'p' 'd'   0.0    1.0
    'Li I LS:1s2.2p 2P* LS:1s2.3d 2D'
case 2:
  8858.071  3.879 -4.036    2.500    4.0  6.76E+07 'd' 'p'   0.0    1.0
    'Li I LS:1s2.3d 2D LS:1s2.11p 2P*'
case 3:
  4305.083  7.685 -4.221   -7.160    5.0  4.79E+08 'x' 'x'   0.0    1.0
    'C I LS:2s2.2p.3s 1P* LS:2s2.2p.5p 3P'
case 4:
  8858.071  3.879 -4.036    2.500    4.0  6.76E+07 'x' 'x'   0.0    1.0
    'Li I LS:1s2.3d 2D LS:1s2.11p 2P*'

One example:
  5349.465  2.709 -0.428   -7.652    7.0  2.69E+06 's' 'p'   0.0    1.0
    'Ca I LS:3p6.3d.4s 1D LS:3p6.3d.4p 1F*'    8   26  '13D1D'       '14P1FP'

fdamp = -7.652 :  the calculation will be made according to the generic  ABO recipe
for s-p transition (case 2).
If I replace 's' 'p' by 'x' 'x' in the line list, broadening will be computed using
-7.652 and the gam_6 recipe from Kurucz (case 3),
If in addition I replace -7.652 by 1.0, The Unsoeld recipe with a fudge factor = 1
will be used.
================================
```

Additional line lists can be constructed from VALD[5] extractions, which are turned into the TS format using Utilities/vald3line-BPz-freeformat.f. In that case, the NLTE identifications are not included in the list. NLTE labels can be added, but that ijmplies matching energy levels in the list with energy levels in the corresponding model atom.

Molecular data can also be downloaded from https://nextcloud.lupm.in2p3.fr/s/r8pXijD39YLzw5T, with part of the data needing formatting using code that can be found on https://nextcloud.lupm.in2p3.fr/s/r8pXijD39YLzw5T?path=%2FPROGRAMS.

6.3 NLTE case

If the `NLTE` flag is `.true.`, extra input is needed for the relevant atoms: model atom, level identifications in the line list, and departure coefficient file for the relevant model atmosphere.

Model atoms and the associated grids of NLTE departure coefficients are available at https://keeper.mpdl.mpg.de/d/fa139c65b650428cb931/.

To avoid heavy data transfer, we recommend to download the data element by element. If the user intends to use the MARCS model atmospheres only, the files with the 'MARCS' keyword in the filename are needed.

[5] http://vald.astro.uu.se

6.3.1 NLTE info file

Control over inclusion of NLTE data into NLTE TS v.20 is done via a separate input file (provided with a key 'NLTEINFOFILE' in bsyn input). For each element one needs to provide atomic number, identification, whether the elements needs to be treated in LTE or Non-LTE, and in the latter case, the model atom and the departure coefficient files as described in 6.3.4. By default, elements not found in this file are treated in LTE.

An example such file could look like:

```
# some comment
# path for model atom files ! this comment line has to be here !
./modelAtoms/
# path for departure files ! this comment line has to be here !
./nlteDepFiles/
# atomic (non)LTE setup (another comment)
6.0   'C'  'lte'  ''          ''                       ''
8.0   'O'  'nlte' 'atom.o41f'  'depCoeff_O_9.591.dat'  'ascii'
12.0  'Mg' 'nlte' 'atom.mg86b' 'depCoeff_Mg_7.428.dat' 'ascii'
14.0  'Si' 'lte'  ''          ' '                      ''
```

6.3.2 Model atoms

A model atom is an ascii file formatted for MULTI2.3. It must be the model atom that was used to generate the departure coefficients. The model atoms can be downloaded along with the departure coefficient files.

6.3.3 grids of departure coefficients

There is one binary file calculated for each model atom across a grid of model atmospheres. It can be interpolated for the desired stellar parameters using the code included in the folder `interpolator/`. We provide a single binary grid of departure coefficients for each chemical element and each type of model atmosphere, i.e. MARCS or average 3D STAGGER. A binary grid of departure coefficients (hereafter NLTE grid) is a dense data file containing departure coefficients for each energy state of the chemical element at each depth point in each model atmosphere at a range of chemical element abundances, usually within [-2, +1] dex around the central solar-scaled value (step 0.1 dex). Each binary NLTE file is accompanied by a human-readable auxiliary ascii file, where one can find information about each entry. Strictly speaking, one does not need to read the whole NLTE grid to produce a single spectrum, as the format of the NLTE grid allows to read a specific entry and therefore save memory and computational resources. To do so use the pointer provided in the auxiliary file (last column) to locate the record in the binary file. FORTRAN and PYTHON examples of how the NLTE grid can be read are provided with the data. The model interpolator included with the NLTE TS v.20 is also capable of reading NLTE grids.

6.3.4 departure coefficient file

Once a desired set of departure coefficients has been extracted (either directly from a NLTE grid or via interpolating the grid) one needs to write the departure coefficients into NLTE TS v.20 formatted input file. See, e.g. routines available in the PYTHON wrapper, Sect. 8 for details.

6.3.5 Interpolator of departure coefficients

An interpolator is included for convenience, in order to allow calculations for models between grid points. It is based on the interpolator created by Thomas Masseron and available on the MARCS web page. It was extended in order to interpolate the departure coefficient files together with the atmosphere model structure using the same interpolation weights. These weights were devised by Masseron to minimize interpolation errors.

7 Output

If all goes well, the output from babsma.f is a file containing the continuous absorption and scattering opacity for all model depths along the desired wavelength range. The output from bsyn.f is the spectrum (flux and/or intensity). The flux is the surface flux F_λ, such that $\int_0^\infty F_\lambda d\lambda = \sigma T_{\rm eff}^4$ In the spherical case,

the flux is normalized at the radius where $\tau_{\rm Ross} = 1$, when the Rosseland optical-depth scale is known. Otherwise τ_{500} is used (500 nm).

There are a number of cases, depending on the `INTENSITY/FLUX:` and `SPHERICAL:` flags.

- `INTENSITY/FLUX: 'Flux'`, and `SPHERICAL: 'F'`:

  Output consists of wavelength (Å), normalised flux (F_λ/F_{cont}), and the flux F_λ (erg/s/cm^2/Å). Example:

```
7050.000     0.99555   2.32544E+06
7050.010     0.99555   2.32544E+06
7050.020     0.99540   2.32509E+06
```

- `INTENSITY/FLUX: 'Intensity'`, and `SPHERICAL: 'F'`:

  The first line starting with a `#` gives the $\mu = \cos\theta$ angles

```
# mu-points    1.001800E-02  5.203500E-02  1.246190E-01  2.228410E-01
    3.400080E-01  4.681380E-01  5.984970E-01  7.222030E-01  8.308250E-01
        9.169580E-01  9.747260E-01  1.000000E+00
```

  The following lines list the wavelength (in Å), the normalised flux (F_λ/F_{cont}), the flux F_λ (erg/s/cm^2/Å), and couples I_λ, I_λ/I_{cont} for all μ-angles.

```
4000.000     0.99798   1.07519E+07   6.33247E+05   0.99987   8.77758E+05
 0.99994   1.23113E+06   0.99997   1.65761E+06   0.99993   2.14819E+06
     0.99978   2.66914E+06   0.99944   3.18446E+06   0.99891   3.65962E+06
         0.99823   4.06532E+06   0.99752   4.37921E+06   0.99690   4.58613E+06
             0.99646   4.67463E+06   0.99626
```

- `INTENSITY/FLUX: 'Flux'`, and `SPHERICAL: 'T'` :

  Output identical to corresponding PP case.

- `INTENSITY/FLUX: 'Intensity'`, and `SPHERICAL: 'T'` :

  Output identical to corresponding PP case. Note that limb ($\mu = 0.$) is defined as the outermost layer of the model.

8 PYTHON wrapper

To ease the start of using NLTE TS v.20 we provide a PYTHON wrapper that can be found under: https://github.com/EkaterinaSe/TurboSpectrum-Wrapper. The wrapper allows to compute model spectra at requested stellar parameters (Teff, log(g), [Fe/H], $v_{\rm mic}$) as well as abundances of individual chemical elements. The computations can be done in NLTE, LTE, or combination of the two. All major configuration options provided by TS (see Sect. 4) are supported.

Requested parameters *might* but *do not have to* match the grid points in the grid of model atmospheres, and consequently, in the grids of departure coefficients. If requested parameters do not correspond to an existing model atmosphere and corresponding NLTE solution, interpolation based on Delaunay triangulation will be performed.

We emphasize that this wrapper was designed to be used primarily for large scale computations, e.g. to compute an extensive grid of model spectra for training a neural network or produce a compact but dense grid for fitting observed spectra. With that in mind, this wrapper is most efficient (in terms of time and computational resources) for computing large spectral datasets and less efficient for producing individual spectra.

The wrapper supports parallelisation within one computing node.
For the detailed description of the code please head over to TurboSpectrum-Wrapper documentation.
If you use this wrapper, please cite Magg. et.al. 2022 We would also appreciate if you cite the source of the NLTE model atom, see Sect. 1.

References

Alvarez, R. & Plez, B. 1998, , 330, 1109

Bergemann, M. 2011, , 413, 2184

Bergemann, M., Collet, R., Amarsi, A. M., et al. 2017, , 847, 15

Bergemann, M., Gallagher, A. J., Eitner, P., et al. 2019, , 631, A80

Bergemann, M., Hansen, C. J., Bautista, M., & Ruchti, G. 2012a, , 546, A90

Bergemann, M., Hoppe, R., Semenova, E., et al. 2021,

Bergemann, M., Kudritzki, R.-P., Würl, M., et al. 2013, , 764, 115

Bergemann, M., Lind, K., Collet, R., Magic, Z., & Asplund, M. 2012b, , 427, 27

Bergemann, M., Pickering, J. C., & Gehren, T. 2010, , 401, 1334

Carlsson, M. 1986, Uppsala Astronomical Observatory Reports, 33

Castelli, F. & Kurucz, R. L. 2003, in Modelling of Stellar Atmospheres, ed. N. Piskunov, W. W. Weiss, & D. F. Gray, Vol. 210, A20

Gallagher, A. J., Bergemann, M., Collet, R., et al. 2020, , 634, A55

Gerber, J. 2022

Grevesse, N., Asplund, M., & Sauval, A. J. 2007, , 130, 105

Gustafsson, B., Edvardsson, B., Eriksson, K., et al. 2008, , 486, 951

Heiter, U., Lind, K., Bergemann, M., et al. 2021, , 645, A106

Kurucz, R. L. 1970, SAO Special Report, 309

Larsen, S. S., Eitner, P., Magg, E., et al. 2022, , 660, A88

Magg, E., Bergemann, M., Serenelli, A., et al. 2022, arXiv e-prints, arXiv:2203.02255

Magic, Z., Collet, R., Asplund, M., et al. 2013, , 557, A26

Mashonkina, L., Sitnova, T., & Belyaev, A. K. 2017, , 605, A53

Mashonkina, L., Zhao, G., Gehren, T., et al. 2008, , 478, 529

Plez, B. 2012, Turbospectrum: Code for spectral synthesis

Semenova, E., Bergemann, M., Deal, M., et al. 2020, , 643, A164

Voronov, Y. V., Yakovleva, S. A., & Belyaev, A. K. 2022, , 926, 173

Yakovleva, S. A., Belyaev, A. K., & Bergemann, M. 2020, Atoms, 8, 34