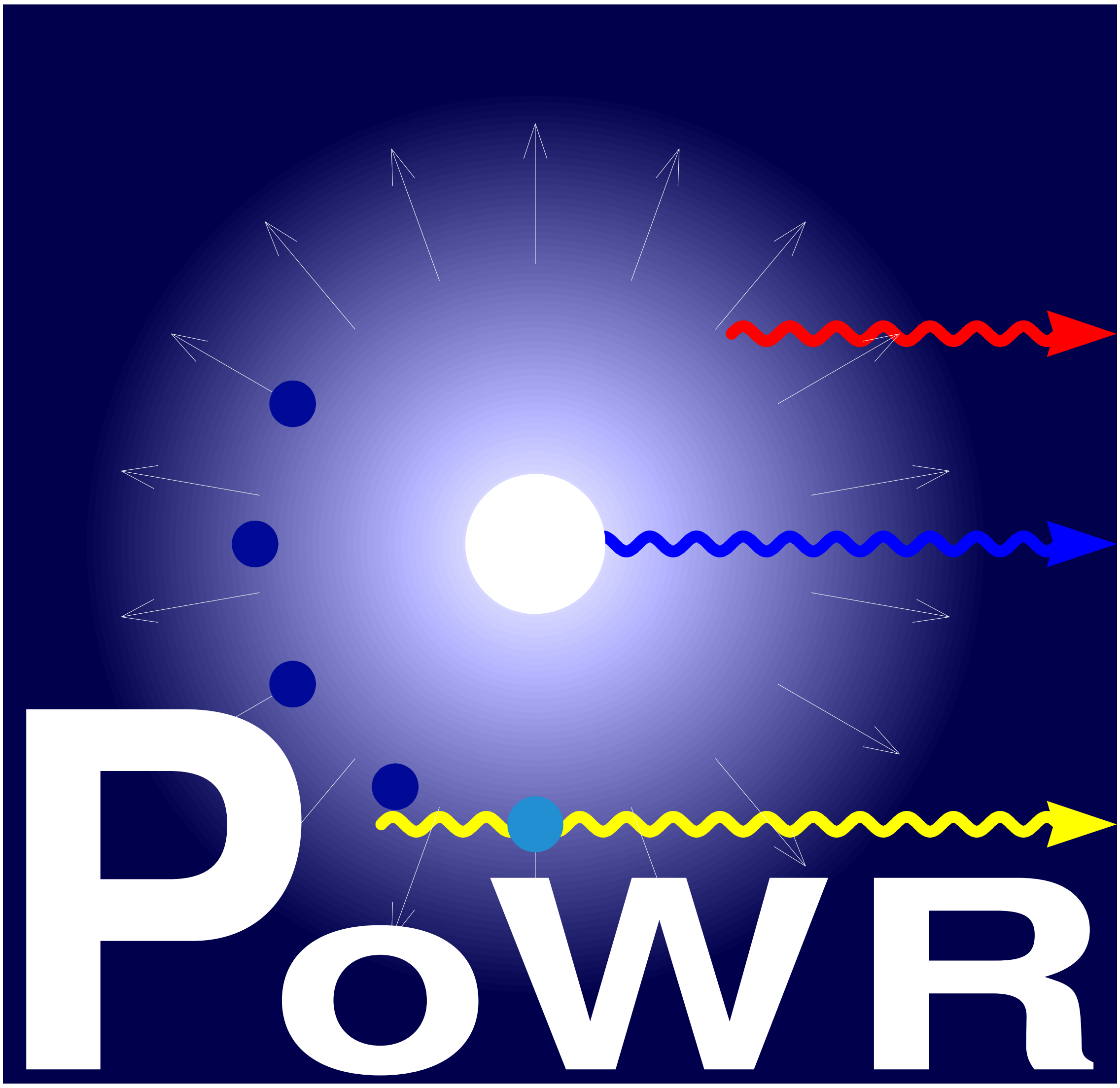
ManPoWR
– the PoWR code manual



April 25, 2025

The PoWR logo symbolizes a a star (white disc) with its wind, with the white arrows indicating its expansion. The colored light rays stand for the emergent radiation. The blue dots resemble the Logo of the University of Potsdam, but can also be understood as clumps in the wind.

Preface

The main purpose of PoWR is to simulate the emergent spectrum of a hot star with given stellar parameters. The models have been successfully applied so far for stars of spectral types O, B, early A, Wolf-Rayet, central stars of planetary nebulae, subdwarfs, extreme helium stars.

Basic assumptions of the code are:

- spherical symmetry;
- stationarity;
- Pre-specified wind velocity law and mass-loss rate.

Physics that is taken into account comprises:

- non-LTE radiative transfer;
- detailed model atoms with up to ∼1000 explicit non-LTE levels;
- iron-group elements with millions of lines in a superlevel approach;
- inhomogeneities on small scales (“microclumping”);
- embedded X-ray sources.

Furthermore, the computation of the emergent spectrum (“formal integral”) can optionally account for:

- pressure broadening of spectral lines;
- inhomogeneities on large scales (“macroclumping”, “porosity”);
- wind rotation;
- a “second model”, which fills a specified part (double-cone or embedded sphere) of the atmosphere’s volume.

Credits. The code has been developed since the late 1970s under the guidance of Wolf-Rainer Hamann, first in Kiel and since 1994 in Potsdam. Many students and colleagues contributed to the development of the code and its various tools: including (in roughly chronological order): Werner Schmutz, Ulf Wessolowski, Gerhard Dünnebeil, Uwe Leuenhagen, Lars Koesterke, Helge Todt, Götz Gräfener, Wolfgang Leindecker, Sonja Burgemeister, Martin Steinke, Tomer Shenar, Andreas Sander.

Contents

I. Operation Manual 9

1. Structure of PoWR 10
1.1. Program flow 10
1.2. The concept of different *chains* 11
1.3. Input files 11
1.4. Output files 11
1.5. Further directories 12

2. Input files 13
2.1. CARDS 13
2.2. FGRID 17
2.3. MODEL 18
2.4. DATOM 18
2.5. FORMAL_CARDS 18
2.6. FEDAT and FEDAT_FORMAL 19

3. How to run a model 20
3.1. Preparing the input files 20
3.2. Submit the model calculation 20
3.3. Supervising model calculations 21
3.4. Save models 22

4. Command-line tools 23
4.1. Job submission 23
4.2. Job status 23
4.3. Load an existing model 24
4.4. Save a model 24
4.5. Create an additional chain 24
4.6. Clone a chain 25
4.7. Reading from the MODEL file 25

5. Troubleshooting 26
5.1. Job submission has no effect 26
5.2. DIMENSION INSUFFICIENT 26
5.3. Solution of the Rate Equations fails at some depth points 26
5.4. Global convergence problems 27
5.5. Flux conservation problems 27
5.6. Manual intervention: job `modify` 28
5.7. Restart converged model 29
5.7.1. Do your model atoms cover the ionization stages as needed? 29

6. Providing atomic data 30
6.1. Assembling a new DATOM file 30
6.2. Assembling a new FORMAL_CARDS file 32
6.3. Harvesting atomic data from the Opacity Project (OP) 33
6.3.1. Source code 33
6.3.2. Input files 33

6.4. Remarks 34
6.4.1. Line strengths 34

7. All CARDS options in alphabetical order 37
AB 37
CD 38
EFG 40
HIJ 42
KLM 43
NO 45
P 48
R 53
ST 54
UV 56
WXYZ 59

8. Syntax of the DATOM file 61
8.1. LEVEL entries 61
8.2. LINE entries 62
8.3. CONTTINUUM entries 62
8.4. K-SHELL entries 63
8.5. DRTRANSIT entries 63
8.6. Iron-group elements 63

9. FORMAL_CARDS 64
9.1. Specifications for the formal integral 64
9.2. FORMAL_CARDS atomic data syntax 73

II. Theory and Algorithms 75

10. Overview on the model atmosphere theory 76
10.0.1. wrstart 76
10.1. Job: wruniq 76
10.1.1. Programs 77
10.2. The aim: emergent spectrum 77
10.3. The way: model atmosphere 78

11. WRSTART: Setting up a model 79
11.1. Radius grid 79
11.2. Velocity field 81
11.2.1. Hydrostatic domain 81
11.2.2. Wind domain 83
11.2.3. Connection point 85
11.3. The coarse frequency grid 87
11.3.1. User-defined frequency points 87
11.3.2. Criteria-based frequency points 87
11.3.3. Removal of extremely small grid steps 89
11.3.4. Calculation of integration weights 89

12. Radiative transfer in the co-moving frame 91

12.1. “Ray-by-ray”: the angle-dependent transfer equation . . . 91
12.2. Short-characteristic integration . . . 92
12.3. Integration of the moments . . . 101
12.4. Moment equations . . . 101
12.5. Cancellation of the Thomson-scattering term . . . 103
12.6. Eddington factors . . . 103
12.7. Solution of the moment equation by a differencing scheme . . . 104
12.7.1. Inner points . . . 104
12.7.2. Boundary in space . . . 108
12.7.3. Boundary in frequency . . . 109
12.7.4. Calculation of H . . . 110
12.7.5. The coefficients of the linear equations . . . 110
12.8. Solution of the tri-diagonal system for J . . . 113
12.8.1. Extraction of the Λ^*-operator . . . 114
12.9. Overall scheme for solving the radiative transfer . . . 115

13. Rate equations 116

13.1. Solving the Rate Equations . . . 116
13.2. Implementation of the Accelerated Lambda Iteration (ALI) . . . 117
13.2.1. Lines . . . 118
13.2.2. Generic (iron group) lines . . . 119
13.2.3. Continua . . . 120
13.2.4. Additional derivative terms resulting from ALI . . . 120

14. The temperature stratification 126

14.1. Temperature corrections from radiative equilibrium . . . 126
14.2. Temperature corrections from thermal balance . . . 127
14.2.1. Free-Free transitions . . . 128
14.2.2. Bound-free transitions . . . 129
14.2.3. Collisions . . . 130
14.3. Connection between radiative equilibrium and thermal balance . . . 131

15. Formal Integral: radiative transfer in the observer’s frame 132

15.1. Wavelength and frequency grid . . . 132
15.2. The emergent flux . . . 133
15.3. The emergent intensity . . . 133
15.4. Frequency redistribution by electron scattering . . . 134
15.5. Line broadening . . . 135
15.5.1. Radiation damping . . . 136
15.5.2. Tabulated profiles: H I . . . 137
15.5.3. Tabulated profiles: He II . . . 138
15.5.4. He I . . . 138
15.5.5. Linear Stark effect . . . 138
15.5.6. Quadratic Stark effect . . . 139
15.5.7. Bandwidth estimate . . . 139
15.6. Wind Rotation . . . 139
15.7. Combining two models . . . 141
15.7.1. Cone model . . . 141
15.7.2. Sphere model . . . 146

15.7.3. Visualization 148
15.7.4. Implementation details 149
15.8. Program Structure 151

16. Atomic transitions 154
16.1. Collisional bound-bound transitions 154
16.1.1. Introduction 154
16.1.2. Specification of the collisional bound-bound data 155
16.1.3. Formulas for collisional bound-bound collision strengths 155
16.1.4. Summary of Subroutines and keywords 159
16.1.5. Defaults and recommendations 159
16.1.6. PLOTCBB - a plotting tool 160
16.2. Collisional bound-free transitions 160
16.3. Radiative line transitions 161
16.4. Radiative bound-free transitions 162
16.4.1. Detailed OP photoionization cross sections, e.g. C II 163
16.4.2. Formulas for the bound-free photo cross sections 164
16.5. K-shell ionization 168
16.6. Dielectronic recombination 171
16.7. Radiative free-free transitions (Bremsstrahlung) 173
16.7.1. Implementation 173
16.7.2. Additional X-rays for superionization 173
16.8. Iron-group elements 175
16.8.1. Superlevel approach 175
16.8.2. The FEDAT files 180

III. Organization of the Code 183

17. The MODEL file 184

18. Reading atomic data 184

19. Initialization 186

20. Radiative transfer 186

IV. Installation Manual 188

21. PoWR-Code-Skripte 189
21.1. wrstart8 189
21.2. wruniq8 189
21.3. set_repeat8 189
21.4. njn8 189

22. List of routines 189

23. Compilation of PoWR code 190
23.1. The way it works - Example 190

A. Atomic Theory **191**
A.1. Levels – energy order 191
A.2. L-S coupling 191
A.2.1. Addition of angular momenta 192
A.2.2. Selection rules for dipole radiation 192

B. Quadrature sums **193**
B.1. General 193
B.2. Linear weight function 194
B.3. Quadratic weight function 194
B.4. Cubic weight function 194
B.5. Inverse-*x* weight function 195
B.6. Exponential weight function 195

C. Interpolation with cubic splines **196**

D. Characteristics of the Moment Equations **197**

E. Conversion, constants, formulae **201**
E.1. Constants and conversions 201
E.2. Formulae 201

F. Contour plots for model grids **203**
F.0.1. How to create contur plots 203

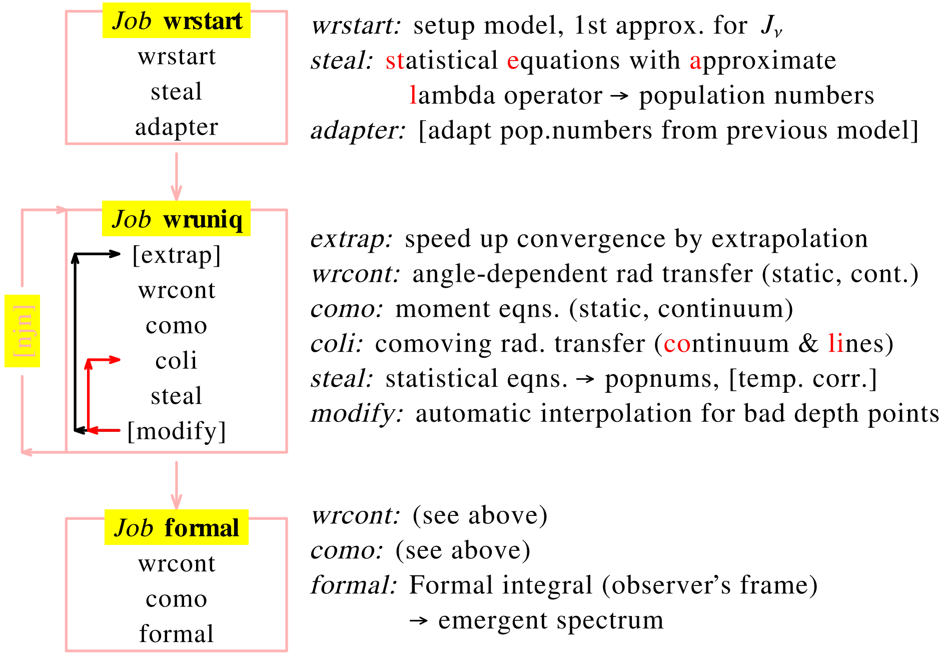
V. References **206**

Part I.
Operation Manual

1. Structure of PoWR

1.1. Program flow

Calculating a PoWR model means to run a sequence of shell-scripts, called *jobs* (see Fig. 1.1. Each of these jobs executes a sequence of programs. The individual programs communicate with each other via the file MODEL, which is a name-indexed random-access mass-storage file. If everything goes smooth, only the first job **wrstart** must be submitted by the user, while the further sequence `wrstart` → `wruniq` → `formal` then runs automatically. Launching and controlling this sequence will be explained below (Sect. 4.1).


Job wrstart
wrstart
steal
adapter
wrstart: setup model, 1st approx. for J_ν
steal: statistical equations with approximate
lambda operator → population numbers
adapter: [adapt pop.numbers from previous model]
Job wruniq
[njn]
[extrap]
wrcont
como
coli
steal
[modify]
extrap: speed up convergence by extrapolation
wrcont: angle-dependent rad transfer (static, cont.)
como: moment eqns. (static, continuum)
coli: comoving rad. transfer (continuum & lines)
steal: statistical eqns. → popnums, [temp. corr.]
modify: automatic interpolation for bad depth points
Job formal
wrcont
como
formal
wrcont: (see above)
como: (see above)
formal: Formal integral (observer's frame)
→ emergent spectrum

- Job **wrstart**

  Program wrstart sets up the model (radius and frequency grids, velocity and density structure, first approximations for the radiation field).

  Program steal calculates a first approximations for the population numbers.

  Program adapter overwrites these population numbers (or part of them) with popnumbers from a previous model, if requested.

- Job **wruniq**

  Program extrap is an optional tool to accelerate the iteration by a Ng extrapolation; the use of the corresponding option is currently discouraged (cf. Sect. **??**).

  Program wrcont is an unimportant program, calculating the continuum radiation transfer without expansion.

  Program como uses the Eddington factors from *wrcont* and calculates the moment equations for the continuum radiation transfer without expansion; the resulting continuum radiation field is only used for start approximations.

Program coli solves the radiative transfer in the co-moving frame (see Sect. 12). Emissivities and opacities are based on the population numbers.

Program steal solves the " rate equations" (equations of statistical equilibrium), based on the current radiation field (see Sect. 13. *steal* might also update the temperature structure and/or the density/velocity stratification.

The programs within the *wruniq*-Job are repeated in a cycle until the iteration is converged, or until a specified maximum number of program runs is exceeded. If converged, the *formal*-job is automatically started.

- Job **formal**

  Programs *wrcont* and *como* for some initializations

  Program *formal* calculates the emergent spectrum, using the population numbers as established by the *wruniq* iteration.

1.2. The concept of different *chains*

The sequence of jobs working on one specific model form a *chain*. Each user can run different chains in parallel. Each chain has its specific chain number *n*. A specific chain be submitted to any computer ("machine") of the cluster (see Sect. 4.1) and stays there until it finishes, or till it is manually moved to another machine (see Sect. 3.3).

If a chain number *n* has never used before, the necessary files and directories must be created by typing
`makechain` *n*
which executes the shell script `$USER/work/dummychain/makechain.bash`

1.3. Input files

Various *input files* must be prepared in the directory `work/wrdata`*n* before the corresponding model chain can be launched:

- CARDS (see Sect. 2.1)
- FGRID (see Sect. 2.2)
- DATOM (see Sect. 2.4)
- FORMAL_CARDS (see Sect. 9)
- FEDAT (see Sect. 2.6)
- FEDAT_FORMAL (see Sect. 2.6)
- MODEL (needed as input only in case the new model shall start from it)

The MODEL file is also saved in this directory after `wrstart` and after the `wruniq` iteration has stopped in a regular way, either because the model is converged, or the JOBNUM counter exceeded the limit, or the chain has been manually interrupted by a `stat stop` or `stat break` command (see Sect. 3).

Note that there is a shortcut for the shell command to change into `wrdata`*n* as present work-directory:
`cddat` *n*

1.4. Output files

In the directory `~/work/output/` the output files are collected from all chains. The different chains are distiguished by their chain number as part of the filenames, e.g. `wrstart3.out`. These files are:

- `wrstart`*n*`.out`
  output listing (ASCII), 132 characters wide, from the `wrstart` job. It contains the set of input parameters, the radius- and frequency grid, and information from the `adapter` program which atomic levels have been identified between the old and the new models if their atomic energy level list differs.
- `wrstart`*n*`.plot`
  contains plots requested from the wrstart job. All files with the extension `.plot`, are in the WRplot format and can be directly viewed with that program (see the WRplot manual).
- `wruniq`*n*`.out`
  output listing (ASCII), 132 characters wide, from the `wrstart` job. It contains a short summary of each iteration cycle.
- `wruniq`*n*`.plot`
  contains plots requested from the wruniq job (last iteration).
- `formal`*n*`.out`
  output listing (ASCII), 132 characters wide, from the `wrstart` job. It contains long lists of all spectral lines which have been taken into account for each of the requested wavelength ranges.
- `formal`*n*`.plot`
  contains the plots requested from the formal job. These are usually the emergent spectra for each of the requested wavelength ranges.

The above-mentioned output files are copied inthe the work/output/ directory only after the corresponding job has been successfully finished.

In contrast, there are further output files of each program (e.g. wrstart, wruniq, formal) which are continuously written by the active programs (even when running on a different machine). These files carry the extension

`.cpr` – detailed protocol file, error messages
`.log` – short logfile with time stamps, error messages

1.5. Further directories

- The directory `~/work/wrjobs` contains the shell scripts for all jobs. If everything goes smooth, the user does not need to care about them.
- The `wrstart`-job copies all necessary files and executables into the directory `~/work/scratch/wrstart`*n*. The output is collected there, until the job is finished.
- The `wruniq`-job copies all necessary files and executables into the directory `/home/$machine/tmp_data/$user/wruniq`*n*. Here, `$machine` stands for the computer to which this chain has been sent, and `$user` is the username. The output is collected in that directory until the job is finished or interrupted, and then copied back.
- The `formal`-job copies all necessary files and executables into the directory `~/work/scratch/formal`*n*. The output is collected there, until the job is finished.
- The `wruniq`-job maintains a further directory on the machine where the chain is executed: `/home/$machine/tmp_data/$user/ass`*n*. It contains a (possibly very large) file with the Eddington factors established and needed by the program `coli`, and the `DMFILE` used for the Broyden method.

  These files are not copied when the chain moves to a different machine, nor when the model is converged and saved. If wruniq cannot find the corresponding files, they will be generated automatically with limited extra efforts.

2. Input files

2.1. CARDS

The file CARDS has multiple functions:

- defining the parameters of the model;
- setting various parameters for numerical details of the model iteration.

Note that the second class of CARDS lines are re-read from the programs in the wruniq-job in each iteration,. Thefore they can be changed in the course of a running model computation. If the CARDS file has been edited in the wruniq*n* directory, the changes must be uploaded to the running chain by the command:
`stat break wruniq`*n*

The files CARDS is read by all programs which are executed in the wrstart-job and in the wruniq-job. Each program interprets only those commands which it can recognize by the first word of the line; all other lines are ignored. Therefore, lines can be commented out by just inserting one additional character at the beginning, like - or *. The same holds (in most cases) for optional parameters of commands: if their keyword is not recognized, the corresponding parameter is ignored.

Valid lines must start with the first character of the line; indention is not allowed. The paramaters which follow the first keyword ra separated by any seperatorr ot of `, = / :` and/or blanks, and they sequence is arbitrary (in most cases).

The lines in the CARDS file do not have to be sorted in any way – their sequence is arbitrary. However, if a command which sets a certain parameter occurs multiple times, the last occcurence overwrites the previous ones. This holds not only for commands with identical keywords, but also for those which are alternative to each other, e.g. defining the stellar mass by specifying it directly or via the surface gravity.

A huge number of possible commands exist for the CARDS file. A full list, sorted alphabetically, is given in Sect. 7. Most of them you will never need to use. In the following we describe a minimalistic example of a CARDS file. This file can be found under `CARDS.template` in the manpowr.dir directory and can be copied from there.

```
===================================================================
HEADLINE: 200kK/dex-0.5/2000 L=5.3 C.4 O.05 D10 Fe1.6E-3 WC
===================================================================
```

This statement defines a text string (max. 65 characters), by which the user can identify the model. The program will print a time stamp (date and time when the model was started) in front, and will issue this header line in all output listings and plot files.

```
--------------------  oldstart options  --------------------------------
OLDSTART
OLD T TAU
LTESTART
-JSTART BLACKEDGE=228.
-TMIN_START= 8000.
```

The above block defines the start approximation for the model to be calculated. OLDSTART means that the MODEL file that is currently in the wrdata*n* directory serves as start approximation for the population numbers. Starting from a similar model greatly facilitates the calculation. If, however, the new model shall be calculated with different atomic data than those used when calculating the old MODEL, special LEVEL lines for the CARDS file must be prepared with the help of the tool `levelcards` (see Sect. 3).

With OLD T TAU the temperature structure is also taken from the old model, scaled to the new stellar temperature, and interpolated over an optical depth scale (recommended). LTESTART or JSTART specify alternatively the start approximation for the popnumbers or the radiation field, respectively. In case of an OLDSTART, these popnumbers are anyhow overwritten by the old model for all atomic levels that both models have in common.

```
-------------------- model parameters
TEFF= 200000.
LOG L=5.3
-RSTAR=10.
RTRANS=-0.5DEX
-MDOT = -6.6
LOG GGRAV = 3.7
-MSTAR=0.6
VDOP=100.
DENSCON = 10.
```

The previous block of lines specifies the stellar parameters in the usual units. Obviously, only two out of the three parameters $L, T_{\rm eff}, R_*$ can be specified independently (Stefan-Boltzmann law). The mass-loss rate can be specified either directly (MDOT as logarithm of solar masses per year), or in the form of the so-called *transformed radius* RTRANS where the directly appended DEX means that the given value is a logarithm.

The stellar mass can be specified directly or via the surface gravity. If none of them is given, the code guesses the mass on the basis of mass-luminosity relations, etimating the type of star from the chemical composition.

VDOP is the Doppler-broadening which adopted for establishing the model stratification. It accounts for turbulence and for unresolved multiplets. It can be overwritten later for calculating the emergent spectrum. Note that with smaller VDOP the computation times grow dramatically, while the results are usually not much affected.

DENSCON is the clumping factor which can be specified as radially dependent (see Sect. 7).

```
--------------------  abundances  -------------------------------------
CARBON:    4.0E-1   (mass fraction)
OXYGEN:    5.0E-2   (mass fraction)
GENERIC:   1.6E-3   (mass fraction)
```

This block of lines specifies the chemical abundances of the respective elements, i.e. oxygen and carbon in this example. The remaining mass fraction is assumed to be helium.
If the string "mass" is detected in the line with the element's name, the number is interpreted as mass fraction. Otherwise, the given number is interpreted as number fraction. Chemical abundance must be specified all *either* as mass fractions *or* as number fractions; a mixture of both formats is not allowed and leads to an error stop.
GENERIC refers to the sum of all iron-group elements, for which the relative composition has been specified already when the corresponding data have been assembled; these elements are treated in the *superlevel approach* (see Sect. **??**).
Abundances must be specified for *all* elements which occur in the atomic data file DATOM, and vice versa.

```
----------- velocity field and quasi-hydrostatic part ------
RMAX_IN_RSUN
VELPAR: VFINAL (KM/S)= 2000. VMIN=1.6 BETA=1.0 RMAX=1000.
HYDROSTATIC INTEGRATION FULL REDUCE=0.3
```

The VELPAR line defines the parameters of the wind velocity field. Default is the so-called *beta law*. VFINAL will be reached at RMAX. VMIN is an initial guess for the velocity at the inner boundary, and will be overwritten when the model iterates to achieve a specified TAUMAX (see below). RMAX defines the location of the outer boundary, as given in units of the stellar radius (default) or in units of solar radii if the preceeding line RMAX_IN_RSUN requests so.

HYDROSTATIC INTEGRATION invokes a detailed treatment of the density stratification in the lower, sub-sonic and thus quasi hydrostatic part of the atmosphere. With the parameter FULL the hydrostatic equation is integrated accounting for the radiation pressure. This sophisticated treatment of the static is not necessary for WR-type atmospheres, but recommended for stars with photospheric absorption lines.

```
--------------------  grids in radius and frequency  ----------------------------
RGRID: ND= 50. NDDENS=  8. NDVELO= 4.  DLOGTAU= 5.0
BLUEMOST-WAVELENGTH = 5.
```

The RGRID line specifies the depth grid. ND is the total number of depth points; which is usually chosen berween 50 und 70. The other parameters influence their distribution (see Sect. **??**). The BLUEMOST-WAVELENGTH of the coarse frequency grid must be given in Å.

Note: the options described in the following influence the numerics of the model iteration. They can be edited "on the fly", but changes become only active after they were uploaded with the command
stat break wruniq*n*

```
--------------------  boundary conditions  ----------------------------
TAUMAX=20. FIX EPS=0.01 REDUCE=0.3 CORRLIMIT=-0.5 SAFE
-OB-VERS 0
NOTDIFFUS
```

TAUMAX specifies the Rosseland-mean continuum-only optical depth at which the inner boundary is placed at model start (usually 20). The FIX parameter ensures that the final model still has this optical depth at the inner boundary. The corresponding update of the radius grid is numerically delicate, and the further parameters regulate this iteration.

```
-------------- Temperature corrections, Unsoeld-Lucy method -----
-NO TEMPERATURE CORRECTION
NO TEMPERATURE CORRECTIONS WHILE COR. .GT. -1.0 1.0
COLI: UNLUPAR GAMMAT=1. TAUMAX=0. TAUMAX2=0.
UNLUTEC LOC=0.1 INT=.2 OUT=.2 TB=.01 TBTAU=0.01
UNLUTEC TAUINT=10.
UNLUTEC SMOOTH TMIN=8000. CUTCORR=0.02 MONOTONIC
```

The temperature stratification must be adjusted in the course of the model iteration in order to establish energy conservation. However, the corresponding temperature corrections should only start to work after a new model has settled a bit and the maximum relative corrextions of any population numbers (CORMAX, see below) dropped below -1.0dex; they should be switched off again if it happens that CORMAX grows above 1.0 again. These limits are set by the line NO TEMPERATURE CORRECTIONS WHILE ...

The temperature corrections are calculated by a modified Unsöld-Lucy method, optionally combined with the *thermal balance* approach (see Sect. 14). This various parameters for these methods are specified on the UNLUTEC lines; for better readability, the UNLUTEC parameters can be arbitrarily distributed over more than one line.

```
---- version for handling lasering lines --------
LASERV = 2
```

Lasering lines are often a problem for the model calculation, and we have different versions to cope with it. Version 2 is currently recommended.

```
--------------- Ensuring hyperbolical type of the moment eqns. -----
-NO EDDIMIX
EDDIMIX START = 1.
EDDIMIX FIX   = 1.
EDDIMIX MAX   = 1.
```

The moment equations (second-order differential equations) require hyperbolical type. Although this type is principally given in the analytical equation, this might not be the case numerically. To always ensure the hyperbolical type, the "mixed" form for the Eddington factor g has been developed (see Sect. 12.6). The EDDIMIX lines specify the starting value and the minimum value of the "eddimix factor". The program automatically increases this facor if necessary, but this might cause some disturbances in the convergence. One might watch the corresponding output in wruniq.cpr, and chose higher values if appropriate.

```
----- Rate equations  -----------------------------------
POPMIN = 1.E-27
SPLITINVERSION
BROYDEN= 2
BROYDEN RESET 10.
```

Popnumbers below POPMIN are set to zero in the radiative transfer. In the rate equations, levels with popnums lower than POPMIN are eliminated under certain circumstances. SPLITINVERSION allows that the matrix inversion is performed separately per element block (recommended). BROYDEN=2 allows that the iterative solution of the non-linear rate equations at each depth point can be performed with the (faster) *Broyden method* instead of the Newton-Raphson method, unless some criterion defined by BROYDEN RESET=10.

```
--------------------  numerical parameters  ----------------------------
REDUCE=0.5
REDISMODE= COHERENCE
```

REDUCE reduces all corrections on popnumbers per iteration by the facto given as parameter (default: 0.5). Using this option might stabilize the iteration against growing oscillations. But it slows down the convergence, and can also make the temperature corrections going weird.

The REDISMODE line is necessary in the present version, but will become obsolete soon. Therefore we refrain from documenting it here.

```
---------Approximate lambda operator parameters ---------
OPC= DIAGTAU
AUTO GAMMA 640. 160. -1.1 0.5 0.1 0.1 1.
```

```
GAMMAD=1.
COLI: GAMMA=20.

NO EXTRAPOLATION
```

OPC choses among the different versions coded for the *Approximate Lambda Operator* (this option refers only to the continua). DIAGTAU seems to work best.

The GAMMA parameters regulate the iteration with approximate lambda operators. Ideally, the γ values should be as small as unity; the larger the values, the less agressive is the pre-estimated feedback of the popnumber corrections on the radiation field. The AUTO GAMMA line invokes a sophisticated mechanism which tightens the γ values as long as the iteration goes fine with it, and releases them otherwise (see Sect. 2.1 for details. GAMMAD refers to the iron superlines, for which the value 1.0 usually works fine. Note, however, that as default GAMMAD=.0 is set automatically, as soon as temperature corrections are active, because the ALO extrapolation is not compatible with a temperature correction factor which is applied to the iron line trasition rates (cf. Sect. **??**).

```
--------------------  plot options  ----------------------------------
PLOT_INBOX
PLOT_POPRANGE=1.E-30
PLOT POP GROUNDSTATES
PLOT FLUX FLAM 10PC
PLOT T(LOG(R-1))
PLOT HTOT
PLOT UNLU
PLOT ACC
PLOT ACC VELOCITY
PLOT RTAU1
-PLOT VELO
```

This block of lines requests standard plots which are assembled in the file steal.plot and finally, when the wruniq-job is finished, in wruniq*n*.plot.

```
--------- Convergence criteria --------------------------
SMALLPOP=1.E-8
EPSILON=0.005
FLUXEPS=0.01
JOBMAX=9000.
```

SMALLPOP defines a threshold for the popnumbers, below which their corrections are not taken into account for evaluationg the maximum relative corrections (default: 1.E-12). EPSILON defines when a model is considered as converged, namely when the maximum of all relative popnumber corrections is smaller then the given value (default: 0.05). If FLUXERR is given, convergence also demands that the radiatibe flux never deviates by more than the given fraction from flux conservation. JOBMAX defines the maximum job counter after which the wruniq iteration terminates without launching the formal integral. If one wants to continue the iteration, one may submit the wruniq job again with increased JOBMAX limit, or reset the JOB counter by sub njn*n*.

2.2. FGRID

The *coarse* grid of frequency (or wavelength) points can be defined in different ways:

- with the CARDS option OLD FGRID, if the model is starting with OLDSTART from an old MODEL file. In this case, the grid is taken from that old model;
- if OLD FGRID is NOT requested, the program checks if the CARDS file contains a line like
  BLUEMOST-WAVELENGTH = x.x
  (only BLUEMOST is significant for the keyword). In this case, x.x specifies the shortest wavelength point (in Å), which, however, might be superseded by a bluer edge or line encountered in the atomic data;
- if neither OLD FGRID nor BLUEMOST-WAVELENGTH = x.x is found in the CARDS file, the FGRID file is read and all wavelenghts listed there are inserted into the coarse grid, in addition to those points induced by the atomic data. If the FGRID file is not found in this case, an error stop is produced.

Thus, **the file FGRID is usually not necessary**. It can be used to define "manually" additional points for the *c*oarse frequency grid. For this purpose, the file must contain wavelengths (in Å) in monotonically increasing order. Lines beginning with * are comment. The first or only value is the bluemost freqency point. If it is intended to study the X-ray range, wavelengths as short as 2 Å might be needed. Otherwise, too short wavelengths might risk a program crash because of an underflow induced by the Boltzmann factor.

Example for an FGRID file:

```
*--- Bluemost Gridpoint set to 2A: This is for testing K-shell absorption!
**********************************************************************************
2.0
```

2.3. MODEL

The file MODEL communicates between the individual programs that work subsequently in the course of a model calculation (cf. Fig. 1.1). Only if the model iteration starts from a previous model, the file MODEL is input to the wrstart job. In any case, wrstart writes the file MODEL into the directory wrdata*n* when finished.

A wruniq-job fetches the file MODEL from wrdata*n* in the beginning, and replaces it when the job terminates regularly. Note that the MODEL-file is not updated if the wruniq-job crashes or is manually killed.

For the formal-job, the MODEL-file is only input.

The MODEL-file is organized as a *name-indexed random-access mass-storage file* with the help of self-written software. Since it is a binary file, it cannot be opened with an editor. However, for the advanced user or developer we have special tools to inspect the content of such files.

2.4. DATOM

The ASCII file DATOM describes the model atoms for all elements except of the *generic element* for the iron-group. An *abundance* must be assigned to each element which appears in the DATOM file, and vice versa.

For the beginner, it might be easiest to use an already existing DATOM file. If, however, you need to to include or exclude certain elements, or account for different ionization stages, you will have to customize your DATOM file accordingly. For this purpose there exists the job newdatom*n*, which is described in Sect. 6.1.

2.5. FORMAL_CARDS

This file is input to the formal-job and has two functions:

1. to specify which kind of emergent spectra shall be synthesized (e.g. for which wavelength ranges);

2. to supply the formal integral with additional atomic data, e.g. for the splitting of lines into multiplet components.

Especially because of the latter purpose, the file `FORMAL_CARDS` is huge and no longer convenient to handle manually.

For the beginner it might be sufficient to work with an existing file `FORMAL_CARDS` (which must be consistent with the `DATOM` file used).

If, however, you have changed the atomic data (`DATOM` file), or you want to change the wavelength ranges, you will have to create new `FORMAL_CARDS` by running the job `newformal_cards`*n*, which is described in Sect. 6.2.

2.6. FEDAT and FEDAT_FORMAL

These files describe the model atom of the *generic element*, i.e. of the iron group, in the superlevel approach. These files are also *name-indexed random-access mass-storage files*. They are created with the help of `Blanketing` program package which is documented elsewhere. This program package makes use of the huge database provided by Kurucz with many millions of lines, plus the *opacity data* for the Fe ions higher than Fe XIII.

Various FEDAT files have been prepared, and can be found in the directory `~wrh/Blanket/Models`. The relevant files are in the respective model directory and carry the extension ".mS".

Each model comes in two versions, distinguished by `BIG` and `SMALL` in their filename. The big version accounts for all lines available in the database, most of them being only theoretically predicted without confirmation in the lab, while the small version has only lines which are confirmed by measurements. Our original idea was to use the BIG version for the model calculation (`FEDAT`) to account for the maximum line-blanketing effect, but apply the small version in the formal integral (`FEDAT_FORMAL`) in order to avoid the appearance of features at possibly wrong wavelength in the emergent spectrum. Since this implies to underestimate the "iron forest", we now prefer to use the BIG version in both cases.

The FEDAT files are really big. Therefore, it is recommended to not copy the files into the `wrdata`*n* directories, but instead only provide symbolic links like, for instance,

```
ln -sf ~wrh/Blanket/Models/G2_BIG_VD100_FeXVI-K2015-parity/G2(...).mS FEDAT
```

The following points shall be taken into account when selecting the FEDAT links:

- Only use versions with the word `parity`. Without grouping the levels according to parity, spurious emissions might appear in the emergent spectra, as we had encountered before 2015 ("Ute emission").
- Make sure that the FEDAT file covers all ionization stages that you need to include. The roman number in the filename gives the highest ion that is in the file with full data, i.e. you can ask one higher ionization stage which will be included with its ground level only.
- Make sure that your FEDAT files has a resolution that is fine enough for your purpose. For FEDAT, the VDOP encoded in the filename shall correspond to the VDOP in your `CARDS` file. For `FEDAT_FORMAL`, the encoded `VDOP` should not be higher than the smallest microturbulence `VMIC` that you are adopting in the FORMAL_CARDS. Unnecessary small `VDOP` versions cost unnecessary computing ressources. Note that versions with different `VDOP` can be combined for `FEDAT` and `FEDAT_FORMAL` without problems, as long as the rest of the specifications (filenames) agree.

3. How to run a model

3.1. Preparing the input files

The command
`stat`
shows your available *chains* (cf. Sect. 1.2). Choose under which chain number *n* you want to run your model. If you need to have an additional chain, you can create one by executing the script
`~/work/dummychain/makechain.bash`

It is recommended to keep the number of chains low (<20), because each chain costs memory resources. Especially, chains should not be abused as a storage for models which are done.

Secondly, you consider if your intended calculation can start from a similar model which already exists. Note that it is much easier to start from a similar model than to run a new model from scratch.

If you know such a suitable start-model, you change into the directory where this model has been saved, e.g.:
`cd ~wrh/science.dir/wnegrid.dir/models.dir/11-11`
Note that this model does not need to be your own, but can belong to any user on the cluster as long as you have read permission.

Then you load this model into your chain *n* by typing:
`loadmod` *n*
By this, the files relevant as input are copied into your directory `work/wrdata`*n*. See Sect. 4.3 for more details.

Now you go into that directory with the shortcut:
`cddat` *n* and edit the file `CARDS` as the most important input file (cf. Sect. 2.1).

In case you choose there the `OLDSTART` option, the current file `MODEL` will serve as start approximation.

Note that this only works without further ado, if the DATOM file (i.e. the atomic data) is also kept the same. However, yoy may want to change the atomic data as well, for instance in order to include an additional chemical element. In this case, you must assemble new files `DATOM` and `FORMAL_CARDS` with the help of the tools `newdatom` and `newformal_cards` (see Sects. 6.1 and 6.2, respectively). Before doing so, you should first rename the previous `DATOM` file:
`mv DATOM DATOM_OLD`.

After having created then the new file `DATOM`, you type the command
`levelcards`
and paste the content of the file LEVELCARDS into your CARDS file. These lines are necessary to correctly identify the levels between the previous and the new model atom.

Finally, make sure that the files `FEDAT` and `FEDAT_FORMAL` are linked to generic-element data files which are appropriate for your purpose (cf. Sect. 2.6).

3.2. Submit the model calculation

Next you have to choose on which computer (“machine”) you want to start your model calculations. With the command
`psx all`
information about the capacities and current workload of all machines in our cluster is displayed. For a specific machine, you can ask
`psx` *machine*
where the default for the *machine* is the computer with your home directory.

The model is finally started with *submitting* the wrstart job by typing

`sub wrstart`*n* *machine*
where the specification of the *machine* is optional (default: the computer with your home directory).

You can check the status of your model computation with the `stat` command. The `ACTIVE` flag should first appear in the `wrstart` column, and then move to `wruniq`. After some time (hours to days), the `wruniq` iteration should report `converged` and the `ACTIVE` flag proceeds to the `formal` job. When finished, the `formal` job finally reports the status `done`.

The following commands may be typed for a running chain *n*:
`stat break wruniq`*n*
Effect: at next occasion, the MODEL file will be saved in its current status of iteration into the directory `wrdata`*n*, and the `CARDS` file will be loaded from there into the scratch directory where the model is calculating. Thus, any changes edited in the CARDS file become activated from now. The wruniq iteration will automatically continue.

`stat to-`*`machine`* `wruniq`*n*
same as `break`, but the wruniq job migrates to the computer *machine*

`stat stop wruniq`*n*
same as `break`, but the interation will not continue automatically. The stat command will display “Non-act”.

`stat KILL wruniq`*n*
kills the wruniq-job immediately. The no files will be saved. The `MODEL` file in `wrdata`*n* will stay as it was written at the last `break` or at the beginning of the `wruniq` job.

You might encounter that the `stat` command shows the wruniq entry “Non-act” (meaning: not active) for a specific chain. If you haven’t stopped the chain by hand (see above), the reason is probably that the maximum number of jobs (specified by `JOBMAX` in the `CARDS` file) has been exceeded before the model finally converged. After having checked this diagnosis and the health of the model with the `check` tools (see below), you may edit the corresponding CARDS file in `work/wrdata`*n* and specify a higher `JOBMAX` limit, and then submit the `wruniq`-job again by typing
`sub wruniq`*n* [to-machine]
for continuing the iteration. Alternatively, one can submit the “new job number” job
`sub njn`*n* [to-machine]
which resets the job counter in the `MODEL` file to 100 and restarts the iteration.

3.3. Supervising model calculations

It is usually a good idea to keep an eye on running model calculations. The corresponding commend-line tools will be described in more detail in Sect. 4. Here we mention already the most important commands:

The progress of the individual jobs can be observed using the respective commands

```
stat tcpr wrstartn
stat tcpr wruniqn
stat tcpr formaln
```

These commands watch the tail of the corresponding cpr-files in the `work/output` directory. To quit the command, hit ctrl-c.

In case of troubles it might be also usefull to look at the tail of the corresponding log-files with, e.g.,
`stat tcpr wrstart`*n*

Note that all `sub` and `stat`-commands are nicely supported by auto-completion (tab-key).

The process of convergence of the wruniq-iteration can be observed with the `check` tools:

`check` *n* [*username*]

displays, for your chain *n*, a logarithmic plot with the maximum relative correction of popnumbers, CORRMAX, over job number. This curve should go down until convergence is reached at EPSILON (unless other criteria, such as sufficiently accurate flux conservation (FLUXEPS) if requested in the `CARDS` file, are not satisfied). With the optional parameter `username` you can check the convergence of models from other users.

`checkt` *n* [*username*]
displays the current temperature stratification in comparison to the previous iteration (difference 100 times enhanced).

The check commands open an X11 window with WRplot, which is very minimalistic and therefore apt for remote use even via slow internet. In worst case, i.e. without X11 display, one can get the CORRMAX as ASCII list by typing
`checkh` *n*

Note that after a `check`-command, the present work directory (pwd) of this shell is set to the temporary directory in which the corresponding `wruniq`-job is running. Hence, you can inspect there, for instance, the current version of the plots by typing:
`wrplot steal`
or inspect the incrementally written output file (here named: `out`), or even look into the MODEL file with the appropriate tools (see Sect. 4).

3.4. Save models

If `stat` shows that in your model chain *n* the `wruniq`-iteration is `converged` and the `formal`-job is `done`, your model is finished and shall be saved:

- create (or change into) the directory where you want to store the models for your current project;
- create there a new subdirectory with an arbitrary name which helps you later to maintain the overview over your different models;
- change into this directory (cd)
- type:
  `modsave` *n*
  where *n* is the chain from which the model is to be saved.

Note that the modsave command warns you if the current directory is not empty before; in this case, you have probably forgotten one the necessary steps described above.

4. Command-line tools

4.1. Job submission

`sub` *jobname* [`to-`*machine*]

The parameter in brackets is optional; default is the machine where your home directory resides. The `sub` command sends the shell-script *jobname* via ssh to the specified computer (*machine*) and starts there its execution. It first checks the availability of this machine. The submit procedure maintains its own logfile `output/submit.log` and also issues messages in the `.log` and `.cpr` file of the submitted job *jobname*.

The `sub` command works independent from the present-work-directory; the job-script to be submitted must exist in your directory `wrjobs`, and must be designed for being directly submitted. The following job-scripts might be submitted:

jobname	action
`wrstart`*n*	start-job of a new model in chain *n*
`wruniq`*n*	continues model iteration if it stopped
`set_repeat`*n*	necessary before continuing iteration if converged before - see (a)
`njn`*n*	sets job counter to 100 and re-starts `wruniq`*n*
`modify`*n*	starts a `modify`-job for manipulation of the model (see Sect. 5.6)
`formal`*n*	manual submission of the formal-job

(a) Each chain stores in `wrdata`*n*`/nextjob` the information, which program has to work next on the `MODEL` file. Once the model in chain *n* converged, this entry is set to `MODEL`. In order to resume the iteration, the entry must be set to `WRCONT`, which is done by the `set_repeat`*n* job, before `wruniq`*n* can be submitted again.

4.2. Job status

`stat` [*parameters*]

The `stat`-commands can be issued from any shell, independent of the present-work-directory. The `stat` tool has many options for checking or changing the status of model chains; possible parameters are:

parameters	action
`help`	displays help
(none)	status of all chains
`-u` *username*	status for a different user; this argument must be the last one
`tcpr` *jobname*	shows incrementally the tail of `output/`*jobname*`.cpr` (a)
`tlog` *jobname*	shows incrementally the tail of `output/`*jobname*`.log` (a)
`break wruniq`*n*	save MODEL to and reload `CARDS` from `wrdata`*n*, then continue `wruniq`*n*
`to-`*machine* wruniq*n*	same as break, but `wruniq`*n* migrates to computer *machine*
`stop wruniq`*n*	wruniq*n* stops at next occasion; `MODEL` is saved
`KILL` *jobname*	kills job *jobname* immediately; nothing is saved
`clear` *jobname*	sets the status entry of *jobname* to `clear` (b)
`name wruniq`*n Name*	sets the name of the chain to *Name*
`+name wruniq`*n string*	inserts *string* at the beginning of the chain's name
`name+ wruniq`*n string*	appends *string* to the chain's name

(a) hit ctrl-c to quit

(b) Submission of a job is aborted if the corresponding chain is still `ACTIVE`; this prevents that you can destroy a running model accidentally. If, however, a program or computer crash has left an `ACTIVE` entry and you made sure that the job is really not executing anymore, the `stat clear` command will make the corresponding chain usable again.

4.3. Load an existing model

loadmod *n*
copies from from working directory to wrdata*n* all files that are needed to (re-)calculate this model, which include:

```
CARDS
FORMAL_CARDS
MODEL
DATOM
FEDAT
FEDAT_FORMAL
```

For the `FEDAT` files, only the links are copied. Write permission is given to all these files. The *Name* entry of chain *n* (see Sect. 3.3) is set to blank.

4.4. Save a model

modsave*n*
saves model data into the current working directory, which should be empty for that purpose. The following files are copied:

```
CARDS
DATOM
FEDAT
FEDAT_FORMAL
FORMAL_CARDS
formal.out
formal.plot
MODEL
modinfo.kasdefs
wrstart.out
wruniq.out
wruniq.plot
```

For the `FEDAT`-files, only symbolic links will be stored. The file `modinfo.kasdefs` defines a long list of variables in WRplot syntax which might be usefull to include in a WRplot-file which displays the model.

The saved files will all become write-protected.

4.5. Create an additional chain

makechain *n*

generates chain *n* with all necessary directories and some dummy files

4.6. Clone a chain

`clone` *n m* [*username*]
copies model input files from wrdata*n* (optionally: of user *username*) to the own wrdata*m*

4.7. Reading from the MODEL file

The file `MODEL` is a binary file (name-indexed random-access mass storage file, using our own structure and syntax definitions) which can only be read with special tools. Other files of this type are the iron-data files (`FEDAT, FEDAT_FORMAL`) and the temporary file with Eddington factors. Note that in the directory where the model calculations are performed, e.g. `/home/corona/tmp_data/wrh/wruniq1`, these mass-storage files are renamed to `fort.3`, `fort.17`, and `fort.21`, respectively, for the reason that fortran cannot open mass-storage files by their name.

`msinfo` *file* [*parameters*]

Possible parameters are:

`msinfo MODEL INFO-L`
displays a list of all variable names that are stored in this name-indexed mass-storage file (here: file MODEL)

`msinfo MODEL INFO-D` *name-of-variable formatstring*
displays the content of the requested variable; the formatstring must be given in FORTRAN syntax (see examples). Quotes are needed. Examples:

`msinfo MODEL INFO-D 'ND' '(I6)'`
displays the variable texttttND (number of depth points) as 6-digit integer;
`msinfo MODEL INFO-D 'TEFF' '(F8.1)'`
displays the effective temperature of the model as floating-point number with 8 digits (one decimal).

A short version of this tool is
`msread` *name-of-variable* [*file*]

which uses default formats. *file* is the name of the mass-storage file to be read; if this parameter is not given, the default is `MODEL` or `fort.3` in the present work directory.

5. Troubleshooting

5.1. Job submission has no effect

The problem: after having submitted a `wrstart` or `wrstart` job, you cannot observe that this job is really executing.

Diagnosis: inspect
`stat tcpr wrstart`*n*
or
`stat tcpr wruniq`*n*

One possible reason is that the corresponding chain is still flagged as `ACTIVE`, although the corresponding job has crashed.

Solution: use the `stat clear` command (see Sect. 3.3).

5.2. DIMENSION INSUFFICIENT

The PoWR executables are provided in three versions which differ regarding the size of the arrays and thus the computer memory they require (remember that FORTRAN programs might still use fixed memory allocation). The versions are:
std (standard)
vd20 (medium, often sufficient for VDOP=20 km/s)
xxl (largest available dimensions).

If you encounter an error message like `DIMENSION INSUFFICIENT`, you must change the corresponding jobfile to a higher-dimensioned version (by commenting/uncommenting corresponding lines). If even the xxl version is insufficient, you might reconsider your requirements. In worst case, a program version with higher dimensioning has to be compiled.

5.3. Solution of the Rate Equations fails at some depth points

`stat tcpr wruniq`*n*
displays from each run of `steal` (statistical equations solver) a block like

```
L=  1   Niter=   3     NBT
L=  2   Niter=   7     NBTTTTB
L=  3   Niter=   7     NBTTTTB
L=  4   Niter=   7     NBTTTTB
L=  5   Niter=   7     NBTTTTB
L=  6   Niter=   7     NBTTTTB
L=  7   Niter=   7     NBTTTTB
L=  8   Niter=   7     NBTTTTB
...
```

Here, L is the depth index, while each letter in the row `NBTTTTB` stands for one iteration step (N = Newton-Raphson, B = Broyden, T = two-point Broyden). Occasionally, the maximum number of iterations ITMAX (default: 50) might be reached without convergence.

If this happens only occasionally, this is not a problem. PoWR will replace the population numbers for this depth point by interpolated values by automatically generating a corresponding `modify`-job (cf. 5.6).

If, however, many such non-converged depth points are encountered, the `wruniq`-job might terminate with a corresponding error message.

In this case, something went fatally wrong with the model. If the problem is encountered immediately after the model was started, the start approximation might be too bad.

In case the calculation started from a previous model, something might have gone wrong with the assignement between the level indices in the old and new MODEL file. Compare the DATOM files and, if they differ, make sure that the LEVELCARDS have been properly created and included in the CARDS file (cf. Sect. 3).

In case the model calculation was started from scratch (i.e. without `OLDSTART` option), one may try different start approximations (`LTESTART`, `JSTART`) and, in the latter case, with different values for the `BLACKEDGE` parameter. If nothing helps, one may re-think to find a previous model that is suitable for an `OLDSTART`.

In case that such massive non-convergence of rate equations occurs later in the course of a model computation, the model went obviously completely wrong and is probably lost. Maybe you have saved the model in some intermediate stage by a `stat break` command; then you may quickly `stat KILL wruniq`*n* the chain and have a chance for continuing from the saved stage with different settings of the numerical parameters (see Sect. 5.4).

5.4. Global convergence problems

The problem: the plot displayed by
`check` *n*
reveals that the corrections do not decrease.

One should understand thet a model calculation consists of several, nested or coupled iteration algorithms. The outer iteration refers to the population numbers; this iteration is accelerated by means of *Approximate Lambda Operators*. Obviously, too strong acceleration bears the risc of over-corrections which might lead to a runaway or an alternating divergence of the iteration.

These *Approximate Lambda Operators* can be tuned by a parameters called `GAMMA`: the lower `GAMMA`, the more the iteration is accelerated. Largest acceleration is for `GAMMA=1.`, lowest for ver large values. In the syntax of the CARDS file `GAMMA=.0` means to switch off the acceleration entirely.

Alternating corrections might also be damped with the `REDUCE` command (cf. Sect. 7).

Often, convergence problems are caused by the temperature correction. There are various parameters to tune the Unsöld-Lucy method, see Sect. 7.

Temperature corrections might go weird if they are already applied at the beginning of a model iteration, when all other equations are also still not fulfilled sufficiently. Therefore, the line in the `CARDS` file:

`NO TEMPERATURE CORRECTION` [parameters]

should be tuned such that temperature corrections remain switched off until the corrections settled below some threshold (see Sect. 7 for the parameters).

5.5. Flux conservation problems

Sometimes, models do not reach flux conservation to the accuracy demanded with the FLUXEPS option. This can have various reasons:

- EDDIMIX issues (see ...)

Another issue has to do with the superlevel approximation. In Eq. **??**

In the original version,

```
IRONLINES-EXPFAC TRADITIONAL
IRONLINES-EXPFAC OFF
IRONLINES-EXPFAC TEFF
```

See also the PoWR-Memo 20180219.txt

5.6. Manual intervention: job `modify`

From inspection of the corresponding plots, you might feel that the temperature stratification of the stratification of population numbers show some defect (e.q. a discontinuity) which the iteration is unable to fix. In this case, one might intervene manually by interpolating (or extrapolating) temperature and/or popnumbers between selected depth points:

First, if the iteration is still running, stop it by typing `stat stop wruniq`*n* and observe that the chain really stopped.

In the meantime, you may edit the file `work/wrjobs/modify`*n*. Somewhere in the middle of this file, you will find a block like

```
cat > MODIFY_INPUT << %
-NO TEMPERATURE CORRECTIONS
-NO POPNUMBER CORRECTIONS
INTERPOLATE FROM POINT 40 TO POINT 44
-INNER EXTRAPOLATION  FROM POINT  69.  SECOND POINT  69.
-OUTER EXTRAPOLATION  FROM POINT 10  SECOND POINT 11
%
```

The meaning of these lines is obvious:

`NO TEMPERATURE CORRECTIONS`
means that the manipulation does not involve temperatures;

`NO POPNUMBER CORRECTIONS`
means that the manipulation only involves the temperatures;

`INTERPOLATE FROM POINT 40 TO POINT 44`
means that for the depth points 41, 42, and 43 the current values are replaced by those obtained by an interpolation between the values of the given points (depth indices 40 and 44 in this example. The interpolation is linear in logarith for popnumbers, and just linear in temperatures.

If the points to be manipulated are attached to the inner or outer boundary, the adequate commands are, for example:

```
INNER EXTRAPOLATION  FROM POINT  69.  SECOND POINT  69.
OUTER EXTRAPOLATION  FROM POINT 10  SECOND POINT 11
```

If the two depth indices differ, a linear extrapolation will be tight to the two given points; if the indices are identical, the values will be constantly repeated from this index to the boundary.

One can give a whole list of interpolation commands which then will be executed sequentially by the modify job.

When the modify job reports success (which can be seen also with `stat tcpr wruniq`*n*), you may now type `sub wruniq`*n* for continuing the iteration.

5.7. Restart converged model

If the model is already converged, but you want it to iterate further for better accuracy. In this case, tighten the convergence criterion, e.g.

- decrease `EPSILON` in the CARDS file
- reset the job status by typing `sub` set_repeat*n*
- restart by typing `sub njn`*n* or `sub wruniq`*n*

5.7.1. Do your model atoms cover the ionization stages as needed?

Please check the plots of the population numbers for the ion's groundstates (requested by the `CARDS` line: `PLOT POP GROUNDSTATES`). Often the lowest as well as the highest ion of each element is represented by only one or two *control levels*. This is especially true for iron. If such *control level* becomes leading in significant parts of the atmosphere, the coreponding ion should obviously described with more levels in the atomic data (see Sect. 6.1).

6. Providing atomic data

The PoWR code is supplied by atomic data via three files:

- **DATOM**
  is an ASCII file read by all individual programs and contains the energy levels and transition cross sections of the model atoms, except for the *generic element* (iron group). For the latter, only the range of ionization stages is specified in `datom`;
- **FEDAT**
  is also read by all individual programs and contains the data for the *g*eneric element (iron group) which is treated in the superlevel approximation. The file is binary (mass-storage). `FEDAT` files are created by the program package `Blanket`;
- **FORMAL_CARDS**
  is an ASCII file only read by program `formal`. It specifies the wavelength ranges and line transitions for which the emergent spectrum is calculated;
- **FEDAT_FORMAL**
  is a file analogues to `FEDAT`, but for the program `formal`. It may be identical to `FEDAT`, or may have been created with the program package `Blanket` under restriction to those lines with confirmed wavelengths. In any case, both `FEDAT` and `FEDAT_FORMAL` must have been created with *the same definition of the superlevels* in the `Blanketing` program, otherwise nonsense will be calculated, although the program may not necessarily crash.

6.1. Assembling a new DATOM file

`DATOM` files from existing models may be re-used. However, we are permanently updating our atomic data base; therefore it is highly recommended to construct a new `DATOM` file when starting a new project. A new DATOM file in `wrdata`*n* (*n* = chain number) is created by submitting a job
`sub newdatom`*n*

This program needs an input file `NEWDATOM_INPUT` in the same `wrdata`*n* directory. The syntax of this file becomes obvious from the following example:

```
-------------------------------------
PATH = /home/corona/wrh/work/wrdata-archive
DRTRANSIT
NO-K-SHELL

ELEMENT HE
ION I NLEVEL=1
ION II
ION III

ELEMENT C
ION II  NO-DRTRANSIT
ION III
ION IV NLEVEL=3

ELEMENT G  3 10
-------------------------------------
```

Any line which does not start with a valid keyword is considered as a comment. Note that each DATOM file generated with newdatom gets a comment header which shows the NEWDATOM_INPUT lines that have been used.

The file starts with specifying the PATH to the data source. The current and maintained data base is presently located in the directory /home/corona/wrh/work/wrdata-archive. The PATH can be re-defined subsequently, e.g. in order to include a test version for one specific ion.

DRTRANSIT is a global switch to include the dielectronic transitions. The opposite command is NO-DRTRANSIT. By default the dielectronic transitions are switched off (NO-DRTRANSIT).

The DRTRANSIT switch can also be turned per element: in this case, the DRTRANSIT or NO-DRTRANSIT command line must follow after the ELEMENT line and holds only for that element. – Note that there is also an individual switch for each ion (see below).

NO-KSHELL is a global switch to suppress the K-shell opacities in the X-ray regime. By default these opacities are included. Contrary to NO-KSHELL acts the switch KSHELL (default). As described above for the DRTRANSIT switch, the KSHELL switch can also be turned back and forth per element when the corresponding command line follows after the ELEMENT line.

Each element has a block starting with the line
ELEMENT <symbol>
where <symbol> denotes the chemical symbol of the element, like HE or C. The chemical symbol must correspond to files in the data base.

Presently, chemical symbol with two characters (like HE, SI for helium or silicon) *must* be written with capital letters. We plan to change this, but since each *level name* in the atomic data must start with the chemical symbol this needs some efforts to stay compatible.

For each element, the ions that are to be included must be specified by a line like
ION IV
or, for example with optional parameters,
ION IV NLEVEL=17 NO-DRTRANSIT

One may chose different sets of ions according to the parameter range of the stellar models to be calculated, e.g. omitting low ions like O II for hot models. Note that we use *roman numbers* for specifying the ionization stages. The file names in the data base are constructed from the chemical symbol and this roman number.

Ions can be restricted in their number of levels by the optional parameter NLEVEL= *n*.

One possible application is to omit the high-excitation part of a model ion which has been introduced only for simulating the infrared spectrum, when the present project has no intention in this wavelength range.

More often, the NLEVEL options will be used to omit ionization stages that are not needed. E.g., for cool models one may want to omit high ions. Usually we include more ions (one of lower and one of higher stage) than actually needed in the form of (one or few) *control levels* for which one can check in the final model that the population is indeed negligible. For this purpose one may chose NLEVEL=1. However, it may happen that by restricting some ion (e.g. C IV) to only one level, some photoionization transitions (e.g. from C III) may loose their target level. In this case, newdatom automatically increases NLEVEL till all target levels are included.

Optionally, an ION line can set the DRTRANSIT switch differently from the global setting with the optional parameter DRTRANSIT or NO-DRTRANSIT.

The line
ELEMENT G *n m*
for the *generic element* has a special syntax. This line has two further mandatory arguments, which are integer numbers specifying the lowest (*n*) and highest (*m*) ionization stages which will be taken into account. The lowest and highest ion are represented by their ground level only ("control level"). For including neutral iron

in detail, the lowest GENERIC stage must be specified as “0”.

6.2. Assembling a new FORMAL_CARDS file

The FORMAL_CARDS file from an existing model may only be re-used if the DATOM file remained also unchanged. If, however, a new DATOM file has been assembled, the FORMAL_CARDS file must also be reconstructed for consistency.

We are permanently updating our atomic data base; therefore it is highly recommended to construct new DATOM and FORMAL_CARDS files when starting a new project. A new FORMAL_CARDS file in wrdata*n* (*n* = chain number) is created by submitting a job
sub newformal_cards*n*

This program needs an input file NEWFORMAL_CARDS_INPUT in the same wrdata*n* directory.

The syntax of this file becomes obvious from the following example:

```
-------------------------------------
STANDARDPATH /home/corona/wrh/work/wrdata-archive/
CALIB
CONT
************************************************
STRING COMMENT  * UV
STRING COMMENT  * IUE
STRING COMMENT  * FUSE
STRING COMMENT  * IUE SHORT
STRING COMMENT  * IUE LONG
RANGE       950.     3000. UV
************************************************

************************************************
STRING COMMENT  * OPT
STRING COMMENT  * VIS
RANGE      3160.     9000. OPT
************************************************
```

The effects of the NEWFORMAL_CARDS_INPUT commands are:

1. Some specific input lines are interpreted by the newformal_cards program itself and guide its actions; these options are listed below.
2. The main action of the newformal_cards program is initiated by the command(s) RANGE (see below): for all lines and multiplets in the specified wavelength range, the FORMAL_CARDS data are assembled from our atomic data base.
3. The newformal_cards program also reads the DATOM file; only those transitions can be included in the FORMAL_CARDS, for which the corresponding levels do exist in the model’s atomic data (i.e. in file DATOM).
4. All other lines encountered in NEWFORMAL_CARDS_INPUT, including comment lines, are directly passed into the NEWFORMAL_CARDS file. Consult Sect. 9 for the list of possible commands.

NEWFORMAL_CARDS_INPUT commands which guide the actions of the program newformal_cards:

STANDARDPATH = *pathname*

Path to the atomic data archive (mandatory!)

TESTPATH = ***pathname***

Path to nother (e.g. own) atomic data archive, with FORMAL_CARDS which you want to test. For those ions for which FORMAL_CARDS are found in the TESTPATH, these test data are taken instead of the corresponding data in the STANDARDPATH.

NOTESTPATH

after this command, the TESTPATH is no longer used.

RANGE $\lambda_1 \lambda_2$

causes the atomic data for the wavelength range from λ_1 to λ_2 (in Å) to be assembled, with all additional options which are active in this moment.

DRTRANSIT

includes the DRTRANSIT cards in FORMAL_CARDS. cf. →NO-DRTRANSIT

NO-DRTRANSIT

switches off the inclusion of DRTRANSIT cars in the FORMAL_CARDS. The default. cf. →DRTRANSIT

RESTRICT *Element* | NONE [ION]

Only the FORMAL_CARDS for the given element or (more restrictive) the given ion are included. One can set serveral RESTRICT cards. Option NONE resets all restrictions.

Examples:

```
RESTRICT NITROGEN
RESTRICT HE II
RANGE 950. 3000.    IUE #ALIAS "UV" "FUSE" "IUE SHORT"

RESTRICT NONE
```

6.3. Harvesting atomic data from the Opacity Project (OP)

This tool written by G. Gräfener uses data from the Opacity Project (theoretical values) and from the NIST Atomic data base (observed levels and lines) to generate atomic input data, e.g. the files DATOM and FORMAL_CARDS.

6.3.1. Source code

The sources of this program and executables for linux and OSF can be found at

```
/home/musca/htodt/lib_opdat
```

the directory includes a makefile to generate the executable.

6.3.2. Input files

6.3.2.1. OP_LEV Please check the following options

- General energy order
- Electron configuration
- Energy (Ryd) wrt ionization potential
- Statistical weight
- Effective quantum number

and uncheck all other Output options.

6.3.2.2. OP_LIN

Please check the following options

- Level order
- gf-value
- Wavelength (A)
- Statistical weight (i)
- Statistical weight (j)

and uncheck all other Output options.

6.3.2.3. OP_CONT

(optional) Please check the following option

- General energy order

6.3.2.4. LEVELS

This file contains a description of the levels of the model ion to be generated.
Comment lines start with a *.
The kind of the ion is described by the keyword `ION` followed by the atomic number, the number of electrons and the ionization energy in $\text{cm}^{-}1$ (unformatted), e.g. for O II:

```
ION      8 7           283270.9
```

Every level to appear in the DATOM file is generated by a line containing the Keyword LEVEL followd by a level designation, e.g.

```
LEVEL   "O 2p3*4S01"
```

Note the format for the level designation: the first two characters (both capitals) must be the symbol of the element, followed by some arbitrary description like the ionization stage, the quantum number, the angular momentum, the mulitplicity and the parity.

Every `LEVEL` keyword must have at least one following line with the keyword `SUBLEVEL` followed by the number of the OP level of the file `OP_LEV` and the energies (from NIST) for every J or an asterisk * if the OP level eneryg should be used, e.g.

```
SUBLEVEL 023           232480.44   232527.09
SUBLEVEL 120 *
```

The keyword `END` stops the further readin of this file.

6.3.2.5. INPUT

contains the output options. Keywords are

```
LEVELS
LINES
CONTINUA HYD
CONTINUA POLYFIT
CHECK
```

6.4. Remarks

6.4.1. Line strengths

The oscillator strength f_{lu} is related to the Einstein coefficient A_{ul} (with λ in Å and A_{ul} in s^{-1}) via (Wiese et al. 1996):

$$f_{lu} = 1.499 \times 10^{-16} \lambda^2 \frac{g_u}{g_l} A_{ul} \quad (1)$$

$$A_{ul} = 6.670 \times 10^{15} \lambda^{-2} \frac{g_l}{g_u} f_{lu} \quad (2)$$

or

$$A_{ul} = \frac{8\nu^2 \pi^2 e^2}{m_e c^3} \frac{g_l}{g_u} f_{lu} \quad (3)$$

$$(4)$$

For the multiplet transitions from a lower state l to an upper state u the following equations hold:

$$f_{\text{multiplet} \quad l-u} = \sum_{J_u} f_{lu} \quad \text{(only upper state } u \text{ is splitted)} \quad (5)$$

$$A_{\text{multiplet} \quad u-l} = \sum_{J_l} A_{ul} \quad \text{(only lower state } l \text{ is splitted)} \quad (6)$$

For the case of only small fine structure splitting, e.g. small ΔE, the levels with different Js are combined to one for the line radiative transfer and the statistical equations. This combined level has a statistical weight and energy:

$$g = \sum_J g_J \qquad E = \overline{E} = \frac{1}{g} \sum_J E_J \, g_J \,, \quad (7)$$

where the statistical weights are

$$g_J = 2J + 1 \qquad g_{\mathrm{M}} = (2L + 1)(2S + 1) \quad (8)$$

The latter ones applies to an intial or final state of a multiplet (M).

So the fine structure components of the line multiplet for the upper and lower level are combined to one single transition with wavelength

$$\overline{\lambda}_{ul}[\text{Å}] = \frac{10^8}{\overline{E}_u[\mathrm{cm}^{-1}] - \overline{E}_l[\mathrm{cm}^{-1}]}. \quad (9)$$

Furthermore, following Wiese et al. (1966) the absorption oscillator strength is a weighted mean by

$$\overline{f}_{lu} = \frac{\sum_{J_l,J_u} (2J_l + 1) \lambda(J_l, J_u) f(J_l, J_u)}{\overline{\lambda}_{ul} \sum_{J_l} (2J_l + 1)} = \frac{\sum_{l,u} g_l \, \lambda(l, u) f(l, u)}{\overline{\lambda}_{ul} \sum_l g_l} \quad (10)$$

$$\approx \frac{\sum_{l,u} g_l \, f(l, u)}{\sum_l g_l} \quad \text{for only small split in energy} \quad (11)$$

where $(2J + 1)$ is the statistical weight of this fine structure level or analogously the transition probabilities

$$\overline{A}_{ul} = \frac{\sum_{J_l,J_u}(2J_u+1)\lambda^3(J_l,J_u)A(J_u,J_l)}{(\overline{\lambda}_{ul})^3\sum_{J_u}(2J_u+1)} = \frac{\sum_{l,u}\, g_u\,\lambda^3(l,u)A(u,l)}{(\overline{\lambda}_{ul})^3\sum_u g_u} \tag{12}$$

$$\approx \frac{\sum_{l,u}\, g_u\, A(u,l)}{\sum_u g_u} \quad \text{for only small split in energy} \tag{13}$$

These are the generalizations of Eqn. (5), (6).

For the usual case of small fine structure splitting, the differences in λ within a multiplet are very small and the wavelength factors can be neglected.

On the NIST web interface the average f and A for a multiplet can be shown via the following output options:

```
Output ordering: Multiplet

Transition strength: f_ik
```

7. All CARDS options in alphabetical order

2BETALAW BETA2=*x.x* FRACTION=*x.x*

e.g. `2BETALAW BETA2=8.  FRACTION=0.4`
uses a different β exponent for the velocity for the outer part. `FRACTION` specifies the velocity fraction of `VFINAL` which is calculated from `BETA2` (WR-Memo 060512). Velocity law is given by

$$v(r) = v_\infty \left[q_1 \left(1 - \frac{r_0}{r}\right)^{\beta_1} + q_2 \left(1 - \frac{r_1}{r}\right)^{\beta_2} \right]$$

where $q_1 + q_2 = 1$.
Note that this card is set in addition to `VELPAR`.

ABORT_AUTO_MODIFY obsolet, see p. 55 `STOP_AUTO_MODIFY`

AUTO GAMMA 640. 40. -1.1 0.5 [0.1 0.1 1.]

Automatic adjustment of GAMMAs, with following arguments:

AG(1)	maximum GAMMA, start GAMMA
AG(2)	minimum GAMMA, final GAMMA
AG(3)	maximum CORRMAX allowed for decreasing GAMMA
AG(4)	minimum CORRMAX threshold for increasing GAMMA
AG(5)	GAMMAL factor
AG(6)	GAMMAR factor
AG(7)	GAMMAD factor

last three parameters are optional and are used to calculate `GAMMAL = GAMMAC / AG(5)` and so on, default is 1. For explanation of GAMMAs see keyword GAMMA. GAMMA is increased by factor 2, if one ore more depth point did not converge in last iteration. GAMMA is decreased, if last 6 iterations converged.
Note: It is strictly recommended not to set the first parameter (maximum GAMMA) to zero. Otherwise GAMMAs can cause floating point overflow. Note: Explicitly set GAMMAs overwrite AUTOGAMMAs.

AUTO_MODIFY `[ TEMP | NOTEMP ]`

without any further options it is the default and be therefore omitted (see p. 46 `NO_AUTO_MOD`). So, by default population numbers and temperature (if T-corrections are active) for a non-converged depth point are interpolated from the neighboring depth points.

Option `TEMP` enforces T-corrections while `NOTEMP` prohibits T-correction regardless of T-corrections are active or not.

BLUEMOST-WAVELENGTH *x.x*

is one of three possible methods to specify shortest wavelength for the *coarse* grid of wavelenths / frequencies:

- In connection with an `OLDSTART`, the `CARDS` option `OLD FGRID` takes that grid also from the old `MODEL`;
- In case `OLD FGRID` is *not* requested, the program checks if the `CARDS` file contains a line like `BLUEMOST-WAVELENGTH =` *x.x*
  (only `BLUEMOST` is significant for the keyword). In this case, x.x (in Angstroem) specifies the shortest wavelength point, which, however, might be superseded by a bluer edge or line.

- If neither `OLD FGRID` nor `BLUEMOST-WAVELENGTH` is found in the `CARDS` file, the code looks for a file `FGRID` in the `wrdata`*n* directory and takes all wavelenghts listed there to the coarse grid. If `FGRID` does not exist in this directory, all methods to specify the coarse wavelength grid failed and an `ERROR STOP` is executed.

BROYDEN= ***n***

use Broyden method for solving the rate equation (steal) not before the *n*-th iteration. So for the first *n - 1* iterations the Newton-Raphson method is used. (This is indicated in the cpr-output file by the letter "B" for Broyden and "N" for Newton, e.g. `L= 13 Niter= 3 NNB`, meaning that for the first and second iteration Newton is used and from the third iteration on - here also the last one - the Broyden method.)

E.g. `BROYDEN= 1`: begin immediately with Broyden method, no inverse Jacobian is calculated (faster). For improving the convergence, the inverse Jacobian from the last steal can be used as start approximation, saved in the DMFILE (fort.16 in tmp directory) in the ass$kn directory.

If convergence gets worse, the steal job switches back to Newton-Raphson method.

As no DMFILE is available for the very first steal, this steal must begin with a Newton-Raphson iteration, and only the subsequent steals can use Broyden form the first iteration on. [1]

BROYDEN RESET 1.

parameter for cancelation of Broyden iteration. Broyden iteration is canceld, if the norm $||\vec{n}||$ > `BRRESET` × $||\vec{n}_{\rm old}||$.
(subroutine: linpop)

COLI: GAMMA

s. GAMMA; this GAMMA is used for iron and continua in the coli job on the fine frequency grid. Recommended are values like `GAMMA=20.` or 10.

COLI: INTEGRATION ELABORATED obsolete option (is now default)

COLI: OPERATOR NOFREQUENCY obsolete option, never used

COLI: UNLUPAR GAMMAT=*x.x* TAUMAX=*x.x* TAUMAX2=*x.x*

Option for ALI-like amplification of the temperature correction of the "first term" in the Unsoeld-Lucy procedure, de-activated if `GAMMAT=0.` Damping of the Temperature-ALO with high Rosseland optical depth and test output to file `taccelerate` and `wruniq$n.cpr`
Defaults: `GAMMAT = 0. UNLU_TAUMAX = 1000. UNLU_TAUMAX2 = 100.`
Typical values:
`COLI: UNLUPAR GAMMAT=1. TAUMAX=0. TAUMAX2=0.`
Routines: read by `deccoli`, used by `frequint` and `frequnorm`.

DENSCON = ***d.d*** **[TAU|VELO|RADIUS x.x1|SONIC x.x2]**
| [HILLIER x.x | EXPRADIUS x.x | EXPTAU x.x | NAJARRO x.x1 d.d2 x.x2]
| [LIST TAU|VELO|RADIUS x.x1 d.d1 x.x2 d.d2 . . .]

defines the density contrast for clumped models between the clumps as seen by the rate equations and the average matter as seen by the radiative transfer. E.g. for [WC] stars `DENSCON =10.`, for WN `DENSCON= 4`, i.e. the density of the clumps is 10 times higher than the average density. Between the clumps is

[1] The DMFILE is copied only at the beginnig and and the end of wruniq to the tmp_data directory and back. Meanwhile it is - like the EDDI file usually in /home/$TMPHOST/tmp_2day/$USER/ass$kn, where $TMPHOST is an entry in work/wrjobs/tmphosts, therefore the ass directory exists only once per chain.

nothing than void, so D is the inverse of the volume filling factor f :

$$D = \frac{\varrho_{\text{clump}}}{\langle\varrho\rangle} = f_{\text{V}}^{-1} \tag{14}$$

For unclumped, smooth models `DENSCON= 1.`

Without any further argument the clumping starts at R_* (innermost depth point).

More "realistic" and needed for hydrodynamic consistent models following Gräfener & Hamann (2005) is a depth dependend clumping. If the second argument (`CLUM_CRIT`) is `TAU`, then the next argument (*x.x1*) defines the optical depth τ where to start followed by the value (*x.x2*) of τ where clumping has reached `DENSCON` value, e.g. `DENSCON = 10. TAU 0.7 0.35`. For the routine `clump_struct.f` the order of τ_1, τ_2 doesn't matter (automatically sorted). So, the smooth transition for every depth point `L` between τ_1, τ_2 is described by

$$D_l = \frac{1}{2}\left[1 - \cos\left(\frac{\pi}{\Delta x}\left[\tau_l - \tau_1\right]\right)\right] + \frac{1}{2}\left[1 + \cos\left(\frac{\pi}{\Delta x}\left[\tau_l - \tau_1\right]\right)\right] D_\infty \tag{15}$$

where D_l is the density contrast at depth point l, $\Delta x = \tau_2 - \tau_1$ and D_∞ is the given (maximum) `DENSCON` value.

If the second argument is `VELO`, arguments $x.x1$ and $x.x2$ are understood as velocities. If the second argument is `RADIUS`, arguments $x.x1$ and $x.x2$ are understood as radii.

The keyword `SONIC` refers to radius, velocity, or optical depth of the sonic point (instead of giving `x.x`). Note that the sonic point here refers only to the sound speed and does not account for any microturbulence which might be specified via the `MIC` card.

For numerical stability (esp. in the `TAUMAX` iteration) one should avoid to have something like a shallow step for D over τ. At least 3 grid points for a not too steep increase of D are recommended.

The `HILLIER` version (Martins et al. 2009) of depth dependent clumping is called via, e.g.

```
DENSCON = 10. HILLIER 100
```

with the last argument being the reference velocity υ_{cl} in the formula for the depth dependent filling factor:

$$f = f_\infty + (1 - f_\infty)\exp(-\upsilon(r)/\upsilon_{\text{cl}}) \tag{16}$$

Note that f_∞ is only reached for $\upsilon(r) \gg \upsilon_{\text{cl}}$

Alternatively, the last argument can also been as a relative number w.r.t. the sonic point, e.g., `0.75S` means 75% υ_{sonic}.

Similiar to the Hillier formula but over radius, is the `EXPRADIUS` formula:

$$f = f_\infty + (1 - f_\infty)\exp\left(-\frac{r - R_*}{r_{\text{cl}} - R_ast}\right) \tag{17}$$

or over the $\tau_{\text{Ross, cont}}$ scale, the `EXPTAU`:

$$f = f_\infty + (1 - f_\infty)\exp\left(-\frac{\tau(r)}{\tau_{\text{cl}}}\right) \tag{18}$$

Morover, the clumping formula by Najarro et al. (2009) via keyword `NAJARRO` also allows for a an increase of the clumping, but also for a decrease after the maximum (`d.d`) has been reached. With `d.d2` = 1 the wind will be unclumped again in the outer parts. Note, however that this maybe unphysical, as usually $\upsilon_{\text{exp}} \ll \upsilon_\infty$ for the clumps.

$$f = f_{\max} + (1 - f_{\max})\exp\left(-\frac{\upsilon(r)}{\upsilon_{\text{cl,1}}}\right) + (f_\infty - f_{\max})\exp\left(-\frac{\upsilon_\infty - \upsilon(r)}{\upsilon_{\text{cl,2}}}\right) \tag{19}$$

Note that for this formula the first density contrast does not denote the outermost value, but only the maximum value $D_{\max}$. Hence $D_{\max} = f_{\max}^{-1}$ = `d.d`, $D_\infty = f_\infty^{-1}$ = `d.d2`, $v_{\mathrm{cl},1}$ = `x.x1`, and $v_{\mathrm{cl},2}$ = `x.x2`.

Examples:

```
DENSCON 20   TAU        0.01   0.1
DENSCON 20.  VELO       0.1    1.
DENSCON 20.  RADIUS     SONIC  2.
DENSCON 10.  HILLIER    0.75S
DENSCON 20.  EXPRADIUS 1.04
DENSCON 25.  EXPTAU     1.0
DENSCON 12.5 NAJARRO    2.5    1.   2.0
```

A different, more flexible way of describing the depth dependence of the density contrast is possible by using `LIST`, e.g,

```
DENSCON = 50.  LIST VELO SONIC 5.  1000.  15.
```

with pairs *depth-coordinate clumping-factor*, where the depth-coordinates can be given as velocities, radii, or optical depths, depending on the keyword after `LIST`. The density contrast before the the `LIST` keyword refers to the default and outermost value while the list has to be given in the direction inside to outside. For the given example this means that $D = 5$ from $v(R_*)$ up to the sonic point, $D = 15$ from the sonic point up to $v = 1000$ and, since no further steps are given, $D = 50$ outwards. To avoid numerically unfavorable harsh steps, a smoothing over three depth points is performed.

DRLINE ALL | *i* | *i* TO *j*

include `DRTRANSIT` lines from file `DATOM` in radiative transfer performed by the `coli` program, so stabilizing lines from autoionization levels are treated as normal line transitions.
Arguments:
`DRLINE ALL` – include all lines
`DRLINE i` – include only the *i*-th `DRTRANSIT` line
`DRLINE i TO j` – include the *i*-th up to the *j*-th `DRTRANSIT` line

Mixing of moment equations with the coeficient `EDDIMIX` is enabled by default, meaning $g+$`EDDIMIX`$\cdot h$, where $h = H/J$ and $g = N/H$. `EDDIXMIX` is autmatically adjusted (minimized), for each radial point (otherwise: `NO EDDIMIX`).

EDDIMIX START =*x.x*

sets start value for `EDDMIX` the automatic adjustment of the EDDIMIX coefficient; default is $x.x = 1.0$.

EDDIMIX FIX=*x.x*

sets the *minimum* value for `EDDIXMIX`; by default, no minimum value is set.

EDDIMIX MAX=*x.x*

sets the *maximum* value for `EDDIXMIX`; by default, no maximum value is set.

Note: By setting all three `EDDIMIX` values (`START`, `FIX` alias minimum, and `MAX`) to the same number, the EDDIMIX coefficient has constantly that vaule throughout all depth points and iterations.

EDDINGTON-GAMMA x.x

setting the Eddington Γ for the hydrostatic part of the atmosphere for `HYDROSTATIC INTEGRATION` instead of calculating it from the model atmosphere. If `HYDROSTATIC INTEGRATION FULL` is chosen, Γ includes the full radiative force.

Option `AUTO` is not supported any more.

This option is only read in `wrstart` and written to the model.

see →`HYDROSTATIC INTEGRATION`, →`LOG GEFF`

EPSILON=*x.x*

threshold for model convergence criterion: As soon as the relative corrections of all levels with population numbers larger than given by `SMALLPOP` (CORMAX in `wruniq.out`) are smaller than this value

$$\texttt{CORMAX} = \left(\max \frac{|n_{\mathrm{new}} - n_{\mathrm{old}}|}{n_{\mathrm{old}}} \right) < \texttt{EPSILON} \qquad (20)$$

the model is considered to be converged. Typically `EPSILON= 0.01`. The `check` command shows log `CORRMAX`.

EXTRAP `[NOTEMP]`

perform `extrap` job based on the method by Ng (1974). The corrections of the population numbers and the temperature stratification of the last 4 iterations are extrapolated (method NG4). Extrapolation is only performed for population numbers larger than `1.E-15` and not for the 5 outermost depthpoints. Option `NOTEMP` disables extrapolation of the temperature.

FLUXEPS=*x.x*

With this option in use, a model will not report as *converged*, and will not proceed to the `formal` job, unless the relative deviation between the actual flux and the correct flux is at all depth points smaller than the value specified by this option (cf. `PLOT HTOT`). This criterion only works when temperature corrections are applied.

GAMMA=*x.x*

sets γ to the given value , e.g. `GAMMA= 1.05`. This parameter controls the Approximate Lambda Operator (ALO) in the following way: a frequency is understood as not beeing a part of the line core if a ray exists, which reaches the outer or inner boundary of the atmosphere on a optical path with $\tau(\nu, r) < \gamma$.

In an expanding atmosphere every photon is redshifted w.r.t. its comoving frame (CMF). Therefore every photon, emitted at the blue line wing has to pass the line core. Therefore the blue part of the line is part of the line core, if $\tau(r) > \gamma$.

The smaller γ, the larger is the core fraction, which are calculated via the ALO method (with core fraction f_{c}),

$$J_{\mathrm{n}ew} = (1 - (1 - \varepsilon) f_{\mathrm{c}})^{-1} \left(J_{\mathrm{F}S} - (1 - \varepsilon) f_{\mathrm{c}} J_{\mathrm{o}ld} \right) . \qquad (21)$$

This acclerates the convergence of the model.

Construction of ALO (Hamann 1987):

$$\Lambda^{*}_{\nu} = 1 - \frac{1 - e^{-\tau_\nu/\gamma}}{\tau/\gamma} \qquad (22)$$

The ALI method can be switched of by $\gamma = 0$.

GAMMAD=*x.x*

similar to `GAMMA` but acts on iron lines only. Note, however, that as default GAMMAD=.0 is set automatically, as soon as temperature corrections are active, because the ALO extrapolation is not compatible with a temperature correction factor which is applied to the iron line trasition rates (cf. Sect. **??**). See also the option `TCORR FERATES OFF`

GAMMAL=*x.x*

similar to `GAMMA` but acts on weak non-resonance linse only.

should maybe be a little bit larger, e.g.
`GAMMAL=5.`

HYDROSTATIC INTEGRATION `[FULL|FULLHD] [MEAN] [REDUCE=x.x] [SONIC] [EPS=x.x]`

Action: perform detailed integration of the hydrostatic equation in `WRSTART` and `SUBROUTINE ENSURETAUMAX`. Default: apply the barometric formula with fixed scale-height.

In `WRSTART`, after calculation of popnumbers and $T(r)$ by subroutine `GREY`, the subroutine `VELTHIN` recalculates the hydrostatic stratification. The geomesh is then setup again, including recalculations by `GREY`. If $\tau_{\rm max}$-enforcement has been enabled (by using the `FIX` or `MIN` option in the `TAUMAX` line) the subroutine `VELTHIN` will be called in a similar way as in `WRSTART`, just without using the `GREY` approximations.

Options:
`FULL` : the radiative force includes all contributions from continua and lines – if `FULL` is not given, only the radiation pressure from Thomson scattering by free electrons is taken into account.
`FULLHD` : same as option `FULL`, but now solving not the hydro*static*, but the hydro*dynamic* equation, i.e. taking into account the *inertia term* $v\,v'$.
`MEAN` : the radiative force is averaged over the hydrostatic part, either for Thomson scattering or all contributions (so together with option `FULL`).
`REDUCE [=x.x]` : damping factor for the corrections of the radiative force between subsequent iterations. (If the REDUCE keyword is set without a value, a damping factor of 0.1 is used. If the REDUCE keyword is missing, the damping factor is 1.0, i.e. no damping is applied.)
`SONIC [=x.x]` : the connection point is forced to coincide with the sonic point; consequently, the velocity gradient is not necessarily continuous there. If the SONIC keyword is followed by a number, the connectionpoint is forced to lie at the velocity $v_{\rm sonic} \times x.x$. $v_{\rm sonic}$ denotes the isothermal sound speed, not accounting for any microturbulence.
`EPS=x.x` : recommended parameter; by default, the fulfillment of the hydrostatic equation is NOT a necessary criterion for considering a model as converged. By adding the EPS parameter, this becomes a convergence critertion. E.g.,

```
HYDROSTATIC INTEGRATION FULL[HD] EPS=0.05
```

requests that the deviation from the hydrostatic equation must be less than 5 percent before a model is considered as being converged.

The HYDROSTATIC line is read in every iteration and can therefore be changed on the fly.

Attention if combined with `LOG GEFF` option:

If `LOG GEFF` is prescribed it must be followed by the option `RADFORCE=FULL` or `RADFORCE=ELECTRON` to be consistent with `HYDROSTATIC INTEGRATION`. The only two allowed combinations of `LOG GEFF` and `HYDROSTATIC INTEGRATION` are therefore:

```
LOG GEFF = x.x RADFORCE=FULL
HYDROSTATIC INTEGRATION FULL
```

which accounts fully for the radiative force or, if only accounting for the Thomson scattering:

```
LOG GEFF=x.x RADFORCE=ELECTRON
HYDROSTATIC INTEGRATION
```

Further options for tests and debugging:

`NORUKU` : emply simpler integration based on the grid points instead of a sub-grid Runge-Kutta integration;

`SMOOTH` : smooth the resulting velocity stratification between neighboring points;

`NONMONO` : allow for non-monotonic velocities in the quasi-hydrostatic part;

`PRINTV` : print a table with the resulting velocity stratification on the new grid, indicating the criterion for each new grid point.

See also →VTURB, →EDDINGTON-GAMMA, →RADGAMMA-START

ITERATION CONTINUUM obsolete, since it is now default.

Sets the BITCONT on TRUE, enables iteration of the continuum at low jobnumbers of 3 . . . 10 (short coli, only steal-help!). Necessary if started with temperature corrections, first COLIs and STEALs are omitted. See also NO CONTINUUM ITERATION

ITMAX=*n*

maximum number of iterations *n* to solve rate equations (population numbers) for every depth point. Usually ITMAX= 50. If a solution after ITMAX iterations can not be found for some depth point, a modify is performed automatically (default) to interpolate population numbers for this depth point from the next (converged) depth points.

The steal program stops if 30 or more of 50 depth points are not converged (AB_steal). However, this stop criterion is not applied if the →NO_AUTO_MODIFY option is set.

JOBMAX=*n*

set the maximum number of jobs to execute, usually JOBMAX = 996., i.e. stopping after 996 iterations; if commented out only one iteration is performed. Every job (steal, wrcont, etc.) is one iteration.

JSTART [BLACKEDGE=*x.x*]

better than LTESTART (?), as beginning from $\tau = 2/3$ a geometrically diluted ($\sim \frac{1}{r^2}$) blackbody radiation field with $T_{\rm eff}$(?) is used for the start approximation of J_ν and this J_ν is used for the first steal to calculate population numbers. If using this option one should give the shorter wavelength cutoff of the emergent flux via the additional argument BLACKEDGE in Å. This cutoff can be determined by comparison with a similar model. In some cases, the proper choice of the BLACKEDGE argument can determine whether a model might fail immediately after the start or run smoothly, since unnaturally high fluxes at very short wavelengths can severely harm the radiative transfer calculation.

Example for a cool model: JSTART BLACKEDGE=504.

For the range τ =TAUMAX (usually TAUMAX = 20.) up to $\tau = 2/3$ the population numbers are calculated under the assumption of LTE.

JSTART is the default for the WRSTART job and is executed by the wrstart program before the adapter program (OLDSTART). Therefore wrstart.plot always contains the population numbers of the start approximation.

LASERV=n [x.x] | [LINES=x.x]

activates *Laser Version n* (default: *n*=1; recommended: $n = 2$)

Background: because we write the radiative transfer equation in second order, negative total opacities must be avoided.

Optional parameter LINES=x.x (in old syntax also without keyword): the minimum total (line + continuum) opacity is limited to *x.x*· continuum opacity (default: $x.x = 0.01$). Hence, the default allows overpopulations in line transitions as long as the remaining opacities extinguish the laser. Strict suppression of lines which would laser for themselves is requested by setting $x.x = 1.0$.

LASERV=2 prevents additionalle that lines with negative opacities are included in the correction amplification via ALOs (see Sect. 13).

Note that in the current code version (since April 2019), *bound-free continua* are strictly prevented from adding negative opacities to the total opacity.

LEVEL *n- m*

for `wrstart` evaluated by `adapter` program, together with `OLDSTART`. E.g.:

```
LEVEL  1-350
LEVEL351-999 SHIFT 36
```

adopts the numbers of the first 350 levels of the `MODEL` file and inserts at the 351th level 36 new levels. Formatted input: after the command `LEVEL` three digits *fixed* for the levelnumber, then "-" and again three digits fixed for level numbers O R free formatted, but with spaces between numbers and '-' (wrh version only).
Further examples:

```
LEVEL  1- 97
LEVEL 98-210 SHIFT 2
LEVEL211-999 SHIFT 4
```

The first 97 levels are kept, between the former 97. and 98. level insert 2 new levels and also insert after the 210. level 2 more new levels (SHIFT = 2 + 2 = 4).
Further options

```
LEVEL n = m
```

Population numbers for level *n* (new) are taken from level *m* (old).

```
LEVEL n NULL
LEVEL n - m NULL
```

Population numbers for level *n* or *n - m* respectively are set to zero (otherwise LTE numbers from the `steal`-program).

```
LEVEL n WEIGHT w
LEVEL n - m WEIGHT w
```

Multiplication of the old population numbers with *w*.

E.g. split an existing level ($n = 7$) into two:

```
LEVEL 1-7
LEVEL 8-999 SHIFT 1
LEVEL 8=7
LEVEL 7 WEIGHT 0.33
LEVEL 8 WEIGHT 0.66
```

LOG G=*x.x*

deprecated, use →`LOG GEFF` or `LOG GGRAV` instead. Due to backward compartibility issues, `LOG G` is interpreted as →`LOG GEFF`.

LOG GEFF=*x.x* RADFORCE=[FULL | ELECTRON]

value of $\log g_{\rm eff}$ (in cgs) for the density scale height (depth grid), e.g. `LOG GEFF= 2.3` for an extreme helium star.
`LOG GEFF= ?` calculates $\log g_{\rm eff}$ from →`MSTAR` and `RSTAR` with initial guessing of Eddington-Gamma Γ.

LOG GGRAV=*x.x*

value of $\log g_{\rm grav}$, i.e. without correction for Γ.
Note: For setting $\log g_{\rm grav}$ and accounting for Γ at the same time, use additionally
→ `EDDINGTON-GAMMA`

LOG L=*x.x*

value of the luminosity in $\log(L/\,\mathrm{L}_\odot)$, e.g. `LOG L=3.7` for 5000 $\mathrm{L}_\odot$.

LOG Q=*x.x*

specification of the parameter Q, which is more common for O-stars, instead of the transformed radius R_t (→`RTRANS`) or mass-loss rate $\dot{M}$ (→`MDOT`), where

$$Q := \frac{\dot{M}\,\sqrt{D}}{(v_\infty\, R_*)^{3/2}} \tag{23}$$

So the transformed radius can be obtained from Q via

$$\begin{aligned} R_\mathrm{t} &= N\, Q^{-2/3}\, v_\infty^{-1/3} & (24)\\ \text{with normalization} \quad N &= \left(\frac{10^{-4}\, M_\odot\, \mathrm{yr}^{-1}}{2500\,\mathrm{km\,s}^{-1}}\right)^{2/3} & (25) \end{aligned}$$

following **?** and also with density contrast $f_\mathrm{cl} \equiv D$ from eq. (54) in **?**.

LTESTART `[BLACKEDGE=x.x]`

the start approximation for J_ν is calculated from a blackbody radiation field of the local temperature in `wrstart`. This J_ν is used for the calculation of the population numbers of the first `steal` running after `wrstart`. This option can also be combined with the OLD TEMPERATURE STRATIFICATION card. The optional argument `BLACKEDGE` (in Å) allows for a cutoff at shorter wavelength to avoid unnatural high fluxes there.

MDOT=*x.x*

value of the mass loss rate in $\log\left(\mathrm{M}_\odot\,\mathrm{a}^{-1}\right)$, e.g. `MDOT= -4.625`, cf. also `RTRANS`. One has to specify either `MDOT`, `MDTRANS`, `RTRANS`, or `LOG Q`.

MDTRANS=*x.x*

value of the transformed mass loss rate $\dot{M}_\mathrm{t}$ in $\log\left(\mathrm{M}_\odot\,\mathrm{a}^{-1}\right)$, e.g. `MDOT= -3.5`. This quantity, introduced by Gräfener & Vink (2013), is an alternative to `RTRANS` and can be seen as the mass-loss rate the star would have if it has $D = 1$, a luminosity of $10^6 L_\odot$, and $v_\infty = 1000\,\mathrm{kms\,s}^{-1}$. The formal definition reads

$$\dot{M}_\mathrm{t} = \dot{M}\,\sqrt{D}\cdot\left(\frac{1000\,\mathrm{km\,s}^{-1}}{v_\infty}\right)\left(\frac{10^6 L_\odot}{L}\right)^{3/4} \tag{26}$$

MLANGER

enforce the use of the M-L-relation by Langer (1989) for WN stars, i.e. if →`WRTYPE` is set or composition of the star corresponds to a WN star. Otherwise the relation by Gräfener et al. (2011) is used.

see →`MSTAR`, →`WRTYPE`

MSTAR=*x.x*

prescribe the value of the stellar mass in $M_\odot$, e.g. for Central Stars of Planetary Nebulae: `MSTAR=0.6`. Needed for calculation of density scale height as an alternative to given parameter `LOG GGRAV`. If neither `MSTAR` nor `LOG GGRAV` of `LOG GEFF` is given, `MSTAR` is calculated from an appropriate mass-luminosity relation for the corresponding composition, i.e. for WC star ($X_\mathrm{C} \geq 0.1$ mass fraction), O star ($X_\mathrm{H} \geq 0.45$), or WN star (else).

see →`WRTYPE`, →`LOG GEFF`, `LOG GGRAV`, `HYDROSTATIC INTEGRATION`

NCORE= *n*

specifies the number of impact parameter points < 1.0 (*core rays*), see Fig. 11 on page 135. Default is `NCORE = 4`

NEWWRC= *n*

solve angle-dependent radiative transfer equation for specific intensity I_ν (only continuum, `WRCONT`) only after each *n*th iteration (default: $n = 5$), i.e. after *n*th `steal`.
Most important: a `COLI+` is initiated, which performs additionally to the usual `COLIMO` – solving the moment equations – a `COLIRAY`, which solves the angle-dependend radiative transfer equation for lines and continua with help of short characteristics ("ray-by-ray"), and by that, it updates the Eddington factors.
Recommended: `NEWWRC= 6`.

NO_AUTO_MODIFY

if this command is given, non-converged depth-points are not interpolated. Instead, the last result of the Newton/Broyden iteration is used. Be aware that the program will not stop even if more than 30 depth points are not converged, as it would do without this option. *Warning:* This should be used under *manual supervision* only and after making a backup copy of the `MODEL` file.

NO EXTRAP

skip extrapolation job `extrap` (usually before `wrcont`)

NO CONTINUUM ITERATION

switches continuum iteration off. See `ITERATION CONTINUUM`.

NO TEMPERATURE CORRECTION

do not apply any temperature corrections. The temperatue stratification remains unchanged. This can also be usefull to "calm" the corrections of the population numbers (convergence problems).

NO TEMPERATURE CORRECTIONS WHILE COR. .GT. -1.0 1.0

do not apply temperature corrections as long as the logarithm of the relative maximal correction of the population numbers are larger than the first value given. And switch off temperature corrections again as soon as logarithm of the relative maximal corrections excess the second value. (s. also `TCORR ALTERNATE`).

NO DATOM OUTPUT FOR CONVERGED MODEL obsolet, see p. 51 `PRINT DATOM IFCONVERGED`

NO POPNUMBER OUTPUT FOR CONVERGED MODEL obsolet, see p. 52 `PRINT POPNUMBERS IFCONVERGED`

NOTDIFFUS

Option to switch off `TDIFFUS` (the latter is now standard). If this option is set, the temperature corrections are perfomed as specified on the `UNLUTEC` cards up to inner boundary. Furthermore, a flux-corrected temperature gradient is used for the diffusion approximation at the inner boundary used in the comoving frame radiative transfer calculations instead of the pure diffusion gradient.

OB-VERS *n* Outer Boundary condition for inward specific intensity I^- (ν always omitted):

0 = no radiation from outer boundary inwards: $I^- = 0$

1 = from damped source function: $I^- = S(1) \cdot (1 - \mathrm{e}^{-\tau_\mathrm{B}})$, where $\tau_\mathrm{B} = \kappa(1)r(1)$

2 = sourcefunction damped by r^{-2}:

$$I^- = \begin{cases} \dfrac{\eta_\mathrm{K}(1)}{\kappa_\mathrm{K}(1)} \dfrac{\tau_\mathrm{B}}{3} & : \quad \tau_\mathrm{B} < 10^{-3} \\ \dfrac{\eta_\mathrm{K}(1)}{\kappa_\mathrm{K}(1)} \min\left(\left[1 - \dfrac{2}{\tau_\mathrm{B}} + \dfrac{2}{\tau_\mathrm{B}^2} - \dfrac{2}{\tau_\mathrm{B}^2}\mathrm{e}^{-\tau_\mathrm{B}}\right], 0.9999\right) & : \quad \tau_\mathrm{B} \geq 10^{-3} \end{cases}$$

4 = (*default*) similar to `OB-VERS 2`,

but S_B is extrapolated via `SFIT(ND, OPA, ETA, ENTOT, XLAM, SBOUND, PLOT)` [2], so:

$$I^- = \begin{cases} S_\mathrm{B}\dfrac{\tau_\mathrm{B}}{3} & : \quad \tau_\mathrm{B} < 10^{-3} \\ S_\mathrm{B}\left[1 - \dfrac{2}{\tau_\mathrm{B}} + \dfrac{2}{\tau_\mathrm{B}^2} - \dfrac{2}{\tau_\mathrm{B}^2}\mathrm{e}^{-\tau_\mathrm{B}}\right] & : \quad \tau_\mathrm{B} \geq 10^{-3} \end{cases}$$

5 = same as `OB-VERS 4`, but $\tau_\mathrm{B} = \kappa 1 r(1)$, where $\kappa(1)$ is only continuum opacity

calculation of I^- from source function:

$$\tau(r) = \int_r^\infty \kappa_\mathrm{B}\frac{r_\mathrm{B}^2}{r'^2}dr' = \frac{\kappa_\mathrm{B} r_\mathrm{B}^2}{r} \tag{27}$$

$$\tau_\mathrm{B} = \tau(r_\mathrm{B}) \tag{28}$$

$$S(r) = S_\mathrm{B}\frac{r_\mathrm{B}^2}{r^2} = S_\mathrm{B}\frac{\tau(r)^2}{\tau_\mathrm{B}^2} \tag{29}$$

$$I^-(r_\mathrm{B}) = \int_0^{\tau_\mathrm{B}} S_\mathrm{B}\frac{\tau^2}{\tau_\mathrm{B}^2}e^{\tau-\tau_\mathrm{B}}d\tau = S_\mathrm{B}\left(1 - \frac{2}{\tau_\mathrm{B}} + \frac{2}{\tau_\mathrm{B}^2} - \frac{2e^{-\tau_\mathrm{B}}}{\tau_\mathrm{B}}\right) \tag{30}$$

OLDSTART [DEPART] [TAU]

for `wrstart` script; the `adapter` program reads in (`decadp.f`) the population numbers from the old `MODEL` file in the `wrdata$kn` directory and replaces the population numbers from the preceding `steal` run of the start approximation by those from the old model.

Default is interpolation of population numbers n_i/n_tot on the radius grid.

See also `OLD T`.

By keyword `DEPART` the departure coefficients[3] from the old `MODEL` file are used instead of population numbers.

By keyword `TAU` interpolation is done on the τ grid instead of the radius grid. If also the temperature should be adapted from old model, please use `OLD T TAU`.

Recommendations:

- Interpolation on radius grid ist appropriate for correct adaption of wind region. E.g. changing mass-loss rate.
- Interpolation on τ grid is appropriate for correct adaption of photospheric region. E.g. if `TAUMAX` is relevant.
- If photosphere matters: If the old model has a very different `TAUMAX` at the end of the iteration (see output in `wruniq.out`), e.g. 30 instead of 20 used by `wrstart`, it is recommended to perform the interpolations of the `adapter` on the τ grid, i.e. `OLDSTART TAU` *and* `OLD T TAU`.
- Deparature coefficients may be useful when changing abundances only.

OLD STRAT

short for `OLD STRATIFICATION`; equivalent to `OLD VELO` or `OLD V`; as effect, the velocity field is taken from the old MODEL for the start approximation, as well as the radius grid and the associated vector

[2] Smooth extrapolation of `MBOUND` outermost points (default: `MBOUND=30`) via polynom of degree `KPLUS1 - 1`, default: `KPLUS1 = 4`, i.e. cubic. Least-square fit of of T_rad from line source function for weighted `MBOUND` depth points, lower weight for outer points.

[3] **D**eparture coefficients are defined as the ratio of the population numbers in the non-LTE case to LTE population numbers:

$$b_{ul} := \frac{n_u}{n_l}\cdot r_{lu} \quad \text{with:} \quad r_{lu} := \left(\frac{n_l}{n_u}\right)_{LTE} \tag{31}$$

I.e. for $b_{ul} = 1$ and same profile function for absorption and emission this is the LTE case again and therefore $S_\nu = B_\nu$.

with the DENSCON table. Note that the velocity field and the DENSCON vector will be updated later when the TAUMAX FIX option is set and requieres a new stratification in order to ensure the requested optical depth TAUMAX at the inner boundary.

Caution: Using `OLD STRAT` together with a change of the terminal velocity v_∞ will result in a linear scaling of the old velocity field to match the new v_∞. If the stratification changes by much, e.g. due to different v_∞ or mass-loss rate, it might be a better start approximation to keep just the radiation pressure in the hydrostatic domain by using the option `RADGAMMA-START=OLD` instead of `OLD STRAT`.

OLD STRATIFICATION

s. OLD STRAT

OLD T

abbrevation for OLD TEMPERATURE STRATIFICATION.
Interpolation over the radius grid (better to use together with OLDSTART).

OLD T TAU

s. OLD T,
interpolation over τ. Should be used together with OLDSTART TAU

OLD TEMPERATURE STRATIFICATION

same as `OLD T`

OLD V

see `OLD STRAT`

OPC= DIAGTAU | DIAGMIN | DIAG | DIAGNC | NONE

different ways of setting up dJ/dS_c (Scharmer weight factor for the continuum) by subroutine `FREQUINT`.

PLANE-PARALLEL TEMPERATURE STRUCTURE

use a plane-parallel temperature stratification for the approximation in wrstart (default is spherical temperature structure.

PLOT ACC `[DEPTHINDEX|RADIUS|VELOCITY]`

plot $\log(a/g)$ as function of the radius grid (depth points). Acceleration a is caused by gas- and radiation pressure, while g is gravitational accerlation. Furthermore the important work ratio a/g is shown. Usually $a/g \leq 1$, hydrodynamical consistent is $a/g = 1$. If $a/g > 1$ the value of the mass loss compared with luminosity and g is too low, rescale at least one of them.

The different options allow to plot this quantity over various depth scales. E.g., with the option `VELOCITY`, $\log(a/g)$ is plotted as function of $v(r)/v_\infty$ within the range $0 \ldots 1$.

PLOT DEPART `[GROUNDSTATES]`

followed by level designations, e.g.
`PLOT DEPART ”HEI 1S1..1” ”HE II....1”`
plots the departure coefficients, i.e.

$$b_i = \frac{n_i}{n_\mathrm{tot}} \left(\frac{n_\mathrm{tot}}{n_i} \right)_\mathrm{LTE} \tag{32}$$

to indicate the departure of the NLTE population numbers from LTE, so in LTE $\log(b_i) = 0$.

With option `GROUNDSTATES` the departure coefficients are automatically plotted for all groundstates.

PLOT FLUX `[TRAD|FLAM|LOGF] [10PC] [XMIN=x.x] [XMAX=x.x]`

delivers a plot of the emergent flux, both of the continuum-only (`wrcont` in black) and including lines (como in red) (but after degrading the spectrum to the coarse frequency grid). The x axis is always in $\log\lambda$[Å], shown is output from `wrcont` (black) und `como` (red).

Options (free in format and position):

`TRAD` in K, i.e. as a radiation temperature (default),
`FLAM` in $erg\,cm^{-2}\,s^{-1}\,\AA^{-1}$, i.e. as $\log F_\lambda$ or
`LOGF` in $erg\,cm^{-2}\,s^{-1}\,\mathbf{Hz}^{-1}$, i.e. as $\log F_\nu$

`10PC` gives the flux at 10pc distance; default is the flux referring to the stellar radius R_*.

`XMIN = x.x` begin of plotted range (in log lambda/Ang); default: `XMIN=2`
`XMAX = x.x` end of plotted range; default: `XMAX = 4`

s. WR-Memo 060306

PLOT GAMMA `[INDEX|RADIUS|TAU|VELOCITY|DENS|ENTOT]`

plot $\Gamma_{\rm rad} := a_{\rm rad}/g$ as a function of the quantity indicated by the given optional third keyword (default: INDEX, i.e. plot over depth points). All contributions to the radiative acceleration are shown on a non-logarithmic scale, thereby focussing on the layers with $\Gamma_{\rm rad} \leq 1$. In addition, the contributions to $a_{\rm rad}$ from the total continuum and from scattering of free electrons are shown. The latter can be calculated in two ways, either directly from the radiative transfer (COLI) or from the electron density (STEAL) multiplied with the L/M-ratio of the model. Since both terms should agree eventually, the difference of these last two curves can also be used as a convergence indicator.

The different options allow to plot this quantity over various depth scales. E.g., with option `VELOCITY` $\log(a/g)$ is plotted as function of $v(r)/v_\infty$ within the range $0\ldots1$.

PLOT HTOT

calls plot `HSUM`: plot flux H in units of $T_{\rm rad}$[kK] over depth grid, with colors:
black, thick line: the intended CMF-flux of the model in radiative equilibrium, obtained by integrating eq. (61) from Hamann & Gräfener (2003), i.e. accounting for the losses due to wind work, starting at inner bound (`HTOTCMF0`)
red, normal line: actual flux H of the model (`HTOTL`)
green, thin line: $T_{\rm eff}$
Radiative equilibrium is therefore reached, if the red line (actual flux `HTOTL`) equals the black line (CMF-flux in radiative equlibirium `HTOTCMF0`), i.e. correct temperature stratification.
Routines: plot written by `PLOTHSUM`, called by `STEAL`

PLOT HTOTC

plot flux as $T_{\rm rad}$ over depth index, done by `como`.

PLOT_INBOX

in the output `steal.plot`, plot only symbols within the plot boundary given by `PLOT_POPRANGE`.

PLOT JNUE `L|K` *n*|*m* `[TRAD]`

calls plot `JNUE`: plot the continuum radiation field $J_{c,\nu}$ (aka XJC variable) in $\rm erg\,cm^{-2}\,s^{-1}\,Hz^{-1}$ or (if TRAD keyword is set) in units of $T_{\rm rad}$[kK] either over log wavelength for a given depth point (keyword L), or over depth index for a given frequency index (keyword K). Note that the fourth parameter, i.e. the specified depth point or frequency index, must be an integer. Multiple lines with PLOT JNUE and different parameters are allowed and can be used to study the radiation field throughout a model stratification.

Example: PLOT JNUE L 15

PLOT JLINE `L|IND` *n*|*m* `[TRAD]`

calls plot `JLINE`: plot the line radiation field $J_{L,\nu}$ (aka XJL variable) in $\rm erg\,cm^{-2}\,s^{-1}\,Hz^{-1}$ or (if TRAD keyword is set) in units of $T_{\rm rad}$[kK] either over log wavelength for a given depth point (keyword L), or over depth index for a given line index (keyword IND). Note that the fourth parameter, i.e. the specified depth point or line index, must be an integer. Multiple lines with PLOT JLINE and different parameters are allowed.

Example: PLOT JLINE L 25

PLOT POP `"LEVEL NN" ...  | GROUNDSTATES [ELEMENT]`

Plot the population numbers of the given levels (from `DATOM`) over density $\log(n_{\text{tot}}/\,\text{cm}^3)$. The depth point grid is added automatically for each plot. E.g. `PLOT POP, "HEI 1S1..1" "HE II....1" "HE III...."` for the ground states of our standard helium.

With option `GROUNDSTATES` population numbers are plotted for each groundstate of all ions. Each element in an extra plot. If additionally `ELEMENT` is given, e.g. HE, then only the groundstates of this element are plotted.

PLOT_POPRANGE `x.x`

defines the minimum of plot range for the population numbers in the output `steal.plot`,
e.g. `PLOT_POPRANGE=1.E-35` plots $\log(n_i/n_{tot})$ from −35 up to 0.

PLOT RTAU1

plot $R(\tau = 1)$ over frequency, done by `como`.

PLOT SIGMAFE `"LEVEL N" "LEVEL M"`

plots the superlevel cross-section σ_{LU} (if existing) between the superleves "LEVEL N" and "LEVEL M" in units of $10^{-15}\,\text{cm}^2$ over wavelength λ in Å. The order of the given levels is not important, they are automatically sorted into lower and upper one. In order to obtain a valid plot, both levels must exist and there must also be a superlevel transition between them. In all other cases, a warning will be issued and no plot will be created.

Example: `PLOT SIGMAFE "G  6....1e" "G  6....5o"`

PLOT T `[TAU] [XMIN=x.x XMAX=x.x YMIN=x.x YMAX=x.x]`

plot temperature stratification over radius grid ($\log\left(\frac{R}{R_*} - 1\right)$, i.e. R_* at $-\infty$.)
Options: `XMIN=x.x XMAX=x.x YMIN=x.x YMAX=x.x` in arbitrary order and selection. Default is to use auto scaling by WRplot.

With option `TAU` the temperature stratification is plotted over τ_{Ross}.

Other options (old options) are ignored.

PLOT UNLU `[ALL|LOCAL|TBALANCE]`

plot temperature corrections over depth grid. With option `ALL` all terms, even if set to 0, are shown for information. Option `LOCAL` enforces plotting of the local term, while option `TBALANCE` enforces plotting of the thermal balance term.

PLOT V

(only wrh) plots $v(r)/100\,\text{k}m/s$ over $\log(r/R_* - 1)$. If a beta-law is connected to a hydrostatic part, the connection point is indicated by a small square.

POPMIN

Typical: `POPMIN = 1.E-27`, population numbers of levels, calculated by solving the rate equation smaller than this value (in n_i/n_{tot}), especially negative ore zero ones, will be set to this value.
wrh: Furthermore, for transition rates between levels set to POPMIN the emissivity will be set to zero, otherwise there can appear “spikes” in the emergent flux from these “unpopulated” levels.

PRINT DATOM

output of the decoded DATOM file, i.e. atomic data, into steal.out or wruniq$1.out respectively, cf. `PRINT ELEMENT`.

PRINT DATOM IFCONVERGED

output of the decoded `DATOM` file to `wruniq.out` only for converged model (`NODATOM=.FALSE.`). Replaces the former option `NO DATOM OUTPUT FOR CONVERGED MODEL`, but with contrary meaning.

PRINT ELEMENT [OXYGEN]

Restricts `PRINT` options to the given element, e.g. oxygen. Instead of the name OXYGEN also the symbol of the element O is possible. Affects
`PRINT POP`
`PRINT DATOM`
`PRINT RATES`

PRINT FLUX

This option is automatically activated at the end of program `steal` in case that the model is *finally converged.*

If set in the CARDS file during iteration, `PRINT FLUX` will create an overwhelming amount of output – even worse since the same action is also executed each time by program `wrcont`.

However, it perfectly makes sense to use `PRINT FLUX` in a separate run of program `steal`, for instance with `OUTPUT ONLY` – the so-called *steal-help* job.

The output demanded by `PRINT FLUX` starts with a long list of fluxes over the coarse frequency grid in various units. Note that these fluxes are *continuum-only* as calculated by program `wrcont`, i.e. corresponding to the black curve returned by `PLOT FLUX`.

Subsequently follows a table with various ionizing fluxes; this table is given first for the continuum flux, and then for the full flux from `COLI`.

Finally follow tables with `ABSOLUTE MONOCHROMATIC MAGNITUDES` and with `ABSOLUTE MAGNITUDES` in various broad-band filters (cf. Fig. 1).

PRINT HTOTC

como will print the total continuum flux as T_{eff}
Format: DEPTH.INDEX | TEFF(L)

PRINT INT

wrcont and como will print mean intensities to stdout
long table for every frequency (continuum, e.g. 769 points), and every depth point (e.g. $L = 1 \ldots 50$).
Format wrcont:
FREQ.INDEX | DEPTH.INDEX | J-NUE | T-RAD | EDDI-f | SPHERICIY-q | EDDI-h
Format como:
FREQ.INDEX | WAVELENGTH | DEPTH.INDEX | J-NUE | T-RAD

PRINT MODINFO

output of model info (to `out` and `modinfo.kasdefs`) as for a stealhelp or converged model (alternatively one may use the CARDS option `PRINT SUMMARY`)

PRINT OPA [*n*]

wrcont and como will print to the corresponding .out file tables with *continuum* opacity, emissivity, source function, etcetera. The table columns are:
FREQ.INDEX | DEPTH.INDEX | OPA | Th-Frac | TAU | R(TAU=1) | MAIN_CONTRIB (PROCESS | LEVEL) | LASER WARN | ETA | S (i.e. source function as brightness temperature)
The tables are grouped per frequency point in the coarse frequency grid (see wrstart.out) i.e. ≈1000 ... 3000 blocks. In each block, the depth index runs from 1 ... `ND`. If the optional number *n* is specified, the output is reduced to depth indices 1, 1+*n*, 1+2*n*, ..., `ND`.

PRINT POP

print population numbers, cf. `PRINT ELEMENT`.

PRINT POP IFCONVERGED

print population numbers to `wruniq.out` only for converged model (`NOPOP=.FALSE.`). Replaces the

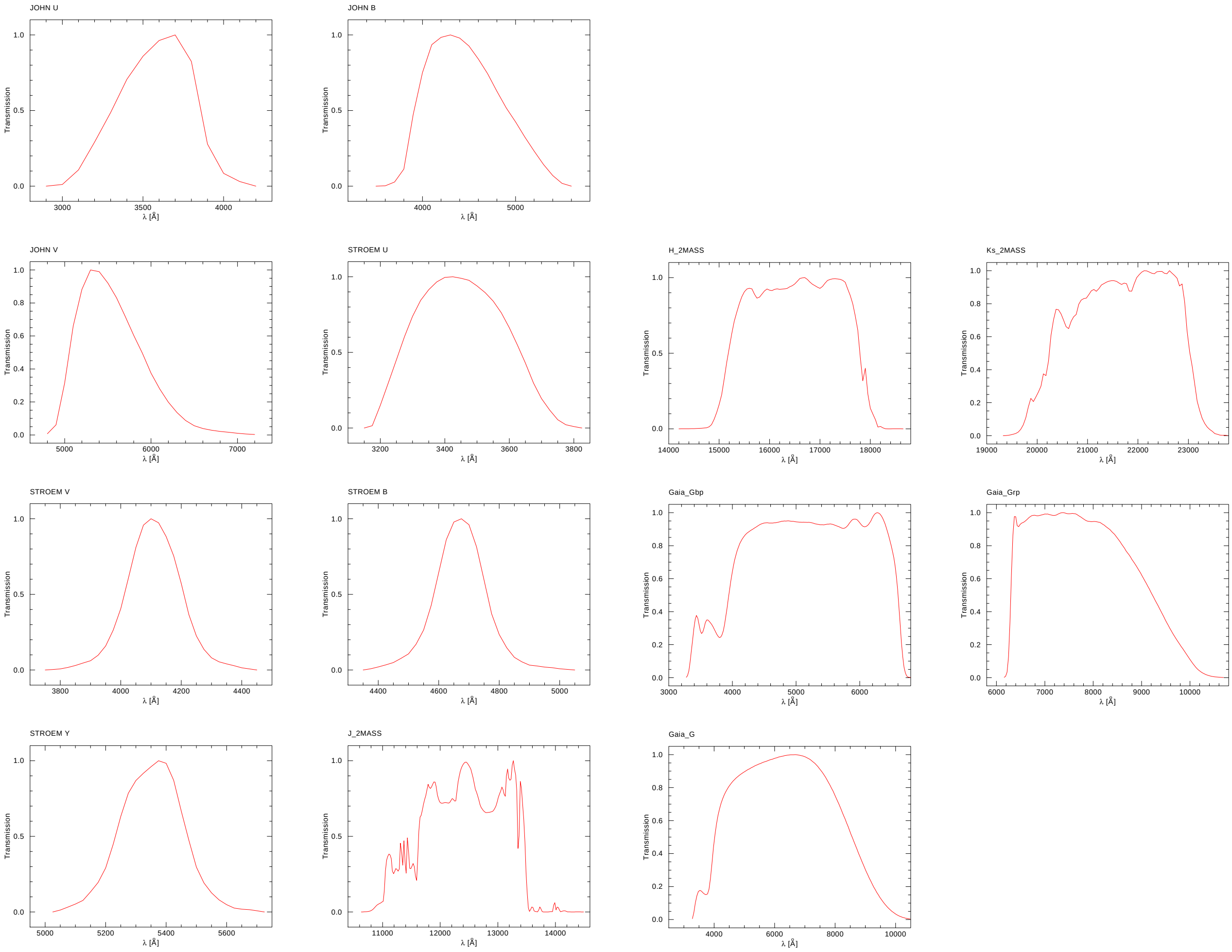


Figure 1: Filter functions used for calculating absolute magnitudes. Those magnitudes are printed by the command `PRINT FLUX`, or automatically at the end of `wruniq.out` when a model is finally converged. Source of the filter functions is mainly the website http://svo2.cab.inta-csic.es/svo/theory/fps3/

former option `NO POPNUMBER OUTPUT FOR CONVERGED MODEL`, but with contrary meaning.

PRINT RATES [i] [FROM d1] [TO d2] [LEVEL="levelname"]

print the rate matrix for every *i*th (integer) depth point between $d1$ and $d2$ into the steal.out or wruniq$1.out respectively. If a levelname is given (wrh version), only the five leading gains and losses (absolute rates, not net rates) for this level are printed E.g.

```
PRINT RATES 1 FROM 35 TO 35 LEVEL="N III5P2.."
```

prints the rate matrix for every ("1") depth point between $L = 35$ and $L = 35$, so for only one, for the level "N III5P2..", so:

```
Leading transition rates from/to level: N III5P2.. at depth index  35

------------- GAINS  ----------------          ------------- LOSSES ----------------

N III3S2.8  LOWER     2.977E-03  RAD           N III3S2.8  LOWER     1.811E-03  RAD
N III5D2..  UPPER     4.755E-04  COL           N III3D211  LOWER     6.809E-04  RAD
N III3D211  LOWER     3.953E-04  RAD           N III4D220  LOWER     6.697E-04  RAD
N 32P2D2.3  LOWER     3.382E-04  RAD           N III5D2..  UPPER     6.331E-04  COL
N III5S2..  LOWER     2.918E-04  COL           N III5S2..  LOWER     4.076E-04  RAD
```

If used in the the `steal`-job, the `OUTPUT ONLY` option must be disabled, as the `steal`-program has to solve the equations instead of just reading the population numbers. This is done with the subroutine `POPZERO`, which otherwise is only used for the very first `steal` run in the `wrstart` job. The more sophisticated algorithms used in the ALI iteration would not be able to provide the pristine rates and rate coefficients `RATCO, CRATE, RRATE`.

`PRINT RATES` can produce huge amounts of output. Thefore, it is inhibited to use it repeatedly within the iteration chain. Instead, `PRINT RATES` makes the next run of the `steal` program to **stop the chain** after the rates have been printed. Moreover, this steal job does not update the MODEL file, i.e. it effectively is for printing only.

routines: `PRI1RAT, STEALCL, DECSTE, POPZERO, NLTEPOP`

PRINT ZERORATES

enables output by subroutine `flag_zerorates` into `wruniq.cpr`. Lists atomic levels and corresponding depth points for which the statistical equation was replaced by $n(j)$ =`POPMIN`.

PROGRAM VERSION: CL obsolete

RADGAMMA-START = x.x | OLD

prescribe an initial guess of the Eddington-Gamma (including the full effect of the radiative force) in order to faciliate the start of the hydrostatic integration in the `WRSTART` if `HYDROSTATIC INTEGRATION FULL` is set.

`x.x` must be smaller than 0.9

`OLD` similar to option `OLD STRATIFICATION`, but instead of the stratification only the full radiative acceleration is adopted from the old model for the start approximation.

see →`HYDROSTATIC INTEGRATION`, →`EDDINGTON-GAMMA`

REDUCE `[.x]`

reduction of all corrections done in steal, i.e. population numbers. The reduction factor `.x` must be between 0. and 1. If 0. or nothing is given, then `.x` is set to 0.5.
Routines: read by `DECSTE`; used by `STEALCL, PRICORR, REDCOR, STHIST.`

RGRID: ND=n_1 NDDENS=n_2 NDVELO=n_3 DLOGTAU=*x.x*

setting up the radius grid (depth points).

E.g. usually used:

```
RGRID: ND= 50 NDDENS= 8 NDVELO= 4 DLOGTAU= 5.0
```

meaning: `ND= 50`: number of depth points; `NDDENS= 8` (old syntax: `ND2= 8`) number of depth points used for density criterion; `NDVELO= 4` (or `ND3= 4`) number of depth points used for the velocity criterion; `DLOGTAU= 5.0` τ-spacing for the remaining points (TAU criterion)

Numbering starts at outer depth point $L = 1$ (RMAX criterion).

Example with more depth points:

ND= 70 NDDENS= 25 NDVELO= 5 DLOGTAU= 3.5

To ensure a proper boundary treatmeant, additional points can be set via the optional CARDS lines `SPECIAL_INNER_POINTS` and `SPECIAL_OUTER_POINTS` (see p. 54) Although such points are called “special”, they are part of the total number of points `ND` which means that they reduce the number of remaining points for the TAU criterion.

RMAX_IN_RSUN

distance to the most outer depth point (RMAX) is given in $R_\odot$

RSTAR `x.x`

specify radius of the star (at τ = `TAUMAX`) in $R_\odot$. You can only prescribe two out of these three quantities: `RSTAR`, `TEFF`, `LOG L`.

Note: The old syntax `RSTAR (SOLAR UNITS): 12.3` is not longer supported.

RTRANS `x.x[DEX]`

specifying the "transformed radius". That quantity is related to the inverse of the wind density $R_\mathrm{t} \sim \varrho^{-1}$. More precisely, it is the ratio between the emission measure of the wind and the area of the stellar surface:

$$R_\mathrm{t} = R_* \left[\frac{v_\infty}{2500\,\mathrm{k}m\,s^{-1}} \bigg/ \frac{\dot{M}\sqrt{D}}{10^{-4} M_\odot yr^{-1}} \right]^{2/3} \tag{33}$$

E.g.: `RTRANS=0.48DEX`, with trailing keyword `DEX` in logarithmic units, or in linear units $R_\mathrm{t} = 3$.

The physical values envolved in R_t can also be given directly, see →`MDOT`, →`DENSCON`, →`VFINAL`, →`RSTAR`. One has to specify either `RTRANS` or `MDOT`.

SMALLPOP=1.E-25

originally used for the CORRMAX criterion

goetz: set to `1.E-25`, population numbers smaller than `SMALLPOP` are not accounted for in the Newton-Raphson iteration; differing from the wrh version, population numbers smaller than 1.0E-12 are not accounted for in the CORRMAX criterion (cannot be set by CARDS options)

wrh: set to `1.E-12` or at maximum `1.E-8`, as relative large corrections for very small population numbers are used for the CORRMAX-criterion, i.e. population numbers smaller than `SMALLPOP` are not accounted for in the `MAX. CORRECTIONS` criterion

SPECIAL_INNER_POINTS *n* [SPACING=*x.x*]

e.g. `SPECIAL_INNER_POINTS 4 SPACING=2.` Inserts additional points between the inner boundary and the next usual grid point. The spacing parameter defines at which fraction between the boundary and the next point the new depth point is inserted. Note that the special points are set iteratively, meaning the if two points should be set with a default spacing of 2.0 the first one will be set at half the distance between the inner boundary and the next point, while the second one will be set at a quarter of the original distance as the next point is now already the first special point. If this line is not found in the CARDS file, four special inner points are inserted. This line is an addition to the `RGRID` line and special points also contribute to the total number of depth points specified in that line.

SPECIAL_OUTER_POINTS *n* [SPACING=*x.x*]

Inserts additional points between the outer boundary and the next usual grid point. The spacing parameter defines at which fraction between the boundary and the next point the new depth point is inserted. The details are described in `SPECIAL_INNER_POINTS` which works in the same way. Note that special outer points contribute to the total number of depth points given in the `RGRID` line and therefore effectively reduce the number of points inserted via τ criterion. If this line is not found in the CARDS file, no special outer points are inserted.

SPHERICAL TEMPERATURE STRUCTURE obsolete, is now default.

See also: →`PLANE-PARALLEL TEMPERATURE STRUCTURE`. Only used in wrstart.

SPLITINVERSION

For inverting the derivative of the rate coefficient matrix (M) the subroutine LINSOL_SPLIT is used instead of LINSOL. The split subroutine uses the fact that M is (almost) block-diagonal and invertes each block instead of the complete matrix. This helps when dealing with larger numbers of atomic levels.

STOP_AUTO_MODIFY | STOP_NONCONVERGED

the `wruniq` iteration is stopped if one ore more depth points are not converged. No subsequent `modify` is executed automatically.

TABLE

(read by `decstar`, used by `wrstart, tabread`).
read in velocity and/or temperature structure from file `TABLE` for not more than 70 gridpoints. Format of file `TABLE` must be:

```
TABLE          001       0.00001       100000.2
XXXXXXXXXXXXXXXIIIXXXXXXXFFFFFFFXXXXXXXFFFFFFFF
= 15X,I3,7X,F7.1,7X,F8.0
```

for depthpoint, v, and T. Alternatively a line can also have the format `I5, 4F10.5` holding information about deptpoint, radius, velocity, and temperature. The table is ended by the keyword `ENDGRID`. The given quantities are then interpolated over the radius.

TAUMAX `x.x [FIX|MIN] [EPS=y.y] [REDUCE=z.z] [CORRLIMIT=w.w]`

Value of the optical depth which should be reached at the inner boundary (largest `L`),i.e. at the “radius” of the star. Usual value is $\tau_{\max} = 20$.
Option `FIX` - ensure the intended optical depth at R_* during `wruniq`-iteration by readjusting the radial depth grid and the velocity field in the `steal` program.
Option `MIN` - same as FIX, but readjustment is only done if the $\tau_{\rm ND}$ drops below the specified value.
Option `EPS=x.x` - accuracy for $\tau_{\max}$ (default: 1.E-4)
Option `REDUCE=x.x` - reduce the applied change to the velocity field by the given factor to obtain a numerically more stable behavior. If the REDUCE keyword is found without a given factor, a damping of 0.2 is used. If the keyword is missing, the factor is 1.0, i.e. no damping is applied.
Option `CORRLIMIT=w.w` - perform readjustments only if log `CORRMAX` is belog the specified *w.w*.

example: `TAUMAX=20.  EPS=1.E-2 FIX REDUCE=0.1`

The optional parameters can be of arbitrary order. Note that `REDUCE` and `CORRLIMIT` require `FIX` or `MIN` to be set.

TCORR ALTERNATE

apply temperature corrections only every second steal-job.

TCORR FERATES OFF

de-activates a temperature correction factor that is applied to the downward line transition rates betwenn the (iron-group) superlevels (see end of Sect. **??**. By default. these correction factors are used because they were found to be beneficial to the convergence. However, they are incompatible with the use of ALOs for iron. Therefore, when the correction factors are used (default), GAMMAD=.0 is enforced automatically as soon as temperature corrections are performed, disregarding any GAMMAD setting in the CARDS file.

TDIFFUS obsolete

This option became standard, and is therefore no longer needed to be given. It requests that the temperature and its gradient at the inner boundary are calculated from the diffusion approximation in constrast to all other depth points.

With the option →`NOTDIFFUS` the innermost temperature is instead obtained in the same way that all other temperatures are obtained, i.e. via the Unsöld-Lucy method.

Using diffusion approximation for temperature adjusting at the inner boundary, i.e. $S_\nu = B_\nu$ for large κ

(inner boundary) after 1st order series expansion of I_ν from formal solution

$$I_\nu \quad = \quad B_\nu + \frac{\mu}{\kappa_\nu}\frac{dB_\nu}{dr} \tag{34}$$

By adjusting the temperature, i.e. dT/dr for the given luminosity / flux at inner boundary via

$$\mathcal{F} = -\frac{16}{3}\frac{\sigma T^3}{\kappa_{\text{Ross}}}\frac{dT}{dr} \tag{35}$$

the incoming intensity is yielded by Eq. (34).

This correction is usually very small therefore it is maybe not necessary to set this card.

Read by `decste` into `bTIDIFFUS`. Subroutine `tdiffus ( steal)` applies T corrections according to diffusion approximation, if `bTDIFFUS` is true.

TEFF=*x.x*

effective temperature in Kelvin at R_* according to

$$L = 4\pi R_*^2 \sigma_{\text{SB}} T^4 \tag{36}$$

e.g. `TEFF= 140000.` for 140 kK.

THIN outdated, use `HYDROSTATIC INTEGRATION` instead (see p. 43)

TMIN outdated syntax for `TMIN-START`

TMIN-START=*x.x*

smallest temperature (in Kelvin) for the intial temperature stratification, i.e. usually at outer boundary. E.g.: `TMIN-START= 15000.`

TWOTEMP *x.x*

This option only works in if OLD TEMPERATURE STRATIFICATION has been set. If a new model is calculated based on an old MODEL file, this option allows you to use a second old model to set up the temperature stratification of the new one. The second model file must be named `model` (in contrast to `MODEL`) in the wrdata folder. The parameter specifies the weight f of the second model temperature in relation to the first one, so the new temperature stratification is

$$T_{\text{new}}(L) = T_{\text{old1}}(L)\,[1-f] + f\,T_{\text{old2}}(L) \tag{37}$$

However, there is hardcoded cutoff at any temperature point if $T_{\text{new}}(L)$ differs more than 20% from $T_{\text{old1}}(L)$.

UNLUTEC [options]

Note: The UNLUTEC options can be distributed over many lines, each one starting with the keyword `UNLUTEC`.

Purpose: Specification of options and parameters for the Unsöld-Lucy temperature corection method. Example:

```
UNLUTEC LOC=0. INT=1. OUT=1. TBALANCE=1. SMOOTH TMIN=8000.
UNLUTEC SMOOTH TMIN=6000. CUTCORR=0.02
```

Options and parameters:

The options:
LOC=x.x
INT=x.x
RMAX=x.x alias **OUT=x.x**
refer to the three different temperature correction terms as described in Hamann & Gräfener (2003), where `x.x` are decimal numbers to specify their weight factor (usually 1.0 or smaller). `LOC` means the local energy balance, `INT` the flux conservation ter,, and `RMAX` the contribution to the flux conservation term from the outer boundary. Defaults are `LOC=0.`, `INT=1.`, and `RMAX=1.`.

Additionally (or alternatively) to the `LOC` term, there is a formulation of the local energy balance as seen from the free electrons (and not from the radiation field), invoked by the option `TB=x.x` (default: 1.0).

In the current form with keywords, e.g. `LOC`, the damping is applied only to the corresponding term.

Note that at least one of the keywords `LOC`, `INT`, `OUT`, `TBALANCE` must appear in order to indicate the new syntax of the `UNLUTEC` command. Not all parameters must be in the same line, it is possible to have up to 10 lines in beginning with `UNLUTEC`.

Keywords:

ACC = x.x determines the accelaration for temperature correction, the smaller this value, the sooner the corrections will be applied (only Götz version).

CORRMAX = x.x works as a damping factor F on the applied temperature corrections ΔT, where F ist `CORRMAX` / $\max_L(|\Delta T|/T_{\rm old})$ over all depth points.

CUTCORR = x.x limits the applied temperature corrections ΔT to 1. + `CUTCORR` for each depth point individually (recommended is 0.02).

EXPTAU=AUTO
Activation of depth-dependent damping factors for all temperature correction terms, with the aim that flux conservation should be established progressingly from inner to outer layers. The factors are:

Term	Damping factor
LOC	$\exp(-\tau/\tau_0)$
INT	$1 - \exp(-\tau/\tau_0)$
OUT	$1 - \exp(-\tau/\tau_0)$
TB	$\exp(-\tau/\tau_0)$

With `AUTO`, τ_0 is set automatically. It can also be defined manually (e.g. `EXPTAU=1.0`).

For test purposes, one can also chose the parameters:

`EXPTAU=AMAX1` → automatic τ_0, but maximal 1.0 part

`EXPTAU=AMIN1` → automatic τ_0, but minimal 1.0

Alternatively to the described damping with an exponential factor, there is a different option `COSTAU=`τ_0 (or `COSTAU=AUTO` etc.) which uses a cosine function for switching the damping factors between 0.0 and 1.0 over a short range in τ.

INT = x.x damping of the integral term (default is 1.0). This term accumulates the deviation from flux conservation from the outer boundary to the current point.

LIMIT-TO-FLUXERR = x.x damping: factor *x.x* by which the maximum relative temperature correction ΔT must stay below the relative deviation from flux conservation `FLUXERR`. All T-corrections are scaled accordingly, i.e. the form of their radial dependence is kept.
Recommendation: `LIMIT-TO-FLUXERR = 0.1`

LOC = x.x damping of the local term (default is 0.0).

MONOTONIC enforces a monotonic temperature structure by interpolation (inward direction) of non-monotonic point pairs (default is off).

OUT = x.x damping of the "RMAX"-term, which gives the deviation from the correct flux at the outer boundary (default is 1.0).

SMOOTH smoothing of the temperatur structure (default is off).

TBALANCE = x.x [TBTAU = x.x] damping of the thermal balance term (default is 1.0). This term accounts for thermal equilibrium of the electrons, i.e. by "heating = cooling" without taking spectral lines into account. This term is thus an alternative to the local term, for details cf. **?**.
In contrast to the LOC term, the "thermal balance" correction does not vanish automatically in the optically thick regime (where the flux corrections are to be prefered). Therefore, the TBALANCE correction is switched off artificially if $\tau_{\mathrm{Ross}} > 0.1$ with a smooth transition starting from $\tau_{\mathrm{Ross}} = 0.01$ The additional option TBTAU specifies the optical depth τ_{Ross}, where the TBALANCE method is switched off. Damping starts at $\tau_{\mathrm{Ross}} = 0.1$ TBTAU. Note that τ_{Ross} includes also line opacities.

TMIN = x.x sets the minimum allowed temperature for the corrections, i.e. every correction leading to a lower temperature is cut off (default is 6000 K).

VDOP

Doppler velocity of micro-turbulence, given in k$m\,s^{-1}$, usually VDOP has to been scaled with VFINAL, i.e. $v_{\mathrm{Dop}} \approx v_\infty/10$, e.g. VDOP=100. for VFINAL=1500. The FEDAT file opacities should be computed with similar VDOP.

VELPAR ...

This line specifies the parameters for the wind-velocity law (β-law), which is

$$v(r) = v_\infty \left(1 - \frac{R_*}{r}\right)^\beta \tag{38}$$

The VELPAR line must carry *all* of the following four parameters:

- VFINAL=<value>
  where the value terminal velocity (v_∞) at outer boundary in km/s;
- VMIN=<value>
  velocity at the inner boundary in km/s. If the TAUMAX option is chosen, the value specified here serves only as initial value for the corresponding iteration in `wrstart`;
- BETA=<value>
  gives the β exponent, e.g. $\beta = 1.0$. There exists a standard version and an alternative, slightly different formulation of the *beta-law* (see Sect. 11.2). This alternative version can be selected by setting β negative, where only the absolute value will be used as exponent and the minus sign is the switch.
- RMAX=<value>
  defines the radius of the outer boundary; by default, the value is in units of the stellar radius RSTAR.
  If the VELPAR line was preceeded by a line
  RMAX_IN_RSUN
  the value is interpreted as solar radii. set.

Example:

```
VELPAR: VFINAL= 3000.  VMIN=1.598 BETA=1.0 RMAX=100.
```

Before 22.03.2014 the VELPAR line had to obey a fixed format.

VMIC = x.x

where `x.x` is the microturbulence in km/s. This option is useful for OB-star models and adds "turbulence pressure" to the hydrostatic equation, i.e. this option is only used together with →HYDROSTATIC INTEGRATION. Internally this quantity is converted to →VTURB.

VTURB = `x.x` obsolete, use instead →`VMIC` (but note that $\upsilon_{\rm mic} = \sqrt{2}\upsilon_{\rm turb}$)
where `x.x` is the turbulence velocity in km/s. This option is useful for OB-star models and adds "turbulence pressure" to the hydrostatic equation, i.e. this option is only used together with →`HYDROSTATIC INTEGRATION`. The gas pressure is than $\propto (a^2 + \upsilon^2_{\rm turb})$.

This line is read by `wrstart`, and `VTURB` is written to the model file, i.e. it cannot be changed later anymore. Watch the occurrence of `VTURB` in various places of the output (`wrstart.out`, `wruniq.out`).

WJCMIN=0. (non default)
if set to zero all continua will be scharmered. Default is `WJCMIN=0.9`, i.e. only for $\tau > 0.9$ continua will be scharmered.

WRTYPE = OB | WN | WC
explicit choice of the `STARTYPE` (alternative keyword) to select a corresponding mass-luminosity relation. Otherwise the type is choosen by its chemical composition (→`MSTAR`). For WC stars the relation by Langer (1989) is used, for OB stars and WN stars the relation from Gräfener et al. (2011) is used. The use of the Langer-relation for WN stars can be enforced by setting the cards option →`MLANGER`.

XJLAPP
approximate radiation field for lines
e.g. `XJLAPP CORE` (old, Koesterke?),`XJLAPP CORE GAMMA=1.5` partially saturated
`XJAPP NEW` saturated line core for radiative transfer
better and new `XJLAPP FINE 20.  12000.  GAMMA=1.05` ($\lambda_{\rm start}$ and $\lambda_{\rm end}$ in units Å) with finer frequency grid and diagonal operator following goetz
`XJLAPP FINE 20.  12000.  GAMMA=1.05` FAST
faster steal-Job, as iron opacities are not taken into account
`XJLAPP NEW`
or
`XJLAPP CORE`
as fallback, if steal stops

XRAY
adding X-ray continuums emission from a shock heated plasma to the model, e.g.:

```
XRAY XFILL 1.E-8 XRAYT 1584893. XRMIN 1.0
```

`XFILL` is the fraction of electrons that are in the hot plasma phase.
`XRAYT` is the X-ray temperatur in K.
`XRMIN` is the minimum radius in R_* at which the shockwaves occurs.

Further methods:
1) Two-temperature X-ray plasma by giving the parameters for a 2nd component, e.g.:

```
XRAY XFILL 120. XRAYT 6200000. XRMIN 1.1 XFILL2 2.E3 XRAYT2 2500000. XRMIN2 1.1
```

2) Multi-temperature X-ray plasma with a given differential emission measure (DEM), distribution is given by a power law with the exponent 1.5 or 2.5, e.g.:

```
XRAY XFILL 12. XRAYT 6200000. XRMIN 1.1 DIFF-EM-EXP 1.5
```

NOTE: While using these additional X-ray emissivity the energy conservation is violated. It is therefore recommended to use X-rays only for an already converged model with correct temperature stratification and for the iterations with X-rays to switch off the temperature corrections, the `TAUMAX` corrections, and the `TDIFFUS` adjustment.

As $T(r)$ etc. should not be altered, is is recommended to use this option with `OLD STRAT` only.
In goetz version: X-ray spectrum by Raymond-Smith code (from file `XDATA`), intended for shocks in

Please add the following to the wrstart-Job:

```
# FETCH AND ASSIGN THE NECESSARY FILES: --------------------
#    echo 'Copy files now (forward)'
     cp $path/XDATA XDATA
..

# ------- END OF JOB ------------------------
# rm fort.*
# rm *exe*
rm XDATA
```

assuming that the XDATA file is in wrdata directory. K-SHELL data must be in the DATOM file.

8. Syntax of the DATOM file

The file DATOM specifies all atomic data for the ions that should be taken into account for the intended model calculation, except for the *Generic Element* representing the iron-group elements. The latter data are provided by the file FEDAT with the help of the *Blanketing* program (cf. Sect. 16.8.1. The DATOM file is usually assembled from the PoWR atomic-data base with the tool newdatom (cf. Sect. 6.1).

In file DATOM, the input columns have fixed positions! Lines starting with an asterisk ("*") are comments.

The content of file DATOM is grouped in the chemical elements (in arbitrary sequence). Only helium is mandatory. Each element section starts with the ELEMENT line, e.g.:

```
************************************************
*KEYWORD--  ---NAME--- SYMB   ATMASS   STAGE
************************************************
ELEMENT     HELIUM     (HE)     4.00     2
*           ------                          *
```

The entry for STAGE specifies an (obsolete?) guess of the main ion's charge that is only used as very first guess.

Subsequent entries then must start with one of the keywords:
LEVEL
LINE
CONTINUUM
DRTRANSIT
K-SHELL

For all ions of the same element, newdatom will group all entries with the same keyword into one block.

8.1. LEVEL entries

Example:

```
*KEYWORD--  ---NAME--- CH WEIG--ENERGY-- EION----- QN
LEVEL       HEI 1S1..1  0    1      0.00 198310.76  1
LEVEL       HEI 2S3..2  0    3 159850.32            2
LEVEL       HEI 2S1..3  0    1 166271.70            2
LEVEL       HEI 2P3..4  0    9 169081.26            2
LEVEL       HEI 2P1..5  0    3 171129.15            2
...
```

Meaning of the entries:

NAME name (unique, 10 characters); the first two characters must be the chemical symbol of this element

CH ion charge

WEIG statistical weight of the level (degenaracy)

ENERGY level energy in wavenumber (cm^{-1}, *Kayser*); the LEVEL entries must be sorted in sequence of increasing ion charge, and within each ion in sequence of increasing level energy.

EION Ionisation energy (only given for ground state)

QN principle quantum number

8.2. LINE entries

LINE entries are required for *all* possible transitions between the bound levels specified by LEVEL entries. The sequence of the LINE entries is arbitrary. Internally, each line transition is assigned to a line index IND numbered in the sequence in which the LINE entries occur in the DATOM file.

Examples:

```
*KEYWORD--UPPERLEVEL  LOWERLEVEL--EINSTEIN  RUD-CEY --COLLISIONAL
COEFFICIENTS--
LINE      NE4p3*2D02  NE4p3*4S01 2.58E-3       KB24    1.00
LINE      NE4p3*2P03  NE4p3*4S01 0.887         KB24    1.00
LINE      NE4p3*2P03  NE4p3*2D02 0.725       X KB24    1.00
LINE      NE4p4 4P04  NE4p3*4S01 -0.23         KB22    0.70
LINE      NE4p4 4P04  NE4p3*2D02             X KB24    1.00
LINE      NE4p4 4P04  NE4p3*2P03             X KB24    1.00
```

Meaning of the parameters:

UPPERLEVEL must correspond to one existing LEVEL name;

LOWERLEVEL must correspond to one existing LEVEL name;

EINSTEIN if positive: $A_{u\ell}$; if preceeded by minus sign: value is the oscillator strength $f_{\ell u}$

RUD if marked with “X”, this line is treated as *rudimental* only; in this case, J_L will not be calculated from line transfer, but interpolated from the background radiation field. Typically used for lines with very small or unknown f-values.

CEY keyword of the formula for the collision cross section to be aplied for this bound-bound transition (cf. Sect. 16.1.2).

COL... COLLISIONAL COEFFICIENTS are further parameters required by the specified collision cross section formula;

8.3. CONTTINUUM entries

Examples:

```
*KEYWORD  LOWERLEVEL ----SIGMA ----ALPHA ----SEXPO -IGAUNT- -KEYCBF-
-IONLEV---
CONTINUUM N III2P2.1    2.16      0.895      2.
CONTINUUM N 32P2P4.2    3.871     2.1302     1.    DETAILN3          N IV 2P3.2
CONTINUUM N 32P2D2.3    3.71      2.457      3.                      N IV 2P3.2
CONTINUUM N 32P2D2.3    1.45      2.519      3.5                     N IV 2P1.3
CONTINUUM N 32P2S2.4    3.42      2.784      3.                      N IV 2P3.2
```

Meaning of the parameters:

LOWERLEVEL must correspond to an existing LEVEL name

SIGMA photo cross section at the threshold, $\sigma_{\rm th}$, in units of 10^{-18} cm^2

ALPHA parameters of Seaton's formula for $\sigma(\nu)$; cf. Sect. 16.4.2.2

SEXPO parameters of Seaton's formula for $\sigma(\nu)$; cf. Sect. 16.4.2.2

IGAUNT keyword of alternative formulas for $\sigma(\nu)$, changing also the meaning of ALPHA, SEXPO; see Sect. 16.4.2.2

KEYCBF keyword for alternative descriptions of the bound-free collisional cross sections;

IONLEV Aim-level of the ionization process; default for `IONLEV` is the ground level of the next-higher ion. If, however, the lower level is doubly excited;, the ionization may lead into an excited state. For each existing LEVEL (except if belonging to the highest ionisation stage of this element) there must be at least one `CONTINUUM` entry.

8.4. K-SHELL entries

optional entries starting with the keyword `K-SHELL`; see Sect. 16.5

8.5. DRTRANSIT entries

optional entries starting with the keyword `DRTRANSIT`; see Sect. 16.6

8.6. Iron-group elements

If iron-group elements shall be considered (*line blanketing*), the file `DATOM` must contains a line like

```
ELEMENT     GENERIC    (G )        3      15
```

This line, usually written towards the end of the `DATOM` file, specifies the range of ionisation stages of the *generic element* that are taken into account. In the given example, the lowest ion is Fe III and the highest Fe XV. The lowest and highest ion are represented by their ground level only ("control level"). For including neutral iron in detail, the lowest `GENERIC` stage must be specified as "0". A fatal error will result if the assigned iron-data file `FEDAT` does not cover the requested ionisation stages.

9. FORMAL_CARDS

The file `FORMAL_CARDS` provides essential input for the program *formal* which is executed by the job *formal* that is automatically submitted after a model has converged. Its purpose is the computation of the emergent spectra in specified wavelength ranges.

Not that the job-scripts `wrjobs/formal`*n* often prefix the `FORMAL_CARDS` with some additional commands, as a kind of shortcut to define default options, see the section begining with: `cat >CARDS << EOF` in those job-scripts. This is definitely a bad habit and should be abandoned in future.

The lines in the `FORMAL_CARDS` must be distinguished into two different groups:

1. **Specifications** for the program `formal`; those are listed in alphabetic order in the subsequent Sect. 9.1.
2. **Atomic data** for the program `formal`, for instance in order to split the spectral lines into multiplet components. The syntax of these data is documented in Sect. 9.2.

It is in general not necessary for the PoWR user to edit the `FORMAL_CARDS` file by hand. Usually, this file is assembled with the help of the `newformal_cards` job (see Sect. 6.2). Instead, the user edits only the file `NEWFORMAL_CARDS_INPUT`. The `newformal_cards` job then

1. passes all specifications (see Sect. 9.1) directly to the `FORMAL_CARDS` file;
2. assembles the atomic data which are needed for synthesizing the requested wavelength range(s) from PoWR's `wrdata-archive`, which contains the files `FORMAL_CARDS.`<*ion*>).

9.1. Specifications for the formal integral

ABS WAVE

an absolute wavelength scale is used for the output spectrum (*default*). The alternative is to plot the spectrum versus $\Delta\lambda$ with the first specified line as reference (cf. →`REL WAVE`).

ALLBROADENING

turns on line broadening for all lines, acoounting for radiation damping as well as pressure broadening. For hydrogen and helium lines, line broadening requires the presence of corresponding files with the broadening data (→`PATH_LEMKE_DAT`, →`PATH_VCSSB`).

BWESEX x.x

sets the half bandwith of the electron redistribution integral in units of the electon doppler velocity (default is 1.0)

CALIB [*option*]

generates, in addition to the plot of the normalized spectrum, the corresponding spectrum in absolute fluxes. Different units can be selected by the *option*:

- `FLAM10PC` (*default*): physical flux f_λ at 10pc distance note: not as logarithm, in contrast to the SED plot!
- `LOGFLAM` log of astrophysical flux per Angstroem, F_λ, through a surface element at stellar radius
- `LOGFNUE` log of astrophysical flux per Hertz, F_ν, through a surface element at stellar radius (this is used internally in the code)
- `MV` flux in magnitudes, with same calibration constant as M_V

Once the flux units have been chosen, they remain in effect until they are overwritten.

CALIB is turned off by →`NOCALIB`.

CONT [*option*]

generates a further plot of the emergent spectrum, but without the lines. The options are the same as for

`CALIB` and specify the flux units. Actually the same units are used for the calibrated spectrum and the continuum, disregarding on which line this is specified.

DXMAX *X.X*

stepwidth, i.e. resolution of the calculated spectrum in Doppler units (cf. Sect. 15.1). Default is 0.3

IDENT [OSMIN=*x.x*]

demands that line-identifications are written to the formal.plot file in WRplot format (\`IDENT ...`). This option is switched on by default, but can be switched off by option → `NOIDENT`

The \`IDENT ...` lines in the plot file are commented out for all lines with an oscillator strength (f-value) less than `OSMIN`. With the optional parameter `OSMIN`=*x.x* this threshold value can be set. Default is `OSMIN`=0.05 However, identifications for lines of hydrogen and helium are never commented, irrespective of their oscillator strength (program version since end of 2020).

IRONLINES

the iron lines are taken into account. Counterpart of options →`NO-IRONLINES`.

IRONVAC

usually, iron lines are calculated for vacuum only. If for the formal iron lines lie in the optical range, i.e. referring to the middle of the range of the current blend block, the vacuum wavelengths of the iron lines are converted to air wavelengths following Morton (1991). This is the default for every new blend block.
By setting the keyword `IRONVAC` before the corresponding blend block, the vacuum wavelengths are not converted to air wavelengths.

IVERSION `TAU | Z`

determines the integration version. Default is the TAU version, which now combines the laser resistance of the Z version with the correct line emissions of the TAU version.

JPFIRST *first-p-point*

JPLAST *last-p-point*

restricting the impact-parameter range for the flux integral to the index range (`jpfirst, jplast`) with the obvious defaults `JPFIRST=1` and `JPLAST=NP`.

In the special case that one requests the same index for `JPFIRST` and `JPLAST`, no integration weight is applied, and the output gives in fact not a flux, but the *emergent intensity* for the ray with impact parameter $p_{\texttt{JFIRST}}$.

Note: the grid of p points is constructed from the radius grid as

`p(1)=0`	(central ray)
...	equidistant values till `p(NP-ND)=1`.
...	as radial points, but inversely sorted `p(NP) = RMAX`

The radius grid (and thus the *p* grid has been established in `wrstart` (see `wrstart.out`), but is updated in the course of iteration. The actual grid can be extracted from the MODEL file by, e.g.,
`msread P`

LIMBDARKENING <parameters> <output options>

(or any keyword starting with `LIMB`): produces plots of intensity I_ν versus impact parameter p for all subsequent spectral ranges (`RANGE` or `BLEND`-block) and adds these plots to formal.plot. Additional listings or output-files are no longer provided.

One out of the three following parameters is mandatory:

`LAM x.x`
intensity taken at the specified wavelength LAM (in Å);

`RANGE x.x y.y`
intensity averaged over the specified wavelength interval (in Å);

`BAND=` <filtername>
with filter being one of: `U, B, V` (Johnson)
`u, b, v, y` (Stroemgren)
`J_2MASS, H_2MASS, Ks_2MASS,`
`Gaia_Gbp, Gaia_Grp, Gaia_G`

Note that the wavelengths or the filter band must be entirely covered by the actual spectral `RANGE` (`BLEND`-block). Otherwise, limb-darkening is skipped for this range with a corresponding warning in the cpr-file.

`OFF`
as first parameters: no limb-darkening computation for any subsequent spectral ranges.

Output options:

`PCUT=y.y`
While per default I_ν is plotted versus impact parameter between 0 and `RMAX` (in R_*), this option restricts the p range to a maximum of y.y

`MIN=y.y`
While per default I_ν is plotted versus all impact parameters out to `RMAX`, this option cuts the p range when the intensity $I(p)$ drops below `x.x` times the value at the disk center.

`MU`
the plot is versus $\mu = \cos\vartheta = \sqrt{1-p^2}$ instead over impact parameter; clearly, p is restricted here to ≤ 1, i.e. the stellar disk.

`NORMALIZED`
(or any keyword starting with `NORM`: while by default the intensities I_ν are plotted in absolute units (erg cm^{-2} s^{-1} Hz^{-1} sterad^{-1}), with this option I_ν is plotted in units of $I_\nu(p=0)$, i.e. the intensity at the center of the stellar disk.

`FILE=<filename>`
writes the limb-darkening profiles as ASCII table into a file – either in the temporary directory of the `formal` job, i.e. `$HOME/work/scratch/formal$kn`, or anywhere else depending on the specified filename that may contain a relative or absolute path. Note that filenames need to be enclosed in quotation marks "..." when containing delimiters like "/".

Example:
`LIMB DARKENING BAND=V MIN=0.01 NORMALISED MU`

LISTONLY

request to only print the recognized lines and multiplets as requested in `FORMAL_CARDS`, but the radiative transfer calculations are not performed.

LPHISTA $= m$

LPHIEND $= n$

These options are only valid in case of wind rotation (`VSINI` option in use). For test purposes, the index range of the φ integration loop can be restricted to the range between `LPHISTA` and `LPHIEND`. By default, the integration is between $\varphi = 0$ and π, corresponding to `LPHISTA=1` and `LPHIEND=`n_φ, where the latter depends on the impact parameter. Note that input values for `LPHISTA` and `LPHIEND` are automatically changed to lie in the range of existing values. For instance, one may request any high number to address the largest existing indices (i.e. $\varphi = \pi$) at any impact parameter.

Note that the range of impact-parameter points can also be restricted for test purposes with the options

→ `JPFIRST` and → `JPLAST`.

The grid of impact-paramer and angle points is visualized in the plot
`$USER/work/scratch/formal<n>/windgrid.plot`
that is always written in case of wind-rotation calculations.

MACROCLUMP [CLUMP_SEP=L_0] [TAU1=τ_1 TAU2=τ_2] | [VELO1=v_1 VELO2=v_2]

enable the macroclumping formalism (optical thick clumps in the formal integral, Oskinova et al. 2007). Parameters are: L_0 (Eq. 12) in Oskinova et al. (2007), the average clump separation $L(r)$ is

$$L(r) = L_0 \sqrt[3]{r^2 w(r)} \qquad \text{with } w(r) = v(r)/v_\infty \tag{39}$$

Examples:

```
MACROCLUMP CLUMP_SEP = 0.2 VELO1=5. VELO2=20.
MACROCLUMP CLUMP_SEP = 0.5 TAU1=0.1 TAU2=0.667
```

NOBROADENING

switches off all line broadening (except of Doppler broadening), thus cancelling `ALLBROADENING` as well as overriding broadening settings at individual lines (`VOIGT`).

NOIDENT

inhibits that line-identifications are written to the formal.plot file. Default is → `IDENT`

NO-IRONLINES

the iron lines are not taken into account, a warning is written. This option is active until option →`IRONLINES` appears.

NOREDIS

Thomson scattering is assumed to be coherent, i.e. line photons are *not* redistributed in frequency due the thermal motion of the electrons. However, since 10-Jun-2014 the continuum is always calculated with frequency redistribution. Default is → `REDIS` meaning that frequency redistribution for line photons is taken into account.

NOWIND `[VELO | RADIUS | TAU] x.x`

This option allows to calculate an emergent spectrum as if the stellar wind is cut off from some point that can be specified.

The point from which on the outer part of the atmosphere is disregarded in the formal integral can be specified in terms of velocity (`VELO`), radius (`RADIUS`), or Rosseland-mean continuum optical depth (`TAU`). Examples:

```
NOWIND VELO   = x.x
NOWIND RADIUS = x.x
NOWIND TAU    = x.x
```

For `VELO`, the numerical value x.x might be replaced by the keywords SONIC (sound speed) or VDOP (Doppler-broadening velocity as used in the MODEL iteration).

For compatibility with the previous syntax, the keyword `VELO` can be omitted; for the same reason, `NOWIND` without any parameters means that the wind is cut off from `RCON`, i.e. the point where the quasi-hydrostatic part of the atmosphere is connected with the wind-velocity law.

In case of using a `SECONDMODEL`, all `NOWIND` specifications refer the first (main) model.

If a `NOWIND` option has been set, it stays in effect for all following `BLEND` blocks (“RANGEs”), untill it is overwritten by another `NOWIND` line, or it is cancelled by
NOWIND OFF

The use of the NOWIND option is reflected in the formal.out and formal.cpr files.

NPHI *number-of-phi-points*: this option is obsolete!
The purpose of this option was to refine the angle integral in case of *wind rotation*. This is now achieved with the `DX` parameter in the `VSINI` line
cf. →`VSINI`, →`LPHISTA`, →`LPHIEND`

marke

PATH_LEMKE_DAT = path
set an alternative path for the file `LEMKE_HI.DAT`, which contains the broadening tables for H I
Default is `/home/corona/wrh/work/wrdata/`.

PATH_VCSSB = path
set an alternative path to the file which contains the tables of Schöning and Butler (Vidal-Cooper-Smith-Schoening-Butler), `VCSSB.DAT` for He II.
Default is `/home/corona/wrh/work/wrdata`. The file has an appoximate size of 0.7 MB and contains coefficients for the pressure broadening of He II lines with main quantum numbers: 2 - 3, 3 - 4 . . . 3 - 10, 4 - 5 . . . 4 - 15.

PLOT VDOP `VELO | TAU | R]`

This command must stand before the `BLEND`-block and will produce a plot of the (depth dependent) →`VDOP` in `formal.plot`, following after the plot of corresponding spectrum. If VDOP differs for the different elements, the plot will display all corresponding curves.

With the option, a different x-axis can be chosen:
`VELO` versus wind velocity
`TAU` versus Rosseland depth
`R` versus radius.
If no option is given the x-axis will correspond to the way how the depth-dependence had been specified.

PRINT OPAL [*n*]
will print to formal.out tables with *line* opacities, emissivity, source function, etcetera. The table columns are:
LINE INDEX | DEPTH INDEX | OPA | LINE/CONT. | TAU | R(TAU=1) | S(LINE) | S(TOTAL)
where source functions S are given as brightness temperature.
This command must stand before the BLEND-block and will produce one such table for each (!) line or subline in the BLEND-block. Hence it should only be used with one or very few lines included, otherwise the output will be overwhelmingly long.

For each line, the depth index runs from 1 ... `ND`. If the optional number *n* is specified, the output is reduced to depth indices 1, 1+*n*, 1+2*n*, ..., `ND`.

RANGE λ_1 λ_2
Stands *before* the `BLEND` block and limits the range of the following block. Wavelengths in Å.

RCOROT `RSTAR r.r | VELOKMS v.v | TAU t.t`
specifies the corotation radius. The keyword indicates the way how this radius is specified. Allowed keywords are:

`RCOROT RSTAR x.x`	(radius in units of RSTAR)
`RCOROT VELOKMS x.x`	(radial wind velocity at corotation radius in km/s)
`RCOROT TAUROSS x.x`	(continuum Rosseland optical depth at corotation radius)

Default is `COROT RSTAR 1.0` Specifying `RCOROT` without `VSINI` leads to an error.
cf. →`VSINI`

REDIS
requests that the frequency redistribution of photons by electron scattering is taken into account, apply-

ing the angle-averaged redistribution function *R* for calculating the emissivity:

$$\eta_{\mathrm{Th}} = \kappa_{\mathrm{Th}} \int J(\nu')\, R(\nu - \nu')\, d\nu' \tag{40}$$

The option is default, but can be switched off with →`NOREDIS`.

REL WAVE

requests that a relative wavelength scale, $\Delta\lambda$, will be used for the output spectrum. The reference wavelength is the first line appearing in the subsequent data. Default is to plot the spectrum over the wavelength λ (cf. →`ABS WAVE`).

SECONDMODEL *<options>*

If specified, the formal integral combines *two* models – the MODEL of the current chain in `wrdata`*n*, and a previously calculated model that has been saved in the directory `PATH` to be specified. The second-model domain can be a double-cone or an embedded sphere, as defined in the *options* (see Sect. 15.7 for details of the geometries).

The *second model* must have been calculated with an identical atomic-data file `DATOM` as the current model. Apart from this requirement, the model parameters can differ arbitrarily.

The options are:

`PATH =` *path to te second model* (mandatory)

`SHAPE = CONE` or `SPHERE`

Mandatory parameters if `SHAPE = CONE` :

`THETA = x.x`
opening half-angle of the cone in degrees

`CONEI = x.x`
inclination angle *i* of the cone in degrees; if $i = 0°$ the cone lies in the plane of the sky. Note that the cone axis always lies in the (y, z) plane, i.e. it can be assumed as aligned with the rotation axis if `VSINI` is activated.

Note: `THETA` *and* `CONEI` *are not allowed to be equal; they must differ by more than 0.1 degrees.*

Mandatory parameters if `SHAPE = SPHERE` :

`ALPHA = x.x`
α in degrees: angle between the direction from the origin to the sphere center and the (y, z) plane (cf. Fig. 19);

`DELTA = x.x`
δ in degrees: the sphere-center's elevation angle above the (x,z) plane (cf. Fig. 18);

Note that α and δ give the direction from the stellar center to the center of the `SPHERE` in spherical coordinates with the y-axis as the polar axis, and (x, z) as the equatorial plane. Longitude $\alpha = 0$ points to the observer, δ is the latitude.

`RSPHERE = x.x`
the sphere's radius in units of R_*:

`DSPHERE = x.x`
distance of the sphere's center from the center of the star, in units of R_*

`PHI_REFINE = x.x`
refinement of the resolution of the azimuthal angle; optional; the larger this value, the finer is the resolution and the longer the computation time; default is `PHI_REFINE = 1.`

`IGNORE_DIFF`
makes the program to continue, even when a difference between the atomic data (`DATOM` or `FEDAT_FORMAL`) of the two models is detected. This will lead to a program crash or to non-sense results, since it ismandatory that there is a one-to-one correspondence of the atomic levels between both models. If, however, differences do not change the level structure, they might be ignored. Example: a corrected f-value in `DATOM`, or the inclusion of ionisation stages in `FEDAT_FORMAL` which are not taken into account in the current model anyhow.

`OFF`
switches off the second-model inclusion for the following range(s); the parameter settings are kept, for the case that `SECONDMODEL` becomes activated again.

The `SECONDMODEL` options can be distributed over more than one input lines, all of them starting with the keyword `SECONDMODEL`. The `SECONDMODEL` treatment is fully integrated into the range-by-range processing of the `FORMAL_CARDS`. All settings remain valid, until they are overwritten.

Examples:

```
SECONDMODEL PATH = "~wrh/science.dir/wnlgrid.dir/models.dir/08-12/"
SECONDMODEL SHAPE=CONE THETA=40. CONEI=60.
SECONDMODEL SHAPE=SPHERE ALPHA=150. DELTA=0. RSPHERE=0.8 DSPHERE=2.
```

SET_POP_ZERO `”levels1” [EXCEPT ”levels2”]`
Set population numbers for all levels beginning with string “levels1” to 0, thus they are not taken into accout in the formal integral. The `EXECEPT` option expects a string “levels2”. Levels beginning with string “levels2” are taken into accout in the formal integral, although they begin with string “levels1”. This option can be reseted before every blend block.
Example:

```
SET_POP_ZERO G   EXCEPT "G  V."
```

(In this example, no iron line is calculated, despite those from Fe v.)
Routines: read in by `DECFORM` in `FORMAL`, used by `SET_POP_ZERO`.

TAUMAX x.x
set the maximum τ for the integration in →`IVERSION TAU` (not used in →`IVERSION Z`). Default is `TAUMAX=10.`
Note: before 3-Jan-2023, the default was 7. Moreover, there was a bug in the code that corrupted the emergent flux at shortest wavelengths (EUV, X-rays), The workaround by setting `TAUMAX` to very high values is now obsolete.

TAUBROAD x.x
sets the minimum optical depth from which on the line wings are truncated for efficiency. Moreover, lines with lower optical depth at their center (evaluated without velocity shift) are flagged as "weak" in the line list in formal.out, and their line identifier (`IDENT` in formal.plot is commented. The default corresponds to `TAUBROAD=0.0001` (since 27-Mar-2025).

TRANS DWL `VELO | VELOLOG | TAUROSS | RSTAR | RSTARLOG]`
this option works only for one single (!) line and creates a plot with the normalized ξ from Hillier (Hillier 1987) over the specified depth scale of the model file, thus gives information about the radial depth mainly contributing to the line formation. The ξ as defined by Hillier is

$$\xi = N_\mathrm{u}(r) r^3 \int_{-1}^{+1} \beta(\mu) \exp[-\tau_\mathrm{c}(\mu)] d\mu \tag{41}$$

being the line emission per radius and thus makes sense only for emission lines. The energy emitted in the line is of course $E \sim \int_{R_\mathrm{core}}^{\infty} \xi d(\log r)$.

Following x-axes can be choosen:

`TRANS DWL`	(log(electron density) - this is default)
`TRANS DWL VELO`	($v(r)$ in km/s)
`TRANS DWL VELOLOG`	(log $v(r)$ in km/s)
`TRANS DWL TAUROSS`	(tau_Ross)
`TRANS DWL RSTAR`	(r/RSTAR)
`TRANS DWL RSTARLOG`	(log (r/RSTAR))

The last version is in fact most sensible, because the emitted line luminosity is proportional to the integral over $\xi \cdot d(\log r)$.

VDOP = *x.x* `MAX=y.y [[VELO1=v.v VELO2=w.w] | [TAU1=t1 TAU2=t2] | [R1=r1 R2=r2]]] | [VELOFRAC [=y.y]]`

setting the velocity parameter v_D for the Doppler broadening of the line opacities and emissivities. Note that the same v_D is applied for all elements, irrespective of their different thermal velocities. This is clearly less physical than setting VMIC (see below), and may rather serve for test purposes, e.g. to compare with results from the previous code version.

If neither VDOP nor VMIC are specified in FORMAL_CARDS, the value of v_D is taken from the MODEL file, i.e. as originally specified in the CARDS file.

The options on the VDOP line allow to specify a depth-dependence of v_D. Their meaning is the same as described for →`VMIC`, but setting here $v_\mathrm{D}(r)$.

The wavelength resolution in formal is determined from the lowest value of VDOP that is encountered among all elements and depth points. This minimum value is now reported as VDOP in formal.out. Note that small Doppler broadening may lead to much longer computing times!

If the model contains iron-group elements, and if these lines are not suppressed in `FORMAL_CARDS` by the option →`NO-IRONLINES`, and if the iron data (`FEDAT_FORMAL`) have been generated with a *smaller* VDOP than the minimum VDOP occuring here, the formal integral is calculated on the finer wavelength grid in order to guarantee that the iron lines are well resolved.

Examples:

```
VDOP=110.
VDOP = 50.  MAX=100.
VDOP = 50.  VELOFRAC=0.1
VDOP = 50.  MAX=100.  VELO1=10.  VELO2=500.
VDOP = 50.  MAX=100.  TAU1=1.  TAU2=5.
VDOP = 50.  MAX=100.  R1=1.3 R2=1.5
```

For checking the effect of VDOP settings, there is a plot option →`PLOT VDOP`.

VMIC = *x.x* `MAX=y.y [[VELO1=v.v VELO2=w.w] | [TAU1=t1 TAU2=t2] | [R1=r1 R2=r2]]] | [VELOFRAC [=y.y]]`

specifies the microturbulence velocity $v_{\mathrm{m}ic}$. The Doppler-broadening velocity v_D is then calculated individually for each element with atomic mass m via the equation

$$v_\mathrm{D}^2(r,m) = v_{therm}^2(r,m) + v_{\mathrm{m}ic}^2(r) = \frac{2kT(r)}{m} + v_{\mathrm{m}ic}^2(r), \tag{42}$$

See also the description for the VDOP command for further notes.

VMIC= *x.x* without any further option sets $v_{\mathrm{m}ic}$ to *x.x* km/s.

The further options on the VMIC line allow to specity a depth-dependence of the mictroturbulence velocity $v_{\mathrm{m}ic}(r)$:

```
VMIC = x.x MAX=y.y
```

VMIC starts with x.x at the inner boundary and grows to y.y at the outer boundary. VMIC grows proportinal the wind velocity, but from a minimal value x.x

`VMIC = x.x VELOFRAC=f.f`
VMIC is a fraction f.f of the wind velocity, but with a minimal value x.x. Default for f.f is 0.05

`VMIC = x.x MAX=y.y VELO1=v.v VELO2=w.w`
VMIC starts with x.x at the inner boundary. Between wind velocities v.v and w.w it switches to the value y.y

`VMIC = x.x MAX=y.y.  TAU1=t1 TAU2=t2`
VMIC starts with x.x at the inner boundary. Between Rosseland optical depths t1 and t2 it switches to the value y.y

`VMIC = x.x MAX=y.y.  R1=r1 R2=r2`
VMIC starts with x.x at the inner boundary. Between the radii r1 and r2 (in stellar radii) it switches to the value y.y

Examples:

```
VMIC = 50.
VMIC = 50.  MAX=100.
VMIC = 50.  VELOFRAC=0.1
VMIC = 50.  MAX=100.  VELO1=10.  VELO2=500.
VMIC = 50.  MAX=100.  TAU1=1.  TAU2=5.
VMIC = 50.  MAX=100.  R1=1.3 R2=1.5
```

E.g. for an O-star model:

```
VMIC=20.  MAX=100.  VELO1=10.  VELO2=500.
```

For checking the effect of VMIC settings, there is a plot option →`PLOT VDOP`.

VSINI *x.x* [RSTAR | RCOROT] [DX=x.x]

gives the projected equatorial rotation velocity $v \sin i$ in km/s. Default is VSINI=0., i.e. no rotation.

Optional parameters:

`RSTAR` or `RCOROT` specifies whether the given value for $v \sin i$ refers to the stellar radius or to the co-rotation radius. By default, this velocity refers to the co-rotation radius `RCOROT`, which might be specified to differ from `RSTAR`!

With VSINI being specified, the program `FORMAL` performs the flux integral not only over the impact parameter p, but also over the azimuthal angle φ. Moreover, the formal integral might refine the grid of impact parameters that fall onto the stellar disk (i.e. with $p < 1$). The larger number of rays for which the emergent intensity has to be calculated leads to much longer CPU time.

`DX=`*x.x* affects the number of grid points (angle as well as impact parameter points). *x.x* denotes a kind of resolution in Doppler units. If `DX` is too large, the emergent profiles of wind lines might display artificial wavy patterns, while a small `DX` value might unneccessarily waste CPU. Default is `DX=1.0`.

cf. also →`RCOROT`, →`LPHISTART`

WAVELENGTH = `AIR | VACUUM`

defines whether for the given range the iron line wavelengths shall be converted to air wavelength. Default is `AIR` for ranges with 3000 Å $< (\lambda_{\max} - \lambda_{\min})/2 <$ 10000 Å, and `VACUUM` else.

XMAX *X.X*

width of the opacity-profile accounted for the calculation of each individual line profile, in Doppler units that refer to the smallest $v_{\rm D}$ occuring in any line anywhere in this model atmosphere (cf. Sect. 15.1). Default is 3.5. In case of `BROADENING` being active, `XMAX` is automatically increased by the subroutine `BANDWIDTH` to the needed value.

XUNIT = MICROMETER | ANGSTROEM

the following wavelengths are given in μm instead of Å (default). The given unit is used until next XUNIT statement. This unit is also used for the output (formal.plot).

9.2. FORMAL_CARDS atomic data syntax

The atomic data parts in the FORMAL_CARDS file are usually assembled from the wrdata-archive by submitting the job newformal_cards (see Sect. 6.2). The data which describe the lines and multiplets which fall into the wavelength range to be synthesized are bracketed by the lines

```
BLEND
```

...

```
-BLEND
```

The wrdata-archive contains separate FORMAL_CARDS files for each individual ion, e.g. FORMAL_CARDS.HE_II for ionized helium. In the following, the syntax of these data is documented.

LINE

setting line card for calculation of a line in the formal program, e.g.:

```
LINE ???   3888.6
UPPERLEVEL=HEI 3P3..8 LOWERLEVEL=HEI 2S3..2

+LINE ???   5015.7
UPPERLEVEL=HEI 3P1.11 LOWERLEVEL=HEI 2S1..3
```

the second argument is optional and gives wavelengthe for the line transition in Å.
By setting the optional keyword VOIGT [γ] with the optional argument γ (radiative decay rate = $1/\tau$), natuaral broadening for this line is enabled.
The next line must contain the designation of the upper and the lower level exactly written as in DATOM. For the first line in a blend the + can be omitted, further lines must begin with a +.The oscilator strenght is read from DATOM

MULTIPLET

setting multiplet card with information how to split levels for substructures of lines (e.g. doublets), which are not already given with LINE. Example:

```
+MULTIPLET ???
UPPERLEVEL=C 22P32P15 LOWERLEVEL=C 23D2D..9
/UPPERLEVEL C 2P32P1/2    2   168729.96
/UPPERLEVEL C 2P32P3/2    4   168748.73
/LOWERLEVEL C 23D2D3/2    4   145549.70
/LOWERLEVEL C 23D2D5/2    6   145551.13
/SUBLINE C 2P32P1/2  C 23D2D3/2  1.79E5    4312.71
/SUBLINE C 2P32P3/2  C 23D2D3/2  1.79E4    4309.21
/SUBLINE C 2P32P3/2  C 23D2D5/2  1.61E5    4309.51
-MULTIPLET
```

In the second line the upper and lower level to be splitted must be called by their names as in DATOM. In the following lines the command UPPERLEVEL an LOWERLEVEL respectively give the detailed information about the levels. First argument given is an arbitrary but unambigous name for the new level, its statistical weight and the level energy. The list must be complete, i.e. the sum of the statistical weights for the upperlevels must be same as given for the unsplitted level in DATOM. Eventually the remaining levels (statistical weights) can be summarized in a “REST” level, not used for line calculation.

The command SUBLINE needs name of the lower and upperlevel created by the commands before. The third argument is the oscilator strength (negative value $\approx$ 1) or Einstein coefficient (positive value $\approx$ 1E5) of the line transition. Optionally a precise wavelength can be added as fourth argument.
By setting the optional keyword VOIGT [γ] with the optional argument γ (radiative decay rate = $1/\tau$), natuaral broadening for this line is enabled.

The splitting is performed with LTE approximation via Boltzmann formula, i.e.

$$\frac{n_i}{n_j} = \frac{g_i}{g_j} \exp\left(-\frac{E_i - E_j}{kT}\right) \tag{43}$$

Since 030909 it is possible to have a line transition within the same level, e.g.:

```
** C III, Multiplet 255 at 21080.59 Ang
+MULTIPLET ???
UPPERLEVEL=C 35L1L.35 LOWERLEVEL=C 35L1L.35
/UPPERLEVEL C 3 5s 1S     1   338514.33
/UPPERLEVEL C 3 5p 1P     3   343258.03
/UPPERLEVEL C 3 5d 1D     5   346658.34
/UPPERLEVEL C 3 5f 1F     7   348859.99
/UPPERLEVEL C 3 5g 1G     9   348859.99         (Energy guessed)
/SUBLINE C 3 5p 1P   C 3 5s 1S   -1.15
-MULTIPLET
```

Upper and lower level in the second line are the same, therefore it is only necessary to split one of them via UPPERLEVEL

WARNING! As line splitting is done in LTE approximation, the case upper and lower level being the same also means, that one gets only an LTE source function without overpopulation of the upper level.

DRTANSIT

handling of dielectronic transitions that lead to visible spectral lines. This mechanism is independent of the occurrence of DRTRANSIT cards in the the DATOM file, but works analogously, i.e. autoionziation levels are taken into account. So far, there exist only entries for C II, e.g.:

```
** C II  4960 A   (MULT.NO. 25)
DRTRANSIT ???
UPPERLEVEL=C 32S1S..1 LOWERLEVEL=C 23P2PP24
/AUTONIVEAU C23D2PP1/2    2   202204.95
/AUTONIVEAU C23D2PP3/2    4   202180.28
/LOWERLEVEL C23P2PP1/2    2   182023.86
/LOWERLEVEL C23P2PP3/2    4   182043.41
/ADDLINE C23D2PP1/2  C23P2PP1/2   -0.1468  4953.85
/ADDLINE C23D2PP1/2  C23P2PP3/2   -0.0367  4958.67
/ADDLINE C23D2PP3/2  C23P2PP1/2   -0.0734  4959.92
/ADDLINE C23D2PP3/2  C23P2PP3/2   -0.1835  4964.73
-DRTRANSIT
```

Part II.
Theory and Algorithms

10. Overview on the model atmosphere theory

10.0.1. wrstart

The job `wrstart` will execute consecutively the following programs:

`wrstart`

- read in model parameter (`decste.f, decstar.f`)
- read in atomic data (`datom.f`)
- calculate missing quantity from $L = 4\pi R_*^2 \sigma_{\mathrm{SB}} T^4$
- get hydrostatic scale height from mass or $\log(g_{\mathrm{eff}})$
- perform TAU-iteration (τ_{Ross}): $v_{\mathrm{min}} \rightarrow$ geomesh $\rightarrow T, \kappa, \vec{n}, \tau$,
  adjust v_{min} until τ = `TAUMAX` (usually `TAUMAX=20.`) is reached at innermost depth point, the photosphere or bottom of the atmosphere
- determine 1st approximation of J_ν (see `LTESTART, JSTART, OLDSTART`

`steal` →called if

- `LTESTART`, `JSTART`, or `LEVEL` option is set, to calculate population numbers from initial J_ν
- write population numbers to `wrstart.plot`

`adapter` →called if

- `OLDSTART` option given to copy population numbers from old `MODEL` file in `wrdata$kn`, renormalize them to new temperature stratification
- `OLD TEMP` option set to copy the temperature stratification from old `MODEL` file in `wrdata$kn`

10.1. Job: wruniq

The main work to solve the non-LTE CMF-radiative transfer problem is done by the `coli` (radiative transfer) the `steal` (rate equations) program.

Due to coherent scattering on free electrons the specific emissivity has the form

$$\eta_\nu = \eta_\nu^{\mathrm{true}} + \frac{1}{4\pi} \oint \kappa^{\mathrm{Th}} I_\nu d\Omega \tag{44}$$

with the Thomson opacity

$$\kappa^{\mathrm{Th}} = n_{\mathrm{e}} \frac{8\pi}{3} r_0 = n_{\mathrm{e}} \frac{8\pi}{3} \left(\frac{e^2}{4\pi m_{\mathrm{e}} c^2} \right)^2$$

which transforms the radiative transfer equation from a simple differential equation to an integro-differential equation. To overcome this problem, the RTE is integrated over $d\mu$ and becomes the 0th moment equation (ME). By integrating over $\mu^n d\mu$ one gets the nth ME. These MEs can be used to eliminate the coherent scattering. The system is then closed by the Eddington factors g, f, and h, which have to be recalculated every n-th iteration. To get started, the `wrcont` solves the angle dependent RTE for the continuum (like static case), i.e. gets specific intensity I_ν, which is used for the ME, calculated by the `como`. Actually this is only needed for the start approximation, but redone after every nth iteration, where n is given via `NEWWRC=n`.

10.1.1. Programs

10.1.1.1. extrap

- if `NO EXTRAP` is not given in `CARDS` file, extrap will extrapolate population numbers and temperature $T(r)$ from last three iterations for acceleration of convergence

10.1.1.2. wrcont

- solve angle-dependent radiative transfer equation (I_ν) for for continuum (this is like in static case) with only ≈ 500 frequency points
- consistent treatment of Thomson scattering (angle dependent)
- →actually only necessary at begin of model iteration

10.1.1.3. como

- solve the moment equations with given Eddington-factors (from file `EDDI`, updated by program `coli`)for the continuum (formal solution from given population numbers)
- advantage: scattering term κ_{Th} cancels out

$$\kappa_\nu(S_\nu - J_\nu) = \eta_\nu - \kappa_\nu J_\nu = \eta_\nu^{\mathrm{true}} + \kappa_{\mathrm{Th}} J_\nu - (\kappa_\nu^{\mathrm{true}} + \kappa_{\mathrm{Th}}) J_\nu = \eta_\nu^{\mathrm{true}} - \kappa_\nu^{\mathrm{true}} J_\nu$$

- main result: J_ν, needed from `coli` for Thomson emissivity $\kappa_{\mathrm{Th}} J_\nu$
- →actually only necessary at begin of model iteration

10.1.1.4. coli

- radiation transfer in the comoving frame (CMF) for continua and lines (formal solution from given population numbers)
- needs pre-calculated Thomson emissivity $\kappa_{\mathrm{Th}} J_\nu$ from `como` when called at begin of model iteration
- calculates Eddington-factors (storage in file `EDDI`)

10.2. The aim: emergent spectrum

The aim is to calculate the emergent flux spectrum from the atmosphere, F_ν^+. With $\mu = \cos\vartheta$, the flux integral is

$$F_\nu^+ = 2 \int_0^1 I_\nu^+ \, \mu \, \mathrm{d}\mu \tag{45}$$

In spherical geometry, the flux integral can be taken over the impact parameter p.

$$F_\nu^+ = R_*^{-2} \int_0^{r_{\max}} I_\nu^+ \, p \, \mathrm{d}p \tag{46}$$

Note that we define p and r in units of the inner-boundary radius R_*, and hence the flux refers to the same reference surface.

The emergent intensity $I_\nu^+(p)$ can be obtained by a *formal intergral* along the ray,

$$I_\nu^+(p) = \int_{z_{\min}}^{z_{\max}} S_\nu(r) \, e^{-\tau} \mathrm{d}\tau \tag{47}$$

with the optical depth τ starting with $\tau(z) = 0$ at $z = z_{\max}$ and

$$\mathrm{d}\tau = -\,\kappa(\nu)\,\mathrm{d}z \tag{48}$$

This integration is carried out in the observer's frame, i.e. all emissivities and opacities have to be evaluated after applying the appropriate Doppler shift for the velocity projection on the ray.

S_ν is the non-LTE source function,

$$S_\nu = \sum \eta_\nu \Big/ \sum \kappa_\nu \tag{49}$$

where the sums go over all non-LTE emissivities η_ν and all opacities κ_ν involved.

The non-LTE emissivities and opacities η_ν and κ_ν are functions of the local population numbers, $n_i(r)$. To establish these population numbers, the model atmosphere must be calculated first.

10.3. The way: model atmosphere

The non-LTE problem implies the consistent solution of two sets of equations, the radiation transfer, symbolically written as a linear mapping $\Lambda : \mathbf{S} \rightarrow \mathbf{J}$,

$$\mathbf{J} = \mathbf{\Lambda}\,\mathbf{S} \tag{50}$$

and the equations of statistical equilibrium, which can be written as a system of algebraic equations for each spatial point,

$$\vec{n} \circ \mathbf{P}(\mathbf{J}) = [0, ..., 0, 1] \tag{51}$$

The following sections shortly summarize the radiative transfer, the rate equations, and the method of finding a consistent solution.

11. WRSTART: Setting up a model

11.1. Radius grid

One of the basic things that has to be established is the generation of the grids as most of the other routines do calculations basend on the them. The `GEOMESH` routine summarizes the generation of the radius (`SUBROUTINE RGRID`), the impact parameters (`SUBROUTINE PGRID`) and the z values. The most complex part is the generation of the radius grid.

Since the revised version, the subroutine `RGRID` decodes all the required information from the CARDS file itself apart from the outer boundary `RMAX` as this value is specified in the velocity field information. It returns the number of depth points `ND`, the array `R` containing radius points in stellar radii - usually called `RADIUS` in other routines - as well as a string array `INCRIT` containing the criterion that has been used for each depth point.

At first the three CARDS lines (`RPAR`, `SPECIAL_INNER_POINTS` and `SPECIAL_OUTER_POINTS`) are decoded. Only the first one is mandatory. If no information about the special points is found, four inner special points and no outer special points are used. The total number of depth points must not be greater than the hardcoded maximum value `NDDIM` which is currently set to 89.

The setup of the radius grid starts with the τ-criterion, meaning that the depth points for this criterion are equally spaced in $\log\tau$ between the outer boundary `RMAX` and the inner boundary `RSTAR`. The radius values are scaled to `RSTAR`, so the inner boundary is equal to 1. In a first step, a fine grid of 1000 points is generated, consisting of an outer part with 800 and an inner part with 200 points. The outer part is calulated with the formula

$$R_{\mathrm{fine},i} = R_{\max}^{\frac{800-i}{799}}. \tag{52}$$

The inner part is handled explicitly to achieve a much finer spacing. For $i \geq 800$ the formula

$$R_{\mathrm{fine},i} = R_{799}^{\frac{1000-i}{201}} \tag{53}$$

is used. Together with this radius fine grid a first approximation of τ is done, using the density obtained from the equation of continuity via

$$\rho_i = \frac{\dot{M}}{4\pi}\frac{1}{R_i^2 v(R_i)}. \tag{54}$$

However, the factor $\frac{\dot{M}}{4\pi}$ is constant and is therefore neglected in the code. The optical depth is calculated by a numerical integration. For τ the relation $\tau = \int \tilde{\kappa}\rho dr$ is used with the assumption of a constant $\tilde{\kappa} = 1$. The numerical version is therefore written

$$\tau_i = \tau_{i-1} + \frac{1}{2}\left(R_i - R_{i-1}\right)\left(\rho_{i-1} + \rho_i\right) \tag{55}$$

The final value τ_{1000} is used for scaling the τ-spacing afterwards. The calculated τ-values are not physical correct values as we have neglegted constant factors like $\frac{\dot{M}}{4\pi}$ in this calculation. However, `RGRID` should just provide radius values and the scaling with τ_{1000} ensures that all constant factors will eventually cancel out. The number of depth points placed by the τ criterion is called `ND1`. One Point is always subtracted for the outer boundary, for the rest their τ value is obtained from the relation

$$\tau_L = \tau_{1000} 10^{\left(\mathtt{DLOGTAU}\left(\frac{L-2}{ND1-2}-1\right).\right)} \tag{56}$$

R(L) is then determined by $R_{fine}(\tau_L)$ which is done by linear interpolation of the tau values over the radius fine grid.

After the generation of the τ-spaced points the radius grid already covers the whole range between the boundaries. The other criteria now insert points in between the existing ones to cover areas where certain values like density or velocity would chance to drastically between two τ-spaced depth points.

The first additional points ND2 are calculated by the velocity criterion. The criterion checks if the density increasement increases inwards. If not and no larger increasement is found afterwards in the inner part, a new grid point is inserted at half way (on radius scale) between the point with the largest density incresement and the point right before. To give an example, let ρ_i – here again calculated with the neglection of $\frac{\dot{M}}{4\pi}$ – be the density at the radius grid point i. The density loop now checks for every grid point except the outmost one if

$$\frac{\rho_i}{\rho_{i-1}} \geq Q \tag{57}$$

with Q usually being the ratio of ρ_{i-1} and ρ_{i-2}. However, if the condition is not met, Q is not updated and in the next cycle the ratio of ρ_{i+1} and ρ_i is compared to Q which would still be the ratio of ρ_{i-1} and ρ_{i-2}. If none of the ratios would be greater or equal to Q, a new depth point would be inserted between i and $i-1$. If finally another ratio is greater than the current value of Q, Q would be updated and no depth point would be inserted here. Instead, if the depth increasement is constantly increasing inwards, the depth criterion grid points would be finally added right before the last depth point that marks the inner boundary.

In the next step, the points from the velocity criterion (ND3) are inserted. The method is roughtly the same as for the density criterion, but now the velocity differences (instead of the ratios) are compared. As long as the velocity difference is increasing inwards no point is inserted. If the velocity difference decreases and does not increase again afterwards, an additional depth point is set in the middle (on the radius scale) of this largest difference.

After the usual criteria special inner and outer points are set, starting with the inner ones. These points are simply inserted right before the inner (or outer) one and the next grid point. The points are set iteratively and the spacing can be specified in fractions of the distance between the boundary and the next grid point. That means, if one special inner point with a spacing of 3 it will be set at one third of the distance between R_* and the next grid point, starting from R_*. If two special points are requested, the first one would be set at $\frac{1}{3}$ while the second one is at $\frac{1}{9}$. The same is exactly the same for the outer special bound, apart from the fact that the distances are calculated inwards from the outer boundary here. If no cards lines for special points are set, the default number of special outer points is zero while the number of special inner points is four. The default spacing is 2 in both cases which means that the first point is inserted right in the middle. Note that even if these points are referred as "special" they are part of the total number of depth points ND and therefore reduce the number of points set via the τ criterion if ND is not increased itself.

After complete insertion an additional smoothin is done for all grid points except the boundaries and the special points to achieve a more equally spaced grid on a logarithmic scale. The smoothing relation is as follows:

$$\log R_{i,\text{smooth}} = \frac{1}{4}\left(\log R_{i-1} + \log R_{i+1} + 2\log R_i\right) \tag{58}$$

The generation of the impact parameter grid is straight forward. The number of impact parameters ND is equal to the number of depth points plus a hardcoded number of core rays (currently NC= 4). The resulting array P contains the equally spaced core rays first, followed by the ones corresponding to a depth point in reversed order. Therefore the array index in P starts at the inner part in contrast to the radius grid indices starting at the outer boundary. The z values are calculated from radius and impact parameters via pythagoras.

11.2. Velocity field

The velocity field is defined in WRSTART before the geometrical mesh has been established. First, SUBROUTINE INITVEL is called in order to provide a couple of parameters. If requested, these parameters are improved by the SUBROUTINE VELTHIN. With these parameters, which are transported via the block COMMON /VELPAR/, the FUNCTION WRVEL(R) gives the velocities. After the radius mesh has been defined, this function is called to calculate the velocities at each radius point, which are stored in the vector VELO. Only this vector is written to the MODEL file. The FUNCTION WRVEL is only available in the WRSTART and the STEAL program.

Input parameters for defining the velocity field are the velocity at the inner boundary, $v(r = R_*) = v_{\rm min}$, and at the outer boundary, $v(r = R_{\rm max}) = v_\infty$.

Note that the velocity field may be re-calculated in WRSTART iteratively if a specific TAUMAX (optical depth of the inner boundary) is requested by the corresponding option. Moreover, if the user asks to maintain this TAUMAX fixed (see TAUMAX FIX), the velocity field is re-calculated within the model iteration (program STEAL).

The velocity field consists of two parts. In the inner part, a hydrostatic density stratification is assumed, while for the outer part a *beta law* is adopted. The connection point between both domains, $r_{\rm con}$, is determined from the conditions that the velocity field and its gradient must be continuous. For thicker winds, the hydrostatic domain may not exist

11.2.1. Hydrostatic domain

According to the continuity equation, a hydrostatic density stratification implies a corresponding velocity field. In the standard described now first, a few simplifications are made. The more accurate integration is requested by the CARDS line HYDROSTATIC INTEGRATION and will be described in Sect. 11.2.1.2.

11.2.1.1. Simple approach: the barometric formula In a simplified approach (subroutine INITVEL, the *barometric formula* is adopted, which implies:

1. constant temperature $T \equiv T_{\rm eff}$;
2. constant effective gravity $\log g_{\rm eff}$ (neglecting spherical extension);
3. constant mean particle mass μ; in WRSTART μ is calculated from the "typical ionization stage" of each element as given in the atomic date file DATOM, while later this number is from the real ionization balance at the innermost depth point.

Under these assumptions, the scale height (in units of the stellar radius) becomes

$$H_0 = \left. \frac{a^2 + v_{\rm mic}^2}{g_{\rm eff}} \right/ R_* \tag{59}$$

where $a = \sqrt{kT/(\mu m_{\rm H})}$ is the isothermal sound speed and $v_{\rm mic}$ the turbulent velocity which might be specified by a corresponding line in the CARDS file. (Note: for historical reasons, in the MODEL file we store the variable VTURB which is VMIC* $\sqrt{2}$.)

With $v_{\rm min}$ being the velocity at the inner boundary $r = 1$ (in units of R_*), the velocity field becomes

$$v(r) = v_{\rm min} \exp \frac{r-1}{H_0} \tag{60}$$

Note: One of the main technical advantages of the static approach is that it works without any grid. This is important for the subroutine RGRID that uses the static results of INITVEL to setup the radius grid. In contrast, the solution via hydrostatic integration requires an already existing grid, so the static results are needed in order to have a start approximation.

11.2.1.2. Full hydrostatic integration For WR stars, this *hydrostatic domain* is of no importance; often it may not even be reached within the calculated range of optical depths. For stars with mainly photospheric spectrum, however, one should consider a more accurate treatment, including an iterative update of the temperature stratification. Especially the mean atomic mass should be updated acording to the actual ionization stratification.

A numerical integration of the hydrostatic equation accounting for the temperature equation is done in the `SUBROUTINE VELTHIN`. This can be optionally chosen by adding the `CARDS`-option `HYDROSTATIC INTEGRATION`.

To obtain the velocity field $v(r)$ we start from the hydrostatic equation

$$\frac{dp}{dr} = -\frac{p(r)}{H(r)} \tag{61}$$

where the scale height $H(r)$ now depends on r,

$$H(r) = \frac{\bar{v}_{\rm th}^2(r) + v_{\rm turb}^2}{g_{\rm eff}(r)} \bigg/ R_* \tag{62}$$

Note that none of the simplifications implied by the berometric formula are kept. Instead of constant gravity we now write

$$g_{\rm eff}(r) = g_{\rm eff}\frac{1}{r^2} \tag{63}$$

To solve Eq. (61) we use the ansatz

$$p(r) = p_0 \exp\left(-\frac{r-1}{H_0} + b(r)\right) \tag{64}$$

where H_0 is the scale height from the barometric formula, and $b(r)$ an arbitary function which accounts for the deviations from that formula. The derivative of the previous equation gives

$$\frac{dp}{dr} = p\frac{d}{dr}\left(-\frac{r-1}{H_0} + b(r)\right) = p\left(-\frac{1}{H_0} + \frac{db}{dr}\right) \tag{65}$$

The difference between this equation and Eq. (61) is

$$\frac{db}{dr} = \frac{1}{H_0} - \frac{1}{H(r)} \tag{66}$$

Eq. (66) can be solved numerically by integration from the inner boundary with the boundary value $b_{\rm ND} = b(R_*) = 0$:

$$b_i = b_{i+1} + \Delta r\left[\frac{1}{H_0} - \frac{1}{H(r_{i+1})}\right] \tag{67}$$

To obtain the velocity $v(r)$ we employ the equation-of-state,

$$(\bar{v}_{\rm th}^2(r) + v_{\rm turb}^2)\,\rho(r) = p(r) \qquad \text{and the continuity equation} \qquad v(r) \propto \frac{1}{\rho(r)\,r^2} \tag{68}$$

to obtain the velocity law

$$v(r) \propto (\bar{v}_{\text{th}}^2(r) + v_{\text{turb}}^2)\, r^{-2} \exp\left(\frac{r-1}{H_0} - b(r)\right) \tag{69}$$

The proportionality constant is obtained via the inner boundary $v(R_*) = v_{\text{min}}$. The final formula for the hydrostatic domain is therefore

$$v(R) = v_{\text{min}}\, \frac{\bar{v}_{\text{th}}^2(r) + v_{\text{turb}}^2}{\bar{v}_{\text{th}}^2(r=1) + v_{\text{turb}}^2}\, r^{-2}\; \exp\left(\frac{r-1}{H_0} - b(r)\right) \tag{70}$$

11.2.2. Wind domain

For the outer part of the atmosphere, i.e. for the stellar wind, we adopt the *beta-law* for the velocity field, which we write in the form

$$v(r) = p_1 \left(1 - \frac{1}{r + p_2}\right)^{\beta} \tag{71}$$

Note: There is an alternative, slightly different version of the *beta-law*:

$$v(r) = p_1 \left(1 - \frac{p_2}{r}\right)^{|\beta|} \tag{72}$$

This alternative version can be selected by setting β negative, where only the absolute value will be used as exponent and the minus sign is the switch. The rest of this subsection refers only to the first version Eq. (71).

The two parameters p_1 and p_2 are fixed by the two boundary values,

$$v(r_{\text{max}}) = v_\infty \text{ and } v(r_{\text{con}}) = v_{\text{con}} \tag{73}$$

Inserting these conditions in the above law, we obtain

$$v_{\text{con}} = p_1 \left(1 - \frac{1}{r_{\text{con}} + p_2}\right)^{\beta} \tag{74}$$

and

$$v_\infty = p_1 \left(1 - \frac{1}{r_{\text{max}} + p_2}\right)^{\beta} \tag{75}$$

The parameter p_1 can be eliminated by dividing the first by the second condition. With the definition

$$Q = \left(\frac{v_{\text{con}}}{v_\infty}\right)^{1/\beta} \tag{76}$$

this yields

$$Q = \frac{1 - \frac{1}{r_{\text{con}} + p_2}}{1 - \frac{1}{r_{\text{max}} + p_2}} \tag{77}$$

Now comes some algebra till we obtain a quadratic equation for p_2:

$$Q - \frac{Q}{r_{\text{max}} + p_2} = 1 - \frac{1}{r_{\text{con}} + p_2} \tag{78}$$

$$\frac{1}{r_{\rm con} + p_2} - \frac{Q}{r_{\rm max} + p_2} = 1 - Q \tag{79}$$

Now we multiply with the denominators:

$$r_{\rm max} + p_2 - Q r_{\rm con} - Q p_2 = (1 - Q)\,(r_{\rm max} + p_2)\,(r_{\rm con} + p_2) \tag{80}$$

or

$$(1 - Q)\,p_2 + r_{\rm max} - Q r_{\rm con} = (1 - Q)\,(r_{\rm max} + p_2)\,(r_{\rm con} + p_2) \tag{81}$$

This allows to divide by $(1 - Q)$

$$p_2 + \frac{r_{\rm max} - Q r_{\rm con}}{1 - Q} = (r_{\rm max} + p_2)\,(r_{\rm con} + p_2) \tag{82}$$

which finally provides the quadratic equation in its normal form:

$$p_2^2 + p_2\,(r_{\rm max} + r_{\rm con} - 1) + r_{\rm max}\,r_{\rm con} - \frac{r_{\rm max} - Q r_{\rm con}}{1 - Q} = 0 \tag{83}$$

Thus the equation has the form

$$x^2 + 2S\,x - T = 0 \tag{84}$$

with the coefficients

$$S = \frac{1}{2}(r_{\rm max} + r_{\rm con} - 1) \tag{85}$$

and

$$T = \frac{r_{\rm max} - Q r_{\rm con}}{1 - Q} - r_{\rm max}\,r_{\rm con} \tag{86}$$

and the (positive) solution

$$p_2 = -S + \sqrt{S^2 + T} \tag{87}$$

The other parameter, p_1, is now easily obtained from the β-law at the outer boundary (Eq. 75), which gives

$$p_1 = v_\infty \bigg/ \left(1 - \frac{1}{r_{\rm max} + p_2}\right)^{\beta} \tag{88}$$

11.2.2.1. Two-beta-law Optionally the velocity in the wind domain can be the sum of two beta-law terms with different exponents β and β_2, i.e.

$$v(r) = p_1 \left(1 - \frac{1}{r + p_2}\right)^{\beta} + p_{1-2} \left(1 - \frac{1}{r + p_{2-2}}\right)^{\beta_2} \tag{89}$$

The mixture is defined by the input parameter $f_{\beta 2}$ (in the code: `BETA2FRACTION`); at $r_{\rm max}$ the contribution of the second beta-term to the total velocity is $f_{\beta 2}\ v_\infty$. Hence, we must replace v_∞ by $(1 - f_{\beta 2})\ v_\infty$ for the calculation of the parameters p_1 and p_2 of the first beta-term in Eqs. (76) and (88).

For the corresponding parameters of the second beta-term, we have the condition

$$f_{\beta 2}\ v_\infty = p_{1-2} \left(1 - \frac{1}{r_{\rm max} + p_{2-2}}\right)^{\beta_2} \tag{90}$$

As the second condition we demand that the second beta-term vanished at the connection point $r_{\rm con}$, i.e.

$$0 = p_{1-2}\left(1 - \frac{1}{r_{\rm con} + p_{2-2}}\right)^{\beta_2} \tag{91}$$

which leads immediately to

$$p_{2-2} = 1 - r_{\rm con} \tag{92}$$

and

$$p_{1-2} = f_{\beta 2}\, v_\infty \Bigg/ \left(1 - \frac{1}{r_{\rm max} + p_{2-2}}\right)^{\beta_2} \tag{93}$$

11.2.3. Connection point

The connection point $r_{\rm con}$ between the hydrostatic and the wind domain is now determined from the condition that in this point also the velocity gradients connect continuously. The gradients are

$$v'_{\rm stat}(r) = v_{\rm min} \exp \frac{r-1}{H} \Bigg/ H \tag{94}$$

and

$$v'_{\rm wind}(r) = \frac{\beta p_1}{(r+p_2)^2}\left(1 - \frac{1}{r+p_2}\right)^{\beta-1} \tag{95}$$

for the (simple) hydrostatic and the wind domain, respectively. If the full hydrostatic integration is done, there is no analytical solution and the inner gradient is obtained by using spline interpolation (`SUBROUTINE SPLINPOX`) on the hydrostatic velocity field. In the wind part we have omitted the contribution from the second beta-term (if present), because its gradient is zero at the connection point anyhow. This holds only for beta-law exponents larger than unity; therefore we allow only $\beta_2 > 1$.

We define a `FUNCTION DELTAGR` (or `DELTAGRTHIN` in the case of full hydrostatic integration) as the difference between the two gradients $v'_{\rm wind} - v'_{\rm stat}$. We then use a *regula falsi* to find $r_{\rm con}$ where this function (`DELTAGR` or `DELTAGRTHIN` respectively) vanishes. In the latter case, the regula falsi is directly implemented in the `SUBROUTINE VELTHIN` due to the variety of required parameters. In the static approach the function `DELTAGR` is handed as formal parameter into our standard `SUBROUTINE REGULA`. With this new value of $r_{\rm con}$, the parameters of the beta law (p_1, ...) are updated, and this is iterated till convergence of $r_{\rm con}$. This is done to ensure $v_{\rm wind}(r_{\rm con}) = v_{\rm stat}(r_{\rm con})$.
The *regula falsi* requires two initial values which enclose the root. In order to chose this interval, a closer discussion of the velocity gradients is necessary:

The run of the beta-law gradient (Eq. 95) is qualitatively different depending on β. For $\beta < 1$ the gradient is $+\infty$ at $r = 1 - p_2$ and is monitonically falling with increasing radius. For $\beta = 1$ it is also monotonically falling, but starting from a finite value ($v'(1-p_2) = p_1$). The gradient of the static part is exponentially growing. Therefore, an intersection point is granted (see Fig. 2).

In the case of $\beta > 1$, the velocity $v(r)$ has a turning point, i.e. the gradient has a maximum. The radius where this maximum is located, $r_{\rm gradmax}$, can be obtained from setting the second derivative of the velocity to zero:

The first derivative Eq. (95) can be written as

$$v'_{\rm wind}(r) = \beta\, v(r) \Bigg/ \left[(r+p_2)^2\left(1 - \frac{1}{r+p_2}\right)\right] = \beta\, v(r) \Big/ \left[(r+p_2)^2 - (r+p_2)\right] \tag{96}$$

Differencing with the quotient rule yields

$$v''_{\rm wind}(r) = \frac{\beta\, v'\, [...] - \beta\, v\, [...]'}{[...]^2} \tag{97}$$

with [...] being the bracket from the previous equation. The derivative of the bracket is

$$[(r + p_2)^2 - (r + p_2)]' = 2r + 2p_2 - 1 \tag{98}$$

Setting now $v'' = 0$ and expressing v' with Eq. (96) yields

$$\beta^2 v = \beta v[2r_{\text{gradmax}} + 2p_2 - 1] \tag{99}$$

and, after cancelling βv, finally

$$r_{\text{gradmax}} = 1 - p_2 + \frac{\beta - 1}{2} \tag{100}$$

The gradient of the static law will have two intersection points with the wind law (see Fig. 3). We want to select always the larger of the two solutions. Therefore we search, by increasing r from $\max(1, 1 - p_2 + \epsilon)$ in

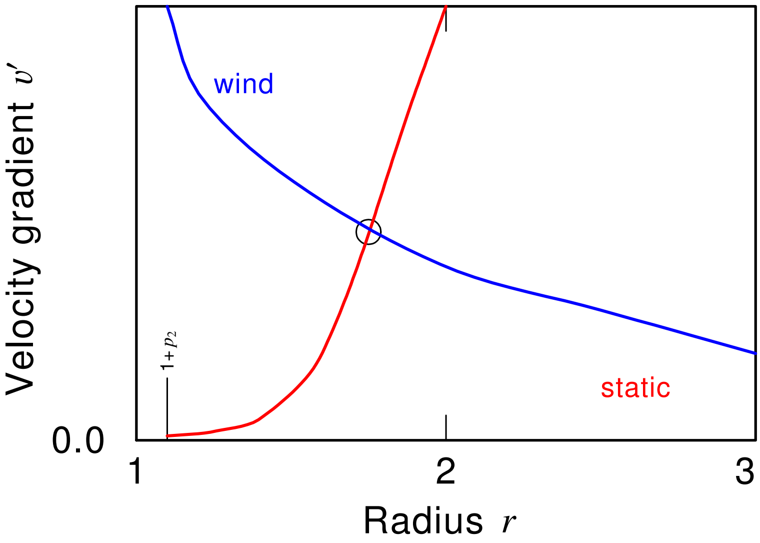


Figure 2: Sketch of the velocity gradient of the hydrostatic law and a beta-law with $\beta \leq 1$. In this case, there is certainly one intersection point (encircled).

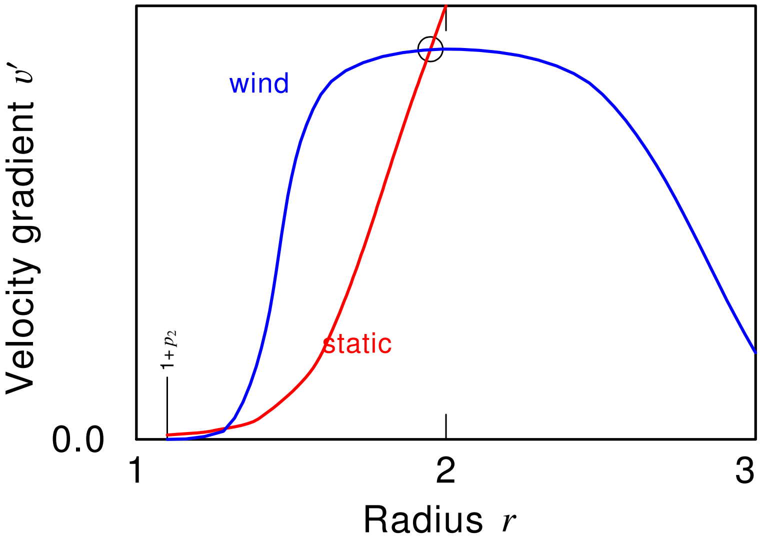


Figure 3: Sketch of the velocity gradient of the hydrostatic law and a beta-law with $\beta > 1$. We are searching for the second intersection point (encircled).

small steps of 0.1 hydrostatic scaleheights ($0.1H$), for the first value where $v'_{\rm wind} > v'_{\rm stat}$. For $\beta \leq 1$ this will be encountered immediately with the smallest value of r. For $\beta > 1$ this will be encountered after the first intersection point. By starting the search interval from here, we make sure that we will find only the second intersection point. If $v'_{\rm wind} > v'_{\rm stat}$ is never encountered below $r_{\rm gradmax}$, there are apparently no intersection points, and INITVEL stops with an error message – but this never happened so far. For finding the upper end of the search interval, we step further through r till $v'_{\rm wind} < v'_{\rm stat}$ is encountered. If this never happens till $r_{\rm max}$ is reached, the model has no wind domain.

All this is coded in SUBROUTINE INITVEL and SUBROUTINE VELTHIN.

11.3. The coarse frequency grid

A coarse frequency grid which is used in several cases where a frequency-dependent quantity has to be handled outside of the CMF radiative transfer program. This grid is defined at the start of a model in the FGRID routine. Various criteria are used to provide a sufficient frequency-space coverage. The frequency points are coming from two branches: The majority of points is automatically generated via several criteria described below. On top, additional points can be specified manually, most notably the blue-most frequency point.

11.3.1. User-defined frequency points

The FGRID file can be used to define frequency points manually. Each non-comment line must contain only one number, namely the wavelength of the frequency in Angstroem as a float number. Comment lines must be marked by an asterisk (*) as the first character. The FGRID file should contain at least one wavelength to define the blue-most point of the frequency grid.

On top of defining frequency points, there is one special line which can be given in the FGRID file. Via the line TREF = $x.x$ you can specify a reference temperature $T_{\rm ref}$ which will be used during the automatic frequency point generation. If this line is not given - which is the standard case - $T_{\rm ref} = T_*$ will be used.

Alternatively, the CARDS option OLD FGRID can be used to take all user-defined frequencies - i.e. those which are not generated by one of the automatically used criteria - from an old model. These points can be identified from the frequency-listing in wrstart.out as they are the only ones which do not have a keyword in the KEY column. The OLD FGRID option extracts only these frequency points from an old model for using them in the new one. If OLD FGRID is set, any current FGRID file is ignored.

The reading of both the FGRID file and the readout from an old model via OLD FGRID are performed in the DECFREQ subroutine.

11.3.2. Criteria-based frequency points

The majority of coarse-grid frequency points are set by automatic criteria, described in the following paragraphs. The order of the paragraphs reflects the order of insertion in the FGRID routine. The manually defined frequency points are read first, then the automatic criteria are used.

11.3.2.1. Red-most line frequency point To ensure that the frequency grid extends to a wavelength regime large enough to cover the red-most line frequency, the first automatically inserted point is calculated from this criterion. A loop over all line transitions (including iron) is performed to determine the transition with the largest wavelength $\lambda_{\rm redmost}$. To include also the line wings, the true redmost wavelength $\lambda_{\rm max}$ is then calculated by

$$\lambda_{\rm max} = \lambda_{\rm redmost} \cdot \left(1. + \frac{v_\infty}{10^6\,{\rm km/s}}\right). \qquad (101)$$

Regardless of the actual calculated value, it is ensured that $\lambda_{\max}$ is never lower than $10\,\mu$m. The keyword in the coarse-grid listing (KEY vector in the MODEL file) depends on some details. If the limit had to be raised to $10\,\mu$m, the point is marked as *REDMIN*. Otherwise it is indicated by the transition index (IND), preceeded by either *LINE* or *RUD*, reflecting whether the Einstein coefficient for the transition is known or not.

11.3.2.2. Insertion of continuum edges

To cover all photoionization edges, frequency points are inserted on both sides close to the edges. If the edge wavelength is given by λ_{KON}, the inserted two wavelengths $\lambda_{\pm}$ are

$$\lambda_{\pm} = (1 \pm w) \cdot \lambda_{\text{KON}} \tag{102}$$

with

$$w := \frac{10}{3} \frac{v_{\text{D}}}{c}. \tag{103}$$

Before an update in 2015, w was not coupled to the Doppler velocity v_{D}, but instead fixed to 0.0001.

Frequency points inserted due to continuum edges are marked with *EDGE-* or *EDGE+* followed by the corresponding element and the ionization stage. For K-shell ionization edges, the same procedure is performed after the “normal” continuum transitions. Instead of *EDGE*, here *K-ED* is used as the keyword indicator.

11.3.2.3. Small relative contributions

The so far existing frequency grid usually covers already a wide span of wavelengths. However, it is so far not guaranteed that their spacing is sufficient for all purposes where it is used.

For the next criterion the relative contribution of a particular interval to the Planck function is checked. For each wavelength interval $\lambda_k \ldots \lambda_{k+1}$ the corresponding frequency difference $\Delta\nu$ is calculated. Then the relative contributions r_B to the Planck function are calculated via

$$r_B = \frac{1}{2} \Delta\nu \frac{B_\nu(\lambda_k, T) + B_\nu(\lambda_{k+1}, T)}{B(T)} \tag{104}$$

for $T = T_{\text{ref}}$ and $T = 2 * T_{\text{ref}}$. If r_B is larger than 2.5% for any of the cases, an additional frequency point is inserted at

$$\lambda_{\text{add}} = \frac{1}{2} (\lambda_k + \lambda_{k+1}) \,. \tag{105}$$

The keyword for frequencies added due to large relative contributions is *ADD*. The whole relative contributions check is started again and again as long as wavelengths have to be inserted due to this criterion.

11.3.2.4. Small relative wavelength steps

The final insertion criterion checks the step sized of the so far existing grid. An additional point is inserted between λ_k and λ_{k+1} at λ_{mid} if

$$\frac{\lambda_{k+1} - \lambda_k}{\lambda_{\text{mid}}} > \epsilon \qquad \text{with} \qquad \lambda_{\text{mid}} := \frac{1}{2} (\lambda_k + \lambda_{k+1}) \,. \tag{106}$$

The value of ϵ depends strongly on the wavelength regime:

$$\lambda < 20\,\text{Å} \qquad \epsilon = 0.1 \tag{107}$$

$$20\,\text{Å} < \lambda < 227.83774\,\text{Å} \qquad \epsilon = 0.1\, f_2^{-1} \tag{108}$$

$$227.83774\,\text{Å} \leq \lambda \leq 300\,\text{Å} \qquad \epsilon = 0.1\, f_2^{-1}\, f_3^{-1} \tag{109}$$

$$300,\text{Å} < \lambda < 504.259\,\text{Å} \qquad \epsilon = 0.1\, f_3^{-1} \tag{110}$$

$$504.259\,\text{Å} \leq \lambda < 2000\,\text{Å} \qquad \epsilon = 0.1 \tag{111}$$

$$2000\,\text{Å} \leq \lambda < 15\,\mu\text{m} \qquad \epsilon = 0.01 \tag{112}$$

$$\lambda \geq 15\,\mu\text{m} \qquad \epsilon = 1.0 \tag{113}$$

The special factors f_2 and f_3 are calculated via

$$f_2 = \left(\frac{\lambda - 20\,\text{Å}}{126.1166\,\text{Å}}\right)^8 + 1 \tag{114}$$

$$f_3 = \left(\frac{\lambda - 300\,\text{Å}}{126.1166\,\text{Å}}\right)^8 + 10 \tag{115}$$

All frequencies added due to the wavelength step criterion are marked with the *ADD_DL* keyword.

11.3.3. Removal of extremely small grid steps

After passing all frequency insertion criteria, some points might be spaced extremely closed to one another. This can cause numerical issues in some of the calculations and thus a point at λ_k is removed from the coarse grid, if

$$\lambda_k < \lambda_{k-1} \cdot \left(1. + 5 \cdot 10^{-5}\right). \tag{116}$$

This usually affects only points inserted by the edge covering criterion. In some cases elements provide several transitions with edges very close to one another, e.g. neon, or some ionization edges of one element are close to those of another one already considered in the model. The removal of extremely small steps thus happens more often if more elements are used in a model.

11.3.4. Calculation of integration weights

As a last step of the coarse-frequency grid setup, a vector integration weights is calculated which is afterwards used for the discretization of dν in all frequency integrals over the coarse grid. The weights $w_{\nu,k}$ are calculated via trapezoidal rule, i.e.

$$w_{\nu,k} = \frac{c}{2}\left(\frac{1}{\lambda_{k+1}} - \frac{1}{\lambda_{k+1}}\right). \tag{117}$$

The first and the last weight have to be calculated with one-sided step, namely:

$$w_{\nu,1} = \frac{c}{2}\left(\frac{1}{\lambda_1} - \frac{1}{\lambda_2}\right) \tag{118}$$

$$w_{\nu,\text{NF}} = \frac{c}{2}\left(\frac{1}{\lambda_{\text{NF}-1}} - \frac{1}{\lambda_{\text{NF}}}\right) \tag{119}$$

Finally the newly calculated integration weights are renormalized to retain the exact integral sum of Planck's function using the reference temperature T_ref:

$$w_{k,\mathrm{norm}} = w_k \frac{B(T_\mathrm{ref})}{\sum_{l=1}^{\mathrm{NF}} w_l B_\nu(\lambda_l, T_\mathrm{ref})} \tag{120}$$

The normalization factor is usually very close to unity with differences on the order of 10^{-5} or lower.

12. Radiative transfer in the co-moving frame

12.1. “Ray-by-ray”: the angle-dependent transfer equation

Note that the statistical equations contain radiative rates of the form

$$R_{lu} = \int \frac{4\pi}{h\nu} \sigma_{lu}(\nu)\, J_\nu \, \mathrm{d}\nu \tag{121}$$

Here J_ν is the *angle-averaged radiation field*

$$J_{\nu_\mathrm{cmf}} = \frac{1}{2} \int_{-1}^{1} I_{\nu_\mathrm{cmf}}(\mu)\, \mathrm{d}\mu \tag{122}$$

Obviously, this angle integral must be taken in the co-moving frame (CMF) of the considered fluid element, i.e. the frequencies in the above equation are co-moving frame frequencies. This is one of the reasons that the whole radiative transfer is calculated in the CMF. This means:

- frequencies ν_cmf are measured in the local, co-moving frame-of-reference (CMF);
- all points are differentially moving (receding);
- propagating photons continuously change their frequency.

The transfer equation becomes a *partial* differential equation for the intensity

$$\pm \frac{\partial I_\nu^\pm}{\partial z} - \frac{\nu}{c} \frac{\mathrm{d}(\mu v)}{\mathrm{d}z} \frac{\partial I_\nu^\pm}{\partial \nu} = \eta_\nu - \kappa\, I_\nu^\pm \tag{123}$$

where the different signs holds for rays in $+z$ and $-z$-direction, respectively.

Now we introduce a dimensionless frequency x in Doppler units referring to a reference velocity v_D as

$$\nu = \nu_0 \left(1 + \frac{v_\mathrm{D}}{c}\right)^x \tag{124}$$

or, equivalently,

$$\ln \nu = \ln \nu_0 \ + \ln(1 + \frac{v_\mathrm{D}}{c})\ x\,. \tag{125}$$

Since $v_\mathrm{D} \ll c$, we have $\ln(1 + \frac{v_\mathrm{D}}{c}) \approx \frac{v_\mathrm{D}}{c}$. Therefore,

$$\mathrm{d}(\ln \nu) = \frac{v_\mathrm{D}}{c}\, \mathrm{d}x \quad \text{or} \quad \mathrm{d}\nu = \nu \frac{v_\mathrm{D}}{c}\, \mathrm{d}x\,. \tag{126}$$

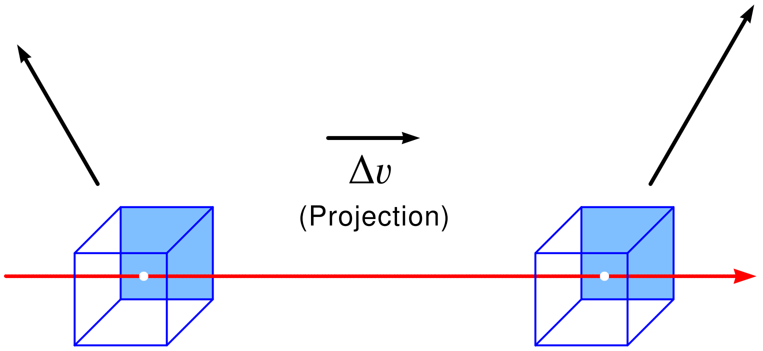


Figure 4: In a differentially expanding atmosphere, any two volume elements are moving with different velocities. A photon travelling from one to the other volume element experieces a Doppler shift according to the veloctity difference, projected on the ray’s direction.

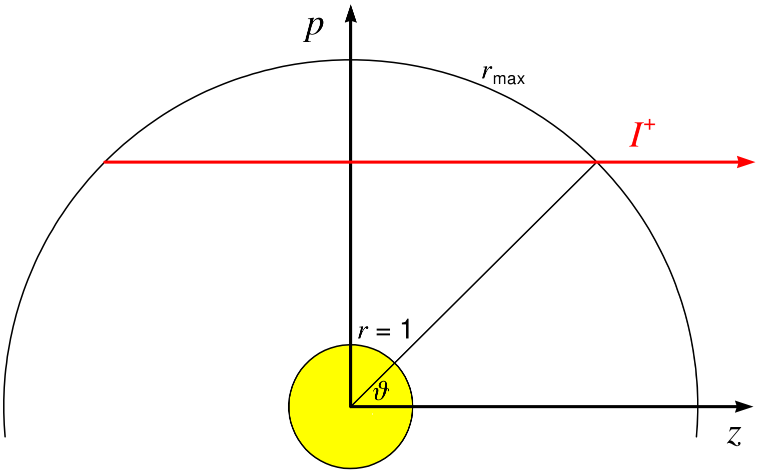


Figure 5: Coordinates are either spherical, with radius r and angle ϑ, or cartesian with impact parameter p and z in the direction to the observer. The atmosphere is calculated between the inner boundary at $r = 1$ and the outer boundary at $r = R_{\max}$.

The dimensionless velocity V is measured in the same Doppler units,

$$V(r) = v(r)/v_{\mathrm{D}} \tag{127}$$

With these dimensionless units, the transfer equation Eq. (123) becomes

$$\pm \frac{\partial I_\nu^\pm}{\partial z} - P(p,z)\frac{\partial I_\nu^\pm}{\partial x} = \eta_\nu - \kappa\, I_\nu^\pm \tag{128}$$

$P(p,z)$ is the projected velocity gradient in dimensionless units,

$$P(p,z) = \frac{\mathrm{d}(\mu V)}{\mathrm{d}z} = \mu\,\frac{\mathrm{d}V}{\mathrm{d}z} + V\,\frac{\mathrm{d}\mu}{\mathrm{d}z} = \mu^2\,\frac{\mathrm{d}V}{\mathrm{d}r} + (1-\mu^2)\,\frac{V}{r} \tag{129}$$

In principle, we can solve this radiative-transfer Eq. (128) for all frequencies and impact parameters, and then perform the angle-averaging and evaluate the radiativ rates. However, note that we need about 200 000 frequency points, 60 impact-parameters and 50 radial points, i.e. the total number of intensities to be calculated in each iteration is about 10^9.

Another problem with the ray-by-ray solution is the Thomson-scattering contribution to the emissivity. It contains scattered photons from all directions, and thus must be taken from the previous iteration when solving the radiative transfer ray-by-ray.

For these two reasons, it is more efficient to solve the *moment equations*; they have one dimension less, and the Thomson term cancels out (see Sect. 12.5).

Nevertheless, we will see below that the Moment Equations need supply with so-called Eddington factors, and these can only be obtained from solving the angle-dependent transfer Eq. (128), which will be described in the following subsection.

12.2. Short-characteristic integration

SUBROUTINE SHORTCHAR

As we will see below, we need a method which assures that the solution, i.e. the radiation field, is always strictly positive irrespective of any numerical errors.

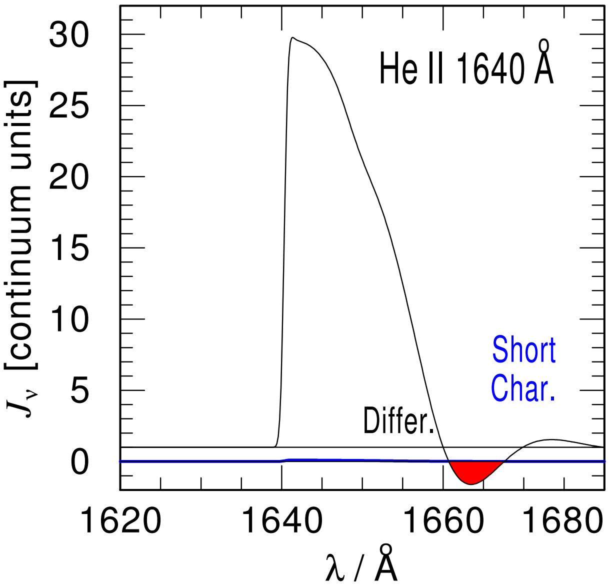


Figure 6: Angle-averaged intensity profile obtained with the short-characteristic intergration, compared to the result from the differencing scheme

The standard method which we tried first is a differencing scheme. According to Feautrier's method, one introduces the *Feautrier intensity* u and flux v as

$$u := \frac{1}{2}(I^+ + I^-) \qquad v := \frac{1}{2}(I^+ - I^-) \tag{130}$$

In the static case, a second-order equation for u is easily obtained, for which one can write a tridiagonal differencing scheme. The expanding case is somewhat more tricky, since v can only be eliminated from the equation *after* differencing. This method will not be described here further.

Practically, we encountered that the radiation intensity – even after angle-averaging – often became negative at some frequencies and depth points. An example is shown in Fig. 6. This is a fatal problem for the subsequent use of the moment equations, as the Eddington factor f becomes singular (see Sect. 12.4). Therefore we introduced a solution of Eq. (128) by integration along short characteristics (Koesterke, Hamann & Gräfener 2002).

The characteristics of the partial differential equation Eq. 128 are formally obtained by integration of the differential equation

$$\mathrm{d}x = \mp P(p, z)\,\mathrm{d}z\,. \tag{131}$$

They simply describe the redshift of the traveling photons in their CMF frequency. The run of the characteristics is sketched in Fig. 7). As $P(p, z) > 0$ for a monotinically expanding atmosphere, the characteristics always go from blue to red frequencies.

A *partial* differential equation (PDE) becomes an *ordinary* differential equation (ODE) along the characteristic. For the coordinate along the characteristic we just take the z coordinate. Then the ODE is just the usual transfer equation, i.e. we have transformed the CMF equation back into the observer's frame.

The concept of the *short characteristic* method is now, that the characteristics are only intergated across one grid cell. The incoming radiation provides the boundary condition at the walls of the cell and is obtained by interpolation between the grid points.

Details of our models can be seen from the sketch in Fig. 8. Assume that we need to calculate $I^+_{l,k}$, i.e. the intensity in $+z$ direction at spatial point z_l and frequency x_k.

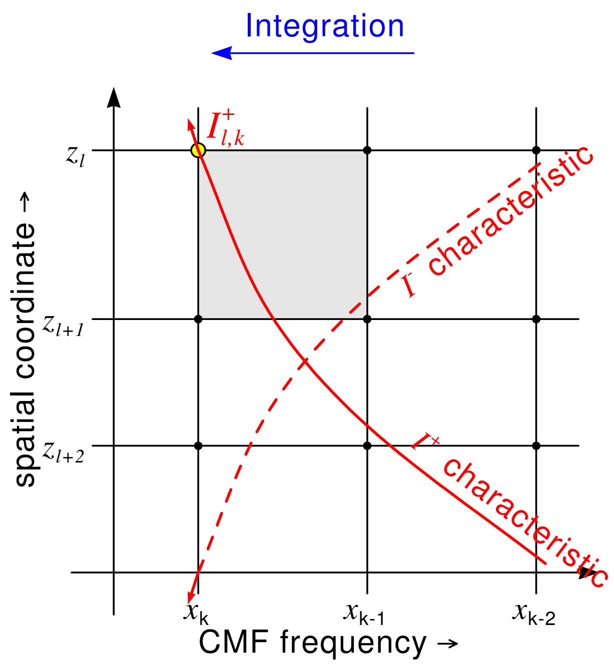


Figure 7: Characteristics of the CMF transfer equation for a monotonic velocity field

Integration the transfer equation yields the well-known solution

$$I^+_{l,k} = \int_{\tau=0}^{\hat{\tau}} S(\tau)\, e^{-\tau} d\tau \;+\; \hat{I}^+ e^{-\hat{\tau}}\,. \tag{132}$$

The variables with “hat” refer to the boundary where the short characteristic enters the cell.

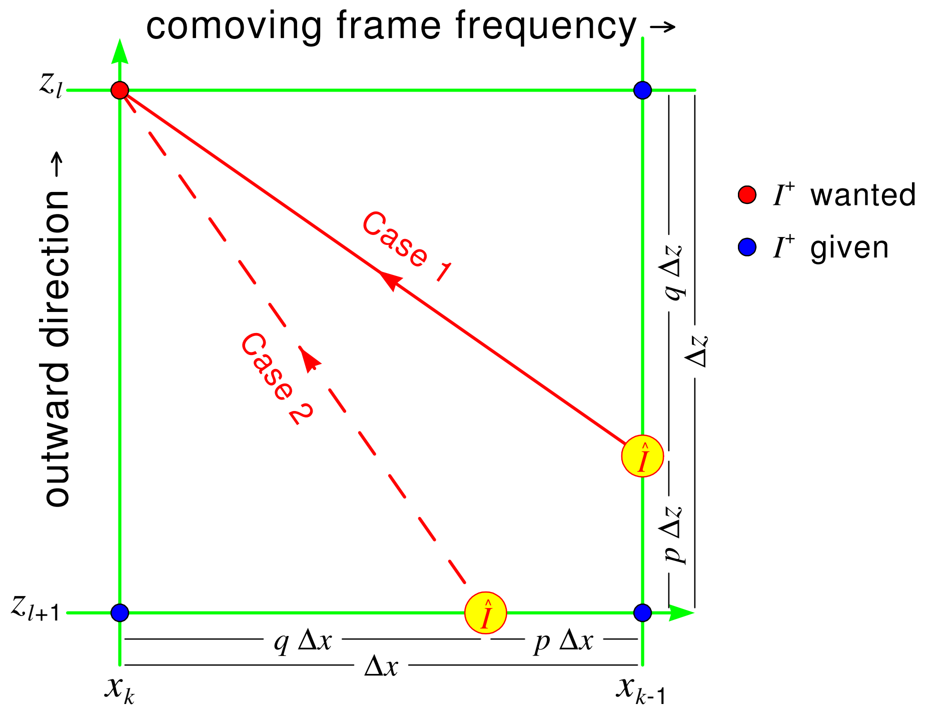


Figure 8: Short characteristics in one cell

We assume that the short characteristic is linear within the cell. Its slope is given by $-P_{l,k}^{-1}$. The point where short characteristic enters the cell is termed the "hat" point, and all quantities referring to that point are notated with a "hat". We must now distinguish whether the "hat" point lies on the frequency boundary (*case 1*) or the spatial boundary (*case 2*) of the cell. This follows from:

$$\text{if } \; P_{l,k}\, \Delta z > \Delta x \quad \text{than} \quad \textit{case}\, 1 \quad \text{else} \quad \textit{case}\, 2 \tag{133}$$

The "hat" point divides its cell wall in fractions p and q as shown in the sketch ($p + q = 1$). In our standard version, we simply assume that the characteristics are linear with the slope $-P_{l,k}^{-1}$. Then we define

$$\text{PPDZ} = \frac{\Delta z\, P_{l,k}}{\Delta x} \tag{134}$$

and obtain

$$\text{for case 1: } \; q = \text{PPDZ}^{-1} \quad \text{and for case 2: } \; q = \text{PPDZ} \tag{135}$$

Alternatively, one may find the "hat" point as the intersection point of the characteristic with the cell boundaries from the corresponding conditions and the (linearly interpolated) velocity field. This version has also been coded (`SUBROUTINE SHORTRAY` in libcr_shortray-test), but I cannot remember the outcome of the tests.

Opacity and source function at the "hat" point are obtained by linear interpolation (see Table 12.2). The optical depth τ between the corner point (x_k, z_l) and the "hat" point is calculated with the trapeziodal rule.

For the interpolation of I, a higher order than linear turned out to be essential for the case 1. However, the interpolation must be *monotonic*, meaning that the interpolated value is never outside the range between the two values at the cell corners. Only this guaranties that the resulting intensitieds are always positive, as long as the opacities and the source function are positive. See Appendix C for details of the spline-interpolation formalism.

In the frequency coordinate linear interpolation is sufficient (case 2). This reflects the fact that the spatial resolution of the grid is much coarser than the frequency resolution.

Table 1: Short-characteristic integration: calculation of the "hat" quantities at the cell boundary

Case 1	Case 2
$\hat{\kappa} = p\, \kappa_{l,k-1} + q\, \kappa_{l+1,k-1}$	$\hat{\kappa} = p\, \kappa_{l+1,k} + q\, \kappa_{l+1,k-1}$
$\hat{S} = p\, S_{l,k-1} + q\, S_{l+1,k-1}$	$\hat{S} = p\, S_{l+1,k} + q\, S_{l+1,k-1}$
$\hat{\tau} = \frac{1}{2}(\kappa_{l,k} + \hat{\kappa}) q\, \Delta z$	$\hat{\tau} = \frac{1}{2}(\kappa_{l,k} + \hat{\kappa})\, \Delta z$
$\hat{I}$ = spline interpol. along x_{k-1}	$\hat{I} = p\, I_{l+1,k} + q\, I_{l+1,k-1}$

Having prepared τ and $\hat{S}$, the integration of the transfer equation Eq. (132) is performed as a quadrature sum with only two points,

$$I_{l,k}^{+} = w_0 S_{l,k} \; + \; \hat{w}\hat{S} \; + \; \hat{I}^{+} e^{-\hat{\tau}} \tag{136}$$

The quadrature weights w_0 and $\hat{w}$ incorporate the kernel function of the integral, $e^{-\tau}$,

$$w_0 = 1 - \frac{1 - e^{-\hat{\tau}}}{\hat{\tau}} \qquad \hat{w} = 1 \; - \; w_0 \; - \; e^{-\hat{\tau}} \tag{137}$$

See Appendix B.6 for the construction of these quadrature weights.

The short-characteristic integration for an inward ray (I^{-}) is completely in analogy. Note that the short-characteristic integration of $I_{l,k}$ requires that the intensities at the other three corners of the cell are already known. We start at the bluemost frequency x_1, where the intensity must be specified as *blue-wing boundary condition*. At the next frequency, the loop stars at the outermost cell with the inward ray. I^{-} is specified by the outer boundary condition is space (see below). Calculating cell by cell, we arrive either at the stellar core, or at the plane of symmetry. In the first case, we start with the integration of I^+ with $I^+_{\rm core}$ as boundary condition in space. In the latter case, the boundary value of I^+ is given by I^- at the symmetry plane from the inward integration. The integration of I^+ proceeds cell by cell outwards until the outer boundary is reached.

Boundary conditions: outer boundary

The standard assumption would be $I_\nu^- = 0$ at the outer boundary (the subscript ν is omitted for brevity throughout this section). However, strong lines and continua are often not yet optically thin at R (in the following short for $R_{\rm max}$). Assuming $I^- = 0$ at the boundary then produces quite a discontinuity in the radiation field, leading to strong gradients in the population numbers which can hamper the convergence. Therefore it is desirable to choose I^- at R such that steep gradients are avoided. We did not find yet an ideal solution to this problem. The PoWR code provides different versions for the outer boundary condition which can be selected by the `OB_VERS` option (cf. Sect. **??**).

Since the region outside R is not calculated, approximations can only be based on extrapolations. For the opacity between R and infinity we assume that it dilutes from its value at the boundary (κ_1) with r^{-2} like the density (for constant υ). The radial optical depth between R and r then becomes

$$\tau(r) = \int_R^r \kappa_1 \left(\frac{R}{r'}\right)^2 \,\mathrm{d}r' = \kappa_1\, R^2 \left(\frac{1}{R} - \frac{1}{r}\right) \tag{138}$$

The radial optical depth from the boundary at radius R to infinity is

$$\tau_{\rm B} = \kappa_1\, R \tag{139}$$

Note that with this optical depth any velocity gradients are neglected, i.e. it holds if the velocity outside of R stays constant. Moreover, we will take the radial optical depth also for the non-radial rays (which slightly under-estimates their optical depth), making I^- independent of the impact parameter.

Different assumptions can be chosen for the source function S_ν:

In `OB-VERS 1` the source function stays constant at its value at the boundary, S_1. Then, $I^- = S_1\,(1 - e^{-\tau_{\rm R}})$

In `OB-VERS 2` it is assumed that the source function decreases with r^{-2}, which would hold if the emission scales with density-squared. Since $S = \eta/\kappa$, this implies that the emissivity scales as $\eta = \eta_1\,(R/r)^4$. The transfer equation can be integrated analytically:

$$\begin{aligned} I^- &= \int_{\rm R}^{\infty} S(r)\, e^{-\tau(r)}\, \kappa(r)\,\mathrm{d}r \\ &= \eta_1\, R^4 \int_{\rm R}^{\infty} r^{-4}\, e^{-\kappa_1 R^2 (\frac{1}{R}-\frac{1}{r})}\,\mathrm{d}r \\ &= \eta_1\, R^4\, e^{-\kappa_1 R} \int_{\rm R}^{\infty} r^{-4}\, e^{\kappa_1 R^2/r} \mathrm{d}r \end{aligned} \tag{140}$$

Now we make a substitution: $x := \kappa_1\, R^2/r$, which implies $\mathrm{d}x = -\kappa_1\, R^2/r^2\,\mathrm{d}r$. The upper boundary of the integral ($r = \infty$) transforms to $x = 0$, and the lower boundary ($r = R$) to $x = \kappa_1 R = \tau_{\rm B}$. The boundaries are swapped to compensate for the minus sign. Thus,

$$\begin{aligned} I^- &= \frac{\eta_1}{\kappa_1}\, R^2 e^{-\kappa_1 R} \int_0^{\tau_{\rm B}} r^{-2}\, e^x\,\mathrm{d}x \\ &= S_1\, R^{-2}\, \kappa_1^{-2} e^{-\kappa_1 R} \int_0^{\tau_{\rm B}} x^2\, e^x\,\mathrm{d}x \\ &= S_1\, \tau_{\rm B}^{-2} e^{-\tau_{\rm B}} \int_0^{\tau_{\rm B}} x^2\, e^x\,\mathrm{d}x \end{aligned} \tag{141}$$

Bronstein (1981, p. 113) gives

$$\int x^2\, e^x\, \mathrm{d}x = e^x\,(x^2 - 2x + 2) \tag{142}$$

With the boundaries $[0, \tau_\mathrm{B}]$ this gives $e^{\tau_\mathrm{B}}(\tau_\mathrm{B}^2 - 2\tau_\mathrm{B} + 2) - 2$. Inserting this for the integral, the exponential terms cancel except for the lower-boundary term, and we finally obtain

$$\begin{aligned} I^- &= S_1 \left[\tau_\mathrm{B}^{-2}(\tau_\mathrm{B}^2 - 2\tau_\mathrm{B} + 2) - \tau_\mathrm{B}^{-2} e^{-\tau_\mathrm{B}} \cdot 2\right] \\ &= S_1 \left[1 - \frac{2}{\tau_\mathrm{B}} + \frac{2}{\tau_\mathrm{B}^2} - \frac{2e^{-\tau_\mathrm{B}}}{\tau_\mathrm{B}^2}\right] \\ &= S_1 \left[1 - \frac{2}{\tau_\mathrm{B}} + \frac{2}{\tau_\mathrm{B}^2}(1 - e^{-\tau_\mathrm{B}})\right] \end{aligned} \tag{143}$$

Two final modifications are applied to this results, for very thick and for very thin opti depths, respectively. For very large τ_B, the formalism yields $I^- \approx S_1$; Götz found that this can lead to a slow runaway of the boundary value and therefore introduced an upper limit as $I^- \le 0.9999\, S_1$.

For very small τ_B, Eq. (143) can produce a floating exception (division by zero). To avoid this, the exponential function can be expanded in a Taylor series: $e^{-\tau_\mathrm{B}} \approx 1 - \tau_\mathrm{B} + \frac{\tau_\mathrm{B}^2}{2} - \frac{\tau_\mathrm{B}^3}{6}$ (third order is needed!). The last line of Eq. (143) then becomes

$$\begin{aligned} I^- &= S_1 \left[1 - \frac{2}{\tau_\mathrm{B}} + \frac{2}{\tau_\mathrm{B}^2}\left(\tau_\mathrm{B} - \frac{\tau_\mathrm{B}^2}{2} + \frac{\tau_\mathrm{B}^3}{6}\right)\right] \\ &= S_1 \left[1 - \frac{2}{\tau_\mathrm{B}} + \left(\frac{2}{\tau_\mathrm{B}} - 1 + \frac{\tau_\mathrm{B}}{3}\right)\right] \\ &= S_1\, \frac{\tau_\mathrm{B}}{3} \end{aligned} \tag{144}$$

Summarizing, the outer boundary condition (version 2) is

$$I^- = \begin{cases} S_1\, \dfrac{\tau_\mathrm{B}}{3} & : \ \tau_\mathrm{B} < 10^{-3} \\ S_1 \min\left(\left[1 - \dfrac{2}{\tau_\mathrm{B}} + \dfrac{2}{\tau_\mathrm{B}^2} - \dfrac{2}{\tau_\mathrm{B}^2}\mathrm{e}^{-\tau_\mathrm{B}}\right], 0.9999\right) & : \ \tau_\mathrm{B} \ge 10^{-3} \end{cases}$$

`OB-VERS 4` is similar to `OB-VERS 2`, but instead of the source function S_1 at the boundary we employ a value S_B which is obtained by a complicate procedure (again, to avoid a too strong feedback which can result in a runaway), namely a least-square fit to the source function as function of radius. This polynomial fit is prepared in SUBROUTINE SFIT. For the outermost MBOUND (presently set to 30) radius points, the following vectors are prepared: $x_\ell = \log n_\ell^\mathrm{tot} - n_1^\mathrm{tot}$ (log of the total number density, in difference to its value at the boundary), $y_\ell = T_\mathrm{rad}(S_\ell)$ (radiation temperature of the source function), $w_\ell = x_{\ell+1} - x_{\ell-1}$ (weight according to the separation of the x points. Then SUBROUTINE POLYFIT performs a weighted least-square fit with a cubic function (other polynomial degrees are presently disabled; in the CRAY branch, the NAG subroutine E02ADF was used instead of POLYFIT which allowed for arbitrary polynomial degrees). The origin of the algorithm in POLYFIT is not documented (numerical recipes?). The proper functioning of this artwork can be checked with the help of a test-plot facility that can be activated in the code of SFIT.

The POLYFIT routine returns the coefficients for the fit polynomial

$$f(x) = a_4 + a_3 x + a_2 x^2 + a_1 x^3 \tag{145}$$

(note the falling numbering of the indices); since $x = 0$ for the boundary, the lowest coefficient a_4 gives the value at the boundary.

Sometimes, the fit polynom shows a minimum and then rises towards the boundary again. This also can lead to a slow runaway of the boundary value with the iteration. Thefore, we check the polynomial function for a possible minimum and, if existing, take the minimum value instead of the value at the boundary.

This needs some more algebra. The first two derivatives of the polynomial function are

$$f'(x) = a_3 + 2a_2 x + 3a_1 x^2 \tag{146}$$

and

$$f''(x) = 2a_2 + 6a_1 x \tag{147}$$

An extrema requires the first derivatiove to vanish. Thus we solve the quadratic equation

$$x^2 + \frac{2a_2}{3a_1} x + \frac{a_3}{3a_1} = 0 \tag{148}$$

which has the standard form $x^2 + px + q = 0$. First we check if real solutions exist, i.e. if $(\frac{p}{2})^2 > q$. If so, we calculate the solutions

$$x_{1,2} = -\frac{p}{2} \pm \sqrt{\frac{p}{2})^2 - q} \tag{149}$$

and check each of them if they lie in the consired range of x-values $(x_1, x_{\mathrm{MBOUND}})$. If so, we evaluate the second derivative and check for a minimum, i.e. if $f'' > 0$. If so, we evaluate the polynomial at this x_1 or x_2, respectively, and convert this radiation temperature back into an intensity which serves then as the boundary value S_{B} instead of S_1 in Eq. (143). Uff!

`OB-VERS 5` is analog to version 4, but works only with the continuum opacities.

Boundary conditions: inner boundary

For the incident radiation I^+ at the inner boundary ($r = 1$) we assume LTE and the diffusion approximation. This means,

$$I^+_\nu(\mu) = B_\nu + \mu \frac{\mathrm{d}B_\nu}{\mathrm{d}\tau} = B_\nu - \mu \left. \frac{\mathrm{d}B_\nu}{\mathrm{d}r} \right/ \kappa \tag{150}$$

While this is also done in `SUBROUTINE SHORTCHAR`, the terms B_ν and $\frac{\mathrm{d}B_\nu}{\mathrm{d}r}$ are prepared in `SUBROUTINE CLDIFFUS`. B_ν is just the Planck function for T_{ND}. $\frac{\mathrm{d}B_\nu}{\mathrm{d}r}$ is written with the chain rule as $\frac{\mathrm{d}B_\nu}{\mathrm{d}T}\frac{\mathrm{d}T}{\mathrm{d}r}$. The analytic expression for $\frac{\mathrm{d}B_\nu}{\mathrm{d}T}$ is coded in the `FUNCTION DBNUEDT`.

$\frac{\mathrm{d}T}{\mathrm{d}r}$ is read from the model file and has been prepared in the `STEAL` program with the details being described further down.

The flux at the inner boundary $H_{\nu,\mathrm{ND}}$ is not necessarily identical to the diffusion flux

$$H_{\nu,\mathrm{diff}} = -\frac{1}{3\kappa_\nu}\frac{\partial B_\nu}{\partial r} = \frac{1}{3\kappa_\nu}\frac{\partial B_\nu}{\partial T}\left|\frac{\mathrm{d}T}{\mathrm{d}r}\right|_{r=R_*} \tag{151}$$

since this would require that also I^-_ν could be described by the diffusion approximation (150). A more precise calculation of the inner boundary flux which can account for deviations from the diffusion approximation is instead obtained by the following procedure. First, we introduce a special quantity

$$H_{\nu,\mathrm{spec}} := \frac{1}{2}\int\limits_0^1 (I^+_\nu + I^-_\nu)\,\mu \mathrm{d}\mu. \tag{152}$$

Note that this quantiy has a flux-like integral weigth, but an intensity-like core. This quantity is calculated in the `SUBROUTINE SHORTCHAR`, together with the regular quantity $J_{\nu,\mathrm{ND}}$. This allows the calculation of a special Eddington factor

$$h_{\nu,\mathrm{in}} := \frac{H_{\nu,\mathrm{spec}}}{J_{\nu,\mathrm{ray},1}} = \frac{\int_0^1 (I_\nu^+ + I_\nu^-)\,\mu\mathrm{d}\mu}{\int_0^1 (I_\nu^+ + I_\nu^-)\,\mathrm{d}\mu}. \tag{153}$$

The such defined $h_{\nu,\mathrm{in}}$ can then be used to remove I_ν^- from the calculation of the total flux at the inner boundary. Summing $H_{\nu,\mathrm{spec}}$ and the ordinary defintion of $H_{\nu,\mathrm{ND}}$ yields:

$$H_{\nu,\mathrm{ND}} + H_{\nu,\mathrm{spec}} = \int_0^1 I_\nu^+ \mu\mathrm{d}\mu \tag{154}$$

$$H_{\nu,\mathrm{ND}} + h_{\nu,\mathrm{in}} J_{\nu,\mathrm{ND}} = \frac{1}{2}B_\nu - \frac{1}{3\kappa_\nu}\frac{\partial B_\nu}{\partial r} \tag{155}$$

$$\Rightarrow H_{\nu,\mathrm{ND}} = H_{\nu,\mathrm{diff}} + \frac{1}{2}B_\nu - h_{\nu,\mathrm{in}} J_{\nu,\mathrm{ND}} \tag{156}$$

The correct flux at the inner boundary $H_{\nu,\mathrm{ND}}$ is thus obtained via Eq. (156), which illustrates that the pure diffusion term has to be corrected for deviations from $J_\nu = B_\nu$.

In the `SUBROUTINE FREQUINT`, the total flux at the inner boundary and the flux resulting only from the correction terms are integrated over the whole frequency range, yielding the two following quantities:

$$H_{\mathrm{ND}} = \int_0^\infty \left(H_{\nu,\mathrm{diff}} + \frac{1}{2}B_\nu - h_{\nu,\mathrm{in}} J_{\nu,\mathrm{ND}}\right) \mathrm{d}\nu \tag{157}$$

$$H_{\mathrm{ND,cor}} = \int_0^\infty \left(\frac{1}{2}B_\nu - h_{\nu,\mathrm{in}} J_{\nu,\mathrm{ND}}\right) \mathrm{d}\nu \tag{158}$$

Usually $H_{\mathrm{ND,cor}}$ should only be a small fraction of H_{ND}.

The diffusion term $H_{\nu,\mathrm{diff}}$ requires the temperature gradient at the inner boundary (see Eq. 151), which is either given as a precalculated quantity from the STEAL `SUBROUTINE TEMPCORR`, or – if the `TDIFFUS` option is set in the CARDS file – from assuming that the proper gradient should arise from pure diffusion, i.e. $H_{\mathrm{diff}} = \int H_{\nu,\mathrm{diff}}\,\mathrm{d}\nu \overset{!}{=} H_0$. The latter is done in the `SUBROUTINE DIFDTDR` and leads to the following simple formula which only requires the Rosseland optical depth and the current electron temperature T_{ND} at the inner boundary to be given:

$$\int_0^\infty H_{\nu,\mathrm{diff}}\,\mathrm{d}\nu = H_0 \tag{159}$$

$$\int_0^\infty \frac{1}{3\kappa_{\nu,\mathrm{ND}}} \frac{\partial B_\nu}{\partial T} \left|\frac{\mathrm{d}T}{\mathrm{d}r}\right|_{r=R_*} \mathrm{d}\nu = \frac{\sigma_{\mathrm{SB}}}{4\pi} T_*^4 \tag{160}$$

$$\frac{1}{3\kappa_{\mathrm{Ross,ND}}} \frac{\partial B}{\partial T} \left|\frac{\mathrm{d}T}{\mathrm{d}r}\right|_{r=R_*} = \frac{\sigma_{\mathrm{SB}}}{4\pi} T_*^4 \tag{161}$$

$$\frac{4}{3\kappa_{\mathrm{Ross,ND}}} \frac{\sigma_{\mathrm{SB}}}{\pi} T_{\mathrm{ND}}^3 \left|\frac{\mathrm{d}T}{\mathrm{d}r}\right|_{r=R_*} = \frac{\sigma_{\mathrm{SB}}}{4\pi} T_*^4 \tag{162}$$

$$\left|\frac{\mathrm{d}T}{\mathrm{d}r}\right|_{r=R_*} = \frac{3}{16} \kappa_{\mathrm{Ross,ND}} \frac{T_*^4}{T_{\mathrm{ND}}^3} \tag{163}$$

The `TDIFFUS` option further implies that the innermost temperature point is calculated by enforcing the gradient from Eq. (163) between the innermost two radius points. This is performed in the STEAL `SUBROUTINE TDIFFUS`.

Without the `TDIFFUS` option, the innermost temperature is calculated similar to all other ones, i.e. from the Usöld-Lucy method performed in the `SUBROUTINE TEMPCORR`. In the same routine, a "special" temperature gradient at the inner boundary is calculated, which is then used for obtaining $H_{\nu,\mathrm{diff}}$ in the COLI `SUBROUTINE CLDIFFUS`. This gradient is calculated from the last electron temperature gradient, corrected with a factor

$$f_{\mathrm{fluxcor}} = \frac{\sigma_{\mathrm{SB}}}{4\pi} \frac{T_*^4}{H_{\mathrm{ND}}} \tag{164}$$

which ensures the correct flux at the inner boundary. The gradient is then obtained as

$$\left|\frac{\mathrm{d}T}{\mathrm{d}r}\right|_{r=R_*} = 1 + (f_{\mathrm{fluxcor}} - 1) \cdot 0.1 \cdot d_{\mathrm{unlu,int}} \tag{165}$$

with $d_{\mathrm{unlu,int}}$ being the damping factor for the integral term in the Usöld-Lucy correction. The additional damping of 0.1 is done to ensure that the temperature correction at the inner boundary is prioritized compared to the gradient correction.

Boundary conditions: frequency

As for our monotonically expanding atmosphere the (short) characteristics always proceeds towards longer wavelengths, the integration needs a boundary condition at the bluemost frequency. Since we start with our frequency grid at a very small wavelength, we can assume that the radiation field is zero:

$$I_{k,l}^- = 0 \;\text{ and }\; I_{k,l}^+ = 0 \;\text{ for }\; k = 1 \;\text{ and for all }\; l < \mathrm{ND} \tag{166}$$

This is achieved by initializing the vectors `XIMINUS_OLD` and `XIPLUS_OLD` to zero. In the loop over the frequency, these vectors are then always replaced by the newly calculated $I_{k,l}^-$ and $I_{k,l}^+$, before proceeding to the next frequency index k.

Note that the quadrature formula cannot assure positive I when the opacities become negative. The Koesterke et al. (2002) paper contains the remark: "Even in case of negative opacities (stimulated emission exceeds absorption) the integral remains positive because negative source functions meet negative τ-steps.

12.3. Integration of the moments

The ray-by-ray radiative transfer calculation described in the previous section proves the radiation intensities I^+ and I^- at each radius point, impact-parameter point and frequency point, i.e. a huge amount of data (gigabytes). However, for the subsequent calculation of the radiative transfer with moment equations, described in the subsequent section' only the *Eddington factors* (i.e. rations between the *moments* of the radiation field, are actually needed.

The moments of the radiation intensity are defined as

$$[\tilde{J}, \tilde{H}, \tilde{K}, \tilde{N}] \;=\; \frac{1}{2}\, r^2 \int_{-1}^{1} I(\mu)[1, \mu, \mu^2, \mu^3]\, d\mu \tag{167}$$

The r^2-factor implied in the tilded quantities in convenient for the spherical geometry; the same definition applies for the source function $\tilde{S}_\nu = r^2 S_\nu$.

Thus the moments are integrals of the intensities. Therefore, each intensity that is calculated in the ray-by-ray radiative transfer can be immediately added to the intergrals (i.e. quadrature sums with the appropriate integration weights), and must not be stored any longer.

The integrals .... quadrature weights ... `SUBROUTINE COLIWM` ... TO BE WRITTEN!!

12.4. Moment equations

We start with the transfer equation in spherical coordinates (r, ϑ). Compared to the transfer Eq. (refeq:cmf-ray) in cartesion coordinates (p, z), the spatial derivative now splits into two terms since along a ray not only r is changing, but also its angle ϑ against the radius vector. Therefore the equation now also gains a term with partial derivative with respect to μ:

$$\begin{aligned} & \mu \frac{\partial I(x,\mu,r)}{\partial r} + \frac{1-\mu^2}{r}\frac{\partial I(x,\mu,r)}{\partial \mu} + \\ & \left(-\mu^2\, \frac{\mathrm{d}V(r)}{\mathrm{d}r} - (1-\mu^2)\frac{V(r)}{r}\right)\frac{\partial I(x,\mu,r)}{\partial x} \;=\; \eta(x,r) \;-\; \kappa(x,r)\, I(x,\mu,r) \end{aligned} \tag{168}$$

The *Moment equations* are obtained by performing the angle integral over the angle-dependent transfer equation. For this step we need to perform partial integration for those terms containing two μ-dependent factors:

Auxiliary calculations (partial integrations):

(1) First term of Eq. (168), integration over dμ

$$\frac{1}{2}\int_{-1}^{+1} \mu \frac{\partial I}{\partial r} \mathrm{d}\mu = \frac{1}{2}\frac{\partial}{\partial r}\int_{-1}^{+1} \mu\, I\, \mathrm{d}\mu = \frac{\partial H}{\partial r}$$

Second term of Eq. (168), integration over dμ

$$\begin{aligned} \frac{1}{2}\int_{-1}^{+1} \frac{1-\mu^2}{r}\frac{\partial}{\partial \mu} I\, d\mu \;&=\; \frac{1}{2}\left[\frac{1-\mu^2}{r} I(\mu)\right]_{\mu=-1}^{+1} - \frac{1}{2}\int_{-1}^{+1} (-2\mu/r) I(\mu)\, d\mu \\ &=\; 0 + \frac{2H}{r} \end{aligned}$$

both term together:

$$\frac{\partial H}{\partial r} + \frac{2}{r}H = \frac{1}{r^2}\frac{\partial(r^2 H)}{\partial r} = \frac{1}{r^2}\frac{\partial(\tilde{H})}{\partial r}$$

(2) Second term of Eq. (168), integration over $\mu\,\mathrm{d}\mu$

$$\begin{aligned}\int \frac{1-\mu^2}{r}\frac{\partial}{\partial\mu} I\,\mu d\mu &= \frac{1}{2}\left[\frac{\mu-\mu^3}{r}I(\mu)\right]_{\mu=-1}^{+1} - \frac{1}{2}\int_{-1}^{+1}\frac{1-3\mu}{r}I(\mu)\,d\mu \\ &= 0 + \frac{1}{r}(3K - J)\end{aligned}$$

With the help of these auxiliary relations, we can integrate the above transfer Eq. (168) and obtain:

$$\mathbf{0.Moment}: \quad -\frac{\partial\tilde{H}}{\partial r} + \left(\frac{\mathrm{d}V(r)}{\mathrm{d}r} - \frac{V(r)}{r}\right)\frac{\partial\tilde{K}}{\partial x} + \frac{V(r)}{r}\frac{\partial\tilde{J}}{\partial x} = \left(\tilde{J}(r) - \tilde{S}(r)\right)\kappa(x,r) \tag{169}$$

$$\mathbf{1.Moment}: \quad \frac{\partial(q\tilde{K})}{-q\partial r} + \left(\frac{\mathrm{d}V(r)}{\mathrm{d}r} - \frac{V(r)}{r}\right)\frac{\partial\tilde{N}}{\partial x} + \frac{V(r)}{r}\frac{\partial\tilde{H}}{\partial x} = \kappa\,\tilde{H} \tag{170}$$

The first term of 170 had been simplified with help of the definition of an Eddington factor $f = K/J$ and the "sphericity factor" q by the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}r}\left(\ln(r^2 q)\right) = \frac{1}{r^2 q}\frac{\mathrm{d}(r^2 q)}{\mathrm{d}r} =: \frac{3f-1}{rf}$$

When chosing arbitrarily $q(r=1) = 1$ as initial condition, and defining $F(r) = (3f(r)-1)/f(r)$, integration over r' yields

$$\ln(r^2 q) = \int_1^r F(r')\frac{\mathrm{d}r'}{r'} \tag{171}$$

Incrementing the integral from radial index $l+1$ to index l means

$$\ln(r_l^2 q_l) = \ln(r_{l+1}^2 q_{l+1}) + \int_{r_{l+1}}^{r_l} F(r')\frac{\mathrm{d}r'}{r'} \tag{172}$$

Replacing the integral by a quadrature sum gives

$$\ln(r_l^2 q_l) = \ln(r_{l+1}^2 q_{l+1}) + w_{l+1}F_{l+1} + w_l F_l \tag{173}$$

where the weights include the inverse-r kernel function as given in the Appendix B.5, i.e.

$$w_{l+1} = \frac{r_l}{r_l - r_{l+1}}\ln\left|\frac{r_l}{r_{l+1}}\right| - 1 \quad , \quad w_l = -\frac{r_{l+1}}{r_l - r_{l+1}}\ln\left|\frac{r_l}{r_{l+1}}\right| + 1 \tag{174}$$

Re-odering the terms gives

$$\ln(r_l^2 q_l) = \ln(r_{l+1}^2 q_{l+1}) + (F_l - F_{l+1}) + F_{l+1}\ln\left(\frac{r_l}{r_{l+1}}\right)^{\frac{r_l}{r_l - r_{l+1}}} + F_l\ln\left(\frac{r_l}{r_{l+1}}\right)^{-\frac{r_{l+1}}{r_l - r_{l+1}}} \tag{175}$$

Now we can take the whole equation to the exponent,

$$r_l^2 q_l = r_{l+1}^2 q_{l+1}\; e^{F_l - F_{l+1}}\left(\frac{r_l}{r_{l+1}}\right)^{\frac{r_l}{r_l - r_{l+1}}F_{l+1}}\left(\frac{r_l}{r_{l+1}}\right)^{-\frac{r_{l+1}}{r_l - r_{l+1}}F_l} \tag{176}$$

As the two terms with exponents have the same basis, they can be assembled to

$$r_l^2 q_l = r_{l+1}^2 q_{l+1}\, e^{F_l - F_{l+1}} \left(\frac{r_l}{r_{l+1}}\right)^{\frac{r_l F_{l+1} - r_{l+1} F_l}{r_l - r_{l+1}}} \tag{177}$$

This formula is now easily coded in `SUBROUTINE COLIMO`:

```
C***  Sphericity factor
      QLF(ND) = 1.
      RRQ = 1.
      RL = RADIUS(ND)
      FL = 3. - 1./EDDIF(ND)
      DO L=ND-1, 1, -1
        RLP = RL
        RL = RADIUS(L)
        FLP = FL
        FL = 3. - 1./EDDIF(L)
        RRQ = RRQ * EXP(FL-FLP) * (RL/RLP)**((FLP*RL-FL*RLP)/(RL-RLP))
        QLF(L) = RRQ / (RL*RL)
      ENDDO
```

12.5. Cancellation of the Thomson-scattering term

The opacity consists of the true opacity and the Thomson-scattering opacity, $\kappa = \kappa_{\rm true} + \kappa_{\rm e}$. The same holds for the emissivity, where $\eta_{\rm e} = \kappa_{\rm e}\, J$.

Therefore, the right-hand side of the 0. moment equation can be rewritten as $\kappa\,(S - J) = \eta_{\rm true} - \kappa_{\rm true}\, J$.

$$\frac{\partial \tilde{H}_\nu}{\partial r} - \left(\frac{{\rm d}V}{{\rm d}r} - \frac{V}{r}\right)\frac{\partial \tilde{K}_\nu}{\partial x} - \frac{V}{r}\frac{\partial \tilde{J}_\nu}{\partial x} = \tilde{\eta}_\nu^{\rm true} - \kappa^{\rm true}\,\tilde{J}_\nu \tag{178}$$

while the 1. moment equation remains unchanged, i.e. here the opacity involved is the *full* opacity κ including the Thomson term:

$$\frac{\partial (q\tilde{K}_\nu)}{-q\,\partial r} + \left(\frac{{\rm d}V}{{\rm d}r} - \frac{V}{r}\right)\frac{\partial \tilde{N}_\nu}{\partial x} + \frac{V}{r}\frac{\partial \tilde{H}_\nu}{\partial x} = \kappa\,\tilde{H}_\nu \tag{179}$$

12.6. Eddington factors

The moment equations can be closed by the introduction of the Eddington factors. The original suggestion by Mihalas et al. was to use

$$f = K/J \quad \text{and} \quad g = N/H\,. \tag{180}$$

However, there is no reason why fluxes cannot become negative at some frequencies and depths. Hence the definition of g can become singular. Therefore we introduce a different definition,

$$g = \frac{N}{H + \epsilon J} \tag{181}$$

Different values for ϵ may be chosen at each radius point, but ϵ may not depend on x in order to avoid frequency derivatives. Our original idea was to minimize the deviation from Mihals' idea, as might have had the right intuition about the best convergence properties. Therefore we have a complicated mechanism to control the minimal choice of ϵ in the program COLI.

In order to avoid a singularity in Eq. (181), we demand $\epsilon > -H/J$. This should never request an $\epsilon > 2$, since from the definition of the moments follows that

$$-J < H < J\,. \tag{182}$$

However, we will see below that we need to put a further constraint (Eq. 204) on ϵ to assure that the denominator in the numerical coefficients α, β and γ (see Eqs. 198–200) is positive. Considering the *Riemann Characteristics* of the coupled moment equations (see Appendix D), the same condition can be shown to assure their hyperbolic type.

Today I think that it would be best to define simply $g := N/J$. This would lead to a simplification of the equations and coefficients. However, we have not yet checked whether this would automatically avoid all singularities in the coefficients, and if the hyperbolic type would be always guaranteed. This should be investigated!

Whatsoever, for given Eddington factors, Eqs. (169,170) provide two partial differential equations for the two unknowns, $\tilde{J}$ and $\tilde{H}$:

0. Moment:

$$\frac{\partial \tilde{H}_\nu}{\partial r} - \left(\frac{\mathrm{d}V}{\mathrm{d}r} - \frac{V}{r}\right)\frac{\partial (f\tilde{J}_\nu)}{\partial x} - \frac{V}{r}\frac{\partial \tilde{J}_\nu}{\partial x} = \tilde{\eta}_\nu^{\mathrm{true}} - \kappa^{\mathrm{true}}\tilde{J}_\nu \tag{183}$$

1. Moment:

$$\frac{\partial (qf\tilde{J}_\nu)}{-q\,\partial r} + \left(\frac{\mathrm{d}V}{\mathrm{d}r} - \frac{V}{r}\right)\frac{\partial (g\tilde{H}_\nu + g\epsilon\tilde{J}_\nu)}{\partial x} + \frac{V}{r}\frac{\partial \tilde{H}_\nu}{\partial x} = \kappa\,\tilde{H}_\nu \tag{184}$$

12.7. Solution of the moment equation by a differencing scheme

12.7.1. Inner points

These equations are solved by a differencing scheme which is of second order in radius and first order in frequency. We introduce radius points r_l and frequency points x_k. The moment J is defined at the radial points, while the moment H (flux) is defined at the "interstices", i.e. the midpoints between the radial points. Interstices are denoted with half-number indices, i.e. $r_{l+1/2}$. The tilde over the discretized moments J, H and source function S is dropped in the following for simplicity.

0. Moment Equation:

Differencing Eq. (183) at (k,l) yields

$$\begin{aligned}\frac{H_{k,l+1/2} - H_{k,l-1/2}}{D_l} + \left(\mathrm{grad}_l - \frac{\mathrm{velo}_l}{r_l}\right)\frac{1}{\Delta x}\,(f_{k-1,l}J_{k-1,l} - f_{k,l}J_{k,l}) \;+& \\ \frac{\mathrm{velo}_l}{r_l\Delta x}\,(J_{k-1,l} - J_{k,l}) \;=& \\ \kappa_{k,l}(J_{k,l} - S_{k,l})&\end{aligned} \tag{185}$$

Here we have changed the notation for the velocity, because the symbol V will be needed in the following for a different meaning. Now "grad" denotes the radial velocity gradient $\frac{\mathrm{d}V}{\mathrm{d}r}$, and "velo" means the dimensionless velocity.

Note that the frequency derivative is onesided: the differences quotient is taken between x_K and x_{k-1}, but the rest of the equation is evaluated at x_k. Since the frequecies are indexed in a falling sequence, we have

$$\frac{\partial J}{\partial x} = \lim_{\Delta x \to 0}\frac{J_{k-1} - J_k}{\Delta x} \tag{186}$$

with

$$\Delta x := x_{k-1} - x_k \ . \tag{187}$$

In contrast, the differencing quotient over r is centered, i.e. the difference is taken between the interstices $r_{l-1/2}$ and $r_{l+1/2}$. Therefore we define

$$D_l := r_{l-1/2} - r_{l+1/2}$$

Since the radius grid is also indexed in a falling sequence,

$$\frac{\partial H}{\partial r} = \lim_{D_l \to 0} \frac{H_{k,l-1/2} - H_{k,l+1/2}}{D_l} \tag{188}$$

All signs have been inverted when going from Eq. (183) to Eq. (185).

For a shorter notation we define

$$V_l := \frac{\text{velo}_l}{r_l \Delta x} \tag{189}$$

and

$$G_l := \left(\text{grad}_l - \frac{\text{velo}_l}{r_l}\right) \frac{1}{\Delta x} \tag{190}$$

and rewrite Eq. (185) to

$$\begin{aligned} \frac{1}{D_l}\left(H_{k,l+1/2} - H_{k,l-1/2}\right) + G_l\left(f_{k-1,l}J_{k-1,l} - f_{k,l}J_{k,l}\right) + V_l\left(J_{k-1,l} - J_{k,l}\right) - \\ \kappa_{k,l}(J_{k,l} - S_{k,l}) \ = \ 0 \end{aligned} \tag{191}$$

1. Moment Equation:

The aim now is to eliminate $H_{k,l+1/2}$ and $H_{k,l-1/2}$ from the previous Eq. (191). For this purpose the 1. moment equation Eq. (184) is discretized at the "interstices" in radius, once for the radius point $r_{l+1/2}$ and a second time for $r_{l-1/2}$. This will lead to a second-order differencing scheme in r.

For the term with the Eddi-mix parameter ϵ we need J at the interstice, which is achieved by taking the its mean from the adjacent full points.

For $(k, l + 1/2)$ we get

$$\begin{aligned} \frac{q_{k,l+1} f_{k,l+1} J_{k,l+1} - q_{k,l} f_{k,l} J_{k,l}}{q_{k,l+1/2}(r_l - r_{l+1})} + \\ \left(\text{grad}_{l+1/2} - \frac{\text{velo}_{l+1/2}}{r_{l+1/2}}\right) \frac{1}{\Delta x}\left(g_{k-1,l+1/2}H_{k-1,l+1/2} - g_{k,l+1/2}H_{k,l+1/2}\right) + \\ \left(\text{grad}_{l+1/2} - \frac{\text{velo}_{l+1/2}}{r_{l+1/2}}\right) \frac{\epsilon_{l+1/2}}{2\Delta x}\left(g_{k-1,l+1/2}(J_{k-1,l} + J_{k-1,l+1}) - g_{k,l+1/2}(J_{k,l} + J_{k,l+1})\right) + \\ \frac{\text{velo}_{l+1/2}}{r_{l+1/2}} \frac{1}{\Delta x}\left(H_{k-1,l+1/2} - H_{k,l+1/2}\right) = \\ \kappa_{k,l+1/2} H_{k,l+1/2} \end{aligned} \tag{192}$$

In analogy to but now for the interstices, we define

$$D_{l+1/2} = r_l - r_{l+1} \tag{193}$$

$$V_{l+1/2} = \frac{\text{velo}_{l+1/2}}{r_{l+1/2}\Delta x} \tag{194}$$

and

$$G_{l+1/2} = \left(\mathrm{grad}_{l+1/2} - \frac{\mathrm{velo}_{l+1/2}}{r_{l+1/2}}\right)\frac{1}{\Delta x} \tag{195}$$

and obtain

$$\begin{aligned}
\frac{1}{D_{l+1/2}}\frac{q_{k,l+1}f_{k,l+1}J_{k,l+1} - q_{k,l}f_{k,l}J_{k,l}}{q_{k,l+1/2}} &+ \\
G_{l+1/2}\left(g_{k-1,l+1/2}H_{k-1,l+1/2} - g_{k,l+1/2}H_{k,l+1/2}\right) &+ \\
G_{l+1/2}\frac{\epsilon_{l+1/2}}{2}\left(g_{k-1,l+1/2}(J_{k-1,l} + J_{k-1,l+1}) - g_{k,l+1/2}(J_{k,l} + J_{k,l+1})\right) &+ \\
V_{l+1/2}\left(H_{k-1,l+1/2} - H_{k,l+1/2}\right) &= \\
\kappa_{k,l+1/2}\ H_{k,l+1/2} &
\end{aligned} \tag{196}$$

Assembling all terms with $H_{k,l+1/2}$ on the right-hand side leads to

$$\begin{aligned}
\frac{1}{D_{l+1/2}}\frac{q_{k,l+1}f_{k,l+1}J_{k,l+1} - q_{k,l}f_{k,l}J_{k,l}}{q_{k,l+1/2}} &+ \\
G_{l+1/2}g_{k-1,l+1/2}H_{k-1,l+1/2} &+ \\
G_{l+1/2}\frac{\epsilon_{l+1/2}}{2}\left(g_{k-1,l+1/2}(J_{k-1,l} + J_{k-1,l+1}) - g_{k,l+1/2}(J_{k,l} + J_{k,l+1})\right) &+ \\
V_{l+1/2}H_{k-1,l+1/2} &= \\
\left(\kappa_{k,l+1/2} + G_{l+1/2}g_{k,l+1/2} + V_{l+1/2}\right) H_{k,l+1/2} &
\end{aligned} \tag{197}$$

We introduce further abbreviations by

$$\alpha_{l+1/2} = \frac{1}{D_{l+1/2}}\frac{1}{\kappa_{k,l+1/2} + G_{l+1/2}g_{k,l+1/2} + V_{l+1/2}} \tag{198}$$

$$\beta_{l+1/2} = \frac{G_{l+1/2}}{\kappa_{k,l+1/2} + G_{l+1/2}g_{k,l+1/2} + V_{l+1/2}} \tag{199}$$

$$\gamma_{l+1/2} = \frac{V_{l+1/2}}{\kappa_{k,l+1/2} + G_{l+1/2}g_{k,l+1/2} + V_{l+1/2}} \tag{200}$$

Note that a sufficient condition for the denominaters being positive ist that $\kappa > 0$ (no Laser), and $gG + V > 0$. From their definition this means

$$g\frac{\mathrm{d}v}{\mathrm{d}r} + (1-g)\frac{v}{r} > 0 \ . \tag{201}$$

From $gG + V > 0$ follows with the definition of g in Eq. (181)

$$\frac{GN}{H + \epsilon J} > -V \ . \tag{202}$$

After having made sure already that the denominator is positive, this leads to

$$GN > -V\,(H + \epsilon J) \tag{203}$$

which finally leads to

$$\epsilon > -\left(\frac{GN}{V} + H\right)\Big/ J \tag{204}$$

as the second, in practice much stronger condition that must be imposed on the choice of ϵ in Subroutine COLI. With the help of the above definitions for α, β, and γ we resolve Eq. (197) for $H_{k,l+1/2}$:

$$
\begin{aligned}
H_{k,l+1/2} \quad = \quad & \alpha_{l+1/2}\frac{q_{k,l+1}f_{k,l+1}J_{k,l+1} - q_{k,l}f_{k,l}J_{k,l}}{q_{k,l+1/2}} + \\
& \beta_{l+1/2}\frac{\epsilon_{l+1/2}}{2}\left(g_{k-1,l+1/2}(J_{k-1,l} + J_{k-1,l+1}) - g_{k,l+1/2}(J_{k,l} + J_{k,l+1})\right) + \\
& \beta_{l+1/2}g_{k-1,l+1/2}H_{k-1,l+1/2} + \\
& \gamma_{l+1/2}H_{k-1,l+1/2}
\end{aligned}
$$

The analogues difference equation is also derived for the other interstice, i.e. at the radial point with index $l-1/2$:

$$
\begin{aligned}
H_{k,l-1/2} \quad = \quad & \alpha_{l-1/2}\frac{q_{k,l}f_{k,l}J_{k,l} - q_{k,l-1}f_{k,l-1}J_{k,l-1}}{q_{k,l-1/2}} + \\
& \beta_{l-1/2}\frac{\epsilon_{l-1/2}}{2}\left(g_{k-1,l-1/2}(J_{k-1,l-1} + J_{k-1,l}) - g_{k,l-1/2}(J_{k,l-1} + J_{k,l})\right) + \\
& \beta_{l-1/2}g_{k-1,l-1/2}H_{k-1,l-1/2} + \\
& \gamma_{l-1/2}H_{k-1,l-1/2}
\end{aligned}
$$

The last two equations are now inserted in (191) for eliminating the H_k, for the price of introducing the H_{k-1} with the previous frequency index into the equation, which is no problem since they are already known:

$$
\begin{aligned}
&\frac{1}{D_l}\left(\alpha_{l+1/2}\frac{q_{k,l+1}f_{k,l+1}J_{k,l+1} - q_{k,l}f_{k,l}J_{k,l}}{q_{k,l+1/2}} + \beta_{l+1/2}g_{k-1,l+1/2}H_{k-1,l+1/2} + \gamma_{l+1/2}H_{k-1,l+1/2}\right) + \\
&\qquad \frac{1}{D_l}\left(\beta_{l+1/2}\frac{\epsilon_{l+1/2}}{2}\left(g_{k-1,l+1/2}(J_{k-1,l} + J_{k-1,l+1}) - g_{k,l+1/2}(J_{k,l} + J_{k,l+1})\right)\right) + \\
&\frac{1}{D_l}\left(-\alpha_{l-1/2}\frac{q_{k,l}f_{k,l}J_{k,l} - q_{k,l-1}f_{k,l-1}J_{k,l-1}}{q_{k,l-1/2}} - \beta_{l-1/2}g_{k-1,l-1/2}H_{k-1,l-1/2} - \gamma_{l-1/2}H_{k-1,l-1/2}\right) \\
&\qquad -\frac{1}{D_l}\left(\beta_{l-1/2}\frac{\epsilon_{l-1/2}}{2}\left(g_{k-1,l-1/2}(J_{k-1,l-1} + J_{k-1,l}) - g_{k,l-1/2}(J_{k,l-1} + J_{k,l})\right)\right) + \\
&\qquad\qquad G_l\left(f_{k-1,l}J_{k-1,l} - f_{k,l}J_{k,l}\right) + V_l\left(J_{k-1,l} - J_{k,l}\right) - \\
&\qquad\qquad\qquad \kappa_{k,l}(J_{k,l} - S_{k,l}) = 0
\end{aligned}
\tag{205}
$$

The terms of this equation are now sorted and multiplied with -1

$$
\begin{aligned}
&\frac{1}{D_l}\left(-\alpha_{l+1/2}\frac{q_{k,l+1}f_{k,l+1}J_{k,l+1} - q_{k,l}f_{k,l}J_{k,l}}{q_{k,l+1/2}} + \alpha_{l-1/2}\frac{q_{k,l}f_{k,l}J_{k,l} - q_{k,l-1}f_{k,l-1}J_{k,l-1}}{q_{k,l-1/2}}\right) \qquad (206)\\
&\qquad +\frac{1}{D_l}\left(\beta_{l+1/2}\frac{\epsilon_{l+1/2}}{2}g_{k,l+1/2}(J_{k,l} + J_{k,l+1}) - \beta_{l-1/2}\frac{\epsilon_{l-1/2}}{2}g_{k,l-1/2}(J_{k,l-1} + J_{k,l})\right) \\
&\qquad\qquad +G_lf_{k,l}J_{k,l} + V_lJ_{k,l} + \kappa_{k,l}J_{k,l} = \\
&\frac{1}{D_l}\left(H_{k-1,l+1/2}(\beta_{l+1/2}g_{k-1,l+1/2} + \gamma_{l+1/2}) - H_{k-1,l-1/2}(\beta_{l-1/2}g_{k-1,l-1/2} + \gamma_{l-1/2})\right) \\
&+\frac{1}{D_l}\left(\beta_{l+1/2}\frac{\epsilon_{l+1/2}}{2}g_{k-1,l+1/2}(J_{k-1,l} + J_{k-1,l+1}) - \beta_{l-1/2}\frac{\epsilon_{l-1/2}}{2}g_{k-1,l-1/2}(J_{k-1,l-1} + J_{k-1,l})\right) \\
&\qquad\qquad +G_lf_{k-1,l}J_{k-1,l} + V_lJ_{k-1,l} + \kappa_{k,l}S_{k,l} \qquad (207)
\end{aligned}
$$

(208)

We sort this equation for terms with J_{l+1}, J_l, J_{l-1} and the left side and derive the following prefactors

C = coefficient of $-J_{k,l+1}$ =

$$\left(\frac{q_{k,l+1}f_{k,l+1}}{q_{k,l+1/2}}\alpha_{l+1/2} - \frac{1}{2}\beta_{l+1/2}\epsilon_{l+1/2}g_{k,l+1/2}\right)\Bigg/ D_l \tag{209}$$

B = coefficient of $J_{k,l}$ =

$$\begin{aligned}\left(\frac{q_{k,l}f_{k,l}}{q_{k,l+1/2}}\alpha_{l+1/2} + \frac{q_{k,l}f_{k,l}}{q_{k,l-1/2}}\alpha_{l-1/2} + \frac{1}{2}\beta_{l+1/2}\epsilon_{l+1/2}g_{k,l+1/2} - \frac{1}{2}\beta_{l-1/2}\epsilon_{l-1/2}g_{k,l-1/2}\right)\Bigg/ D_l \\ +G_l f_{k,l} + V_l + \kappa_{k,l}\end{aligned} \tag{210}$$

A = coefficient of $-J_{k,l-1}$ =

$$\left(\frac{q_{k,l-1}f_{k,l-1}}{q_{k,l-1/2}}\alpha_{l-1/2} + \frac{1}{2}\beta_{l-1/2}\epsilon_{l-1/2}g_{k,l-1/2}\right)\Bigg/ D_l \tag{211}$$

W = Right-hand-side =

$$\begin{aligned}&\frac{1}{D_l}\left(H_{k-1,l+1/2}(\beta_{l+1/2}g_{k-1,l+1/2} + \gamma_{l+1/2}) - H_{k-1,l-1/2}(\beta_{l-1/2}g_{k-1,l-1/2} + \gamma_{l-1/2})\right) \\ +&\frac{1}{D_l}\left(\beta_{l+1/2}\frac{\epsilon_{l+1/2}}{2}g_{k-1,l+1/2}(J_{k-1,l} + J_{k-1,l+1}) - \beta_{l-1/2}\frac{\epsilon_{l-1/2}}{2}g_{k-1,l-1/2}(J_{k-1,l-1} + J_{k-1,l})\right) \\ &+G_l f_{k-1,l}J_{k-1,l} + V_l J_{k-1,l} + \kappa_{k,l}S_{k,l}\end{aligned} \tag{212}$$

12.7.2. Boundary in space

The boundary condition is based on equation (192). To get an equation in J we substitute

$$gH = N = nJ \quad \text{and} \quad H = hJ$$

and derive at the outer boundary ($l = 1$) with the spacial derivation being righthanded

$$\begin{aligned}\frac{q_{k,2}f_{k,2}J_{k,2} - q_{k,1}f_{k,1}J_{k,1}}{q_{k,1+1/2}(r_1 - r_2)} &+ \\ \left(\text{grad}_1 - \frac{v_1}{r_1}\right)\frac{1}{\Delta x}(-n_{k,1}J_{k,1} + n_{k-1,1}J_{k-1,1}) &+ \\ \frac{v_1}{r_1}\frac{1}{\Delta x}(-h_{k,1}J_{k,1} + h_{k-1,1}J_{k-1,1}) &= \kappa_{k,1}H_{k,1} \\ &= \kappa_{k,1}(H^- + hJ_{k,1})\end{aligned}$$

Again we sort for $J_{k,1}$, $J_{k,2}$ and the right-hand-side and get with the short cut $D_1 = r_1 - r_2$

$$J_{k,2} \rightarrow \frac{q_{k,2}f_{k,2}}{q_{k,1+1/2}D_1}$$

$$\begin{aligned}J_{k,1} \quad \rightarrow \quad &\frac{-q_{k,1}f_{k,1}}{q_{k,1+1/2}D_1} - \left(\text{grad}_1 - \frac{v_1}{r_1}\right)\frac{1}{\Delta x}(n_{k,1}) - \frac{v_1}{r_1}\frac{1}{\Delta x}(h_{k,1}) - \kappa_{k,1}h_{k,1} = \\ &\frac{-q_{k,1}f_{k,1}}{q_{k,1+1/2}D_1} - G_1 n_{k,1} - V_1 h_{k,1} - \kappa_{k,1}h_{k,1}\end{aligned}$$

$$\text{R.S.} \quad \rightarrow \quad -\left(\text{grad}_1 - \frac{v_1}{r_1}\right)\frac{1}{\Delta x}(n_{k-1,1}J_{k-1,1}) - \frac{v_1}{r_1}\frac{1}{\Delta x}(h_{k-1,1}J_{k-1,1}) + \kappa_{k,1}H^- =$$
$$-G_1(n_{k-1,1}J_{k-1,1}) - V_1(n_{k-1,1}J_{k-1,1}) + \kappa_{k,1}\,H^-$$

At the inner boundary we write equation (192) with the spatial derivative being lefthanded as

$$\frac{q_{k,\text{ND}}f_{k,\text{ND}}J_{k,\text{ND}} - q_{k,\text{ND}-1}f_{k,\text{ND}-1}J_{k,\text{ND}-1}}{q_{k,\text{ND}-1/2}(r_{\text{ND}-1} - r_{\text{ND}})} +$$
$$\left(\text{grad}_{\text{ND}} - \frac{v_{\text{ND}}}{r_{\text{ND}}}\right)\frac{1}{\Delta x}(-n_{k,\text{ND}}J_{k,\text{ND}} + n_{k-1,\text{ND}}J_{k-1,\text{ND}}) +$$
$$\frac{v_{\text{ND}}}{r_{\text{ND}}}\frac{1}{\Delta x}(-h_{k,\text{ND}}J_{k,\text{ND}} + h_{k-1,\text{ND}}J_{k-1,\text{ND}}) \quad = \quad \kappa_{k,\text{ND}}H_{k,\text{ND}}$$

We write Eq. (156) in the form

$$H_{k,\text{ND}} = \int I^+\mu d\mu - \int u\mu d\mu = H^+ - hJ_{\text{ND}}$$

with assuming diffusion approximation (Eq 150) for I^+, yielding

$$\int I^+\mu d\mu = H^+ = \frac{B}{2} + \frac{1}{3\kappa_{\text{ND}}}\frac{\partial B}{\partial r}$$

and the Eddington factor h

$$h = \frac{\int u\mu\, d\mu}{\int u\, d\mu} = \frac{H_{\text{spec}}}{J_{k,\text{ND}}}$$

with $D_{\text{ND}} = r_{\text{ND}-1} - r_{\text{ND}}$ we get

$$J_{k,\text{ND}-1} \rightarrow \frac{-q_{k,\text{ND}-1}f_{k,\text{ND}-1}}{q_{k,\text{ND}-1/2}D_{\text{ND}}}$$

$$J_{k,\text{ND}} \quad \rightarrow \quad \frac{q_{k,\text{ND}}f_{k,\text{ND}}}{q_{k,\text{ND}-1/2}D_{\text{ND}}} - \left(\text{grad}_{\text{ND}} - \frac{v_{\text{ND}}}{r_{\text{ND}}}\right)\frac{1}{\Delta x}(n_{k,\text{ND}}) - \frac{v_{\text{ND}}}{r_{\text{ND}}}\frac{1}{\Delta x}(h_{k,\text{ND}}) + \kappa_{k,\text{ND}}h_{k,\text{ND}} =$$
$$\frac{q_{k,\text{ND}}f_{k,\text{ND}}}{q_{k,\text{ND}-1/2}D_{\text{ND}}} - G_{\text{ND}}n_{k,\text{ND}} - V_{\text{ND}}h_{k,\text{ND}} + \kappa_{k,\text{ND}}h_{k,\text{ND}}$$

$$\text{R.S.} \quad \rightarrow \quad -\left(\text{grad}_{\text{ND}} - \frac{v_{\text{ND}}}{r_{\text{ND}}}\right)\frac{1}{\Delta x}(n_{k-1,\text{ND}}J_{k-1,\text{ND}}) - \frac{v_{\text{ND}}}{r_{\text{ND}}}\frac{1}{\Delta x}(h_{k-1,\text{ND}}J_{k-1,\text{ND}}) + \kappa_{\text{ND}}H^+ =$$
$$-G_{\text{ND}}(n_{k-1,\text{ND}}J_{k-1,\text{ND}}) - V_{\text{ND}}(n_{k-1,\text{ND}}J_{k-1,\text{ND}}) + \kappa_{\text{ND}}H^+$$

12.7.3. Boundary in frequency

Equation (205) is recursive to the moments H_{k-1} at the previous frequency index. Thus, in order to initialize this recursion, one needs to establish the radiation field at the bluemost frequency point. At this frequency point, we neglect all terms with frequency derivatives. Thus. Eq. (191) simplifies to

$$\frac{1}{D_l}\,(H_{k,l+1/2} - H_{k,l-1/2}) + \kappa_{k,l}(J_{k,l} - S_{k,l}) = 0 \tag{213}$$

This is achieved in the program by setting in subroutine `COLIMO` the factor `DNUEINV = 1. / DX` $= 1/\Delta x$ to `DNUEINV = 0`. Then, in the coefficients (Eqs. 209-212) automatically all terms with frequency derivatives vanish. Formally, the variables `XHLMO_OLD` $= H_{k-1}$ have to be initialized (in subroutine `COLI_SETZERO`).

12.7.4. Calculation of H

H is calculated from equation (192). The same prefactors are needed as before and additionally the old eddington factor g and the old flux H_{k-1}. Starting from this equation

$$
\begin{aligned}
&\frac{q_{k,l+1}f_{k,l+1}J_{k,l+1} - q_{k,l}f_{k,l}J_{k,l}}{q_{k,l+1/2}(r_l - r_{l+1})} + \\
&\left(\text{grad}_{l+1/2} - \frac{v_{l+1/2}}{r_{l+1/2}}\right)\frac{1}{\Delta x}\left(-g_{k,l+1/2}H_{k,l+1/2} + g_{k-1,l+1/2}H_{k-1,l+1/2}\right) + \\
&\frac{v_{l+1/2}}{r_{l+1/2}}\frac{1}{\Delta x}\left(-H_{k,l+1/2} + H_{k-1,l+1/2}\right) = \\
&\kappa_{k,d+1/2}H_{k,d+1/2}
\end{aligned}
\tag{214}
$$

we yield

$$
\begin{aligned}
&\frac{q_{k,l+1}f_{k,l+1}J_{k,l+1} - q_{k,l}f_{k,l}J_{k,l}}{q_{k,l+1/2}(r_l - r_{l+1})} + \\
&\left(\text{grad}_{l+1/2} - \frac{v_{l+1/2}}{r_{l+1/2}}\right)\frac{1}{\Delta x}\left(g_{k-1,l+1/2}H_{k-1,l+1/2}\right) + \\
&\frac{v_{l+1/2}}{r_{l+1/2}}\frac{1}{\Delta x}\left(H_{k-1,l+1/2}\right) = \\
&H_{k,l+1/2}\left[\kappa_{k,l+1/2} + \left(\text{grad}_{l+1/2} - \frac{v_{l+1/2}}{r_{l+1/2}}\right)\frac{1}{\Delta x}\left(g_{k,l+1/2}\right) + \frac{v_{l+1/2}}{r_{l+1/2}}\frac{1}{\Delta x}\right]
\end{aligned}
$$

With the above given abbreviations we derive

$$
H_{k,l+1/2} = \alpha_{l+1/2}\frac{(qfJ)_{k,l+1} - (qfJ)_{k,l}}{q_{k,l+1/2}} + \beta_{l+1/2}(gH)_{k-1,l+1/2} + \gamma_{l+1/2}H_{k-1,l+1/2} \tag{215}
$$

12.7.5. The coefficients of the linear equations

We predefine the geometry in the arrays `DLF, VLF2, GLF2` and `DLH, VLH2, GLH2`. This is done in SUBROUTINE COLIMOP. The variables ending with “`F`” are defined at the radius points while the variables ending with “`H`” are defined at the interstices. The former arrays have a length of ND while the latter ones have the length ND-1. The index i points to the interstice between i and $i + 1$.

`DLF, VLF` and `GLF` at inner points are defined by

$$
\begin{aligned}
\texttt{DLF}_i &= \frac{1}{2}(r_{i-1} - r_{i+1}) = -\Delta r \\
\texttt{VLF2}_i &= \frac{v_i}{r_i} \\
\texttt{GLF2}_i &= -\frac{v_{i+1} - v_{i-1}}{2\,\texttt{DLF}_i} - \frac{v_i}{r_i} = \text{grad}_i - \texttt{VLF2}_i
\end{aligned}
$$

At the inner boundary we write

$$
\begin{aligned}
\texttt{DLF}_{\text{ND}} &= r_{\text{ND}-1} - r_{\text{ND}} \\
\texttt{VLF2}_{\text{ND}} &= \frac{\text{velo}_{\text{ND}}}{r_{\text{ND}}} \\
\texttt{GLF2}_{\text{ND}} &= -\frac{\text{velo}_{\text{ND}} - \text{velo}_{\text{ND}-1}}{\texttt{DLF}_{\text{ND}}} - \frac{\text{velo}_{\text{ND}}}{r_{\text{ND}}} = \text{grad}_{\text{ND}} - \texttt{VLF2}_{\text{ND}}
\end{aligned}
$$

while at the outer boundary we write

$$\begin{aligned}
\mathtt{DLF}_1 &= r_1 - r_2 \\
\mathtt{VLF2}_1 &= \frac{v_1}{r_1} \\
\mathtt{GLF2}_1 &= -\frac{v_2 - v_1}{\mathtt{DLF}_1} - \frac{v_1}{r_1} = \mathrm{grad}_1 - \mathtt{VLF2}_1
\end{aligned}$$

At the interstices, DLH, VLH and GLH are defined by

$$\begin{aligned}
\mathtt{DLH}_i &= r_i - r_{i+1} \\
\mathtt{VLH2}_i &= \frac{v_{i+1} + v_i}{r_{i+1} + r_i} \\
\mathtt{GLH2}_i &= -\frac{v_{i+1} - v_i}{\mathtt{DLH}_i} - \mathtt{VLH2}_i
\end{aligned}$$

After the ray-by-ray computation has finished one frequency point the SUBROUTINE COLIMO is entered. First of all the quantities *GLF*, *GLH*, *VLF* and *VLH* are calculated with *DX* beeing the step in frequency

$$\begin{aligned}
VLF_i &= VLF2_i \,/\, DX \\
VLH_i &= VLH2_i \,/\, DX \\
GLF_i &= GLF2_i \,/\, DX \\
GLH_i &= GLH2_i \,/\, DX
\end{aligned}$$

Then the sphericity factor *QLF* is calculated at all depth points. This is taken from SUBROUTINE ELIMIN. The the interstice arrays (*QLH*, *OPAKH*, *EDDIGH*, *ALH*, *BLH* and *CLH* are calculated. The are defined by

$$\begin{aligned}
\mathtt{QLH}_i &= \frac{1}{2}(QLF_i + QLF_{i+1}) \\
\mathtt{OPAKH}_i &= \frac{1}{2}(OPAK_i + OPAK_{i+1}) \\
\mathtt{EDDIGH}_i &= \frac{1}{2}(EDDIG_i + EDDIG_{i+1})
\end{aligned}$$

We define a temporary variable *T*

$$T = \frac{1}{OPAKH_i - GLH_i * EDDIGH_i + VLH_i}$$

and define

$$\begin{aligned}
ALH_i &= \frac{T}{DLH_i} \\
BLH_i &= GLH_i * T \\
CLH_i &= VLH_i * T
\end{aligned}$$

The Eddington factors are defined in the MAIN PROGRAM COLI by

$$\begin{aligned}
EDDIF_i &= \frac{XKL_i}{XJL_i} \\
EDDIG_i &= \frac{XNL_i}{XHL_i}
\end{aligned}$$

with *XJL*, *XHL*, *XKL* and *XNL* beeing the moments of the intensity. The old Eddington factors from the last frequency are stored in the arrays *EDDIFO* and *EDDIGO*

The the tridiagonal Matrix $(-A, B, -C)$ and the right-hand-side (W) are calculated in order to solve the equation by SUBROUTINE INVTRI. For the inner points ($i = 1...\mathrm{ND} - 1$) we write

$$\begin{aligned}
-A_i &= -\frac{QLF_{i-1} * EDDIF_{i-1} * ALH_{i-1}}{QLH_{i-1} * DLF_i} \\
B_i &= -\frac{QLF_i * EDDIF_i}{DLF_i} * \left(\frac{ALH_i}{QLH_i} + \frac{ALH_{i-1}}{QLH_{i-1}}\right) - GLF_i * EDDIF_i - VLF_i - OPAK_i^{\mathrm{noth}} \\
-C &= -\frac{QLF_{i+1} * EDDIF_{i+1} * ALH_i}{QLH_i * DLF_i} \\
W &= -\frac{(BLH_i * EDDIGO_i + CLH_i) * XHLMO_i}{DLF_i} + \\
&\quad \frac{(BLH_{i-1} * EDDIGO_{i-1} + CLH_{i-1}) * XHLMO_{i-1}}{DLF_i} - \\
&\quad (GLF_i * EDDIFO_i + VLF_i) XJLMO_i - ETAK_i^{\mathrm{noth}}
\end{aligned}$$

At the inner boundary we define from the diffusion approximation incl. correction terms (see mathematical description above)

$$HPLUS = \frac{BCORE}{2} + \frac{DBDR}{3 * OPAK_{\mathrm{ND}}}$$

with $BCORE$ and $DBDR$ beeing the Planck radiation field at T_{ND} and $DBDR$ beeing the derivative with respect to r. $EDDIH2$ and $EDDIN2$ are the Eddington factors from

$$EDDIHIN = \frac{\int u_{\mathrm{ND}} \mu d\mu}{\int u_{\mathrm{ND}} d\mu}$$

$$EDDININ = \frac{\int u_{\mathrm{ND}} \mu^3 d\mu}{\int u_{\mathrm{ND}} d\mu}$$

We write the coefficients (incl. EDDIMIX)

$$-A_{\mathrm{ND}} = \frac{2}{DLF_{\mathrm{ND}}} \cdot \left(\frac{QLF_{\mathrm{ND}-1} \cdot EDDIF_{\mathrm{ND}-1} \cdot ALH_{\text{ND-1}}}{QLH_{\text{ND-1}}} + GEPS\,B_{\text{ND-1}}\right) \tag{216}$$

$$\begin{aligned}
B_{\mathrm{ND}} = {} & \frac{2}{DLF_{\mathrm{ND}}} \cdot \left(\frac{QLF_{\mathrm{ND}} \cdot EDDIF_{\mathrm{ND}} \cdot ALH_{\text{ND-1}}}{QLH_{\text{ND-1}}} - GEPS\,B_{\text{ND-1}}\right) \\
& + GLF_{\mathrm{ND}} \cdot EDDIF_{\mathrm{ND}} + VLF_{\mathrm{ND}} + OPAK_{\mathrm{ND}}^{\mathrm{noth}} \\
& + \frac{2}{DLF_{\mathrm{ND}}} \cdot EDDIHIN
\end{aligned} \tag{217}$$

$$-C_{\mathrm{ND}} = \text{not used} \tag{218}$$

$$\begin{aligned}
W_{\mathrm{ND}} = {} & \frac{2}{DLF_{\mathrm{ND}}} \cdot \Big[\Big(BLH_{\text{ND-1}} \cdot EDDIGO_{\text{ND-1}} + CLH_{\text{ND-1}}\Big) \cdot XJLMO_{\text{old,ND-1}} \\
& + GEPS\,BO_{\text{ND-1}} \cdot \Big(XJLMO_{\text{old,ND}} + XJLMO_{\text{old,ND-1}}\Big)\Big] \\
& + \Big(GLF_{\mathrm{ND}} \cdot EDDIFO_{\mathrm{ND}} + VLF_{\mathrm{ND}}\Big) \cdot XJLMO_{\text{old,ND}} + ETAK_{\mathrm{ND}}^{\mathrm{noth}} \cdot RADIUS_{\mathrm{ND}}^2 \\
& + \frac{2}{DLF_{\mathrm{ND}}} \cdot HPLUS
\end{aligned} \tag{219}$$

Note that the terms containing the frequency derivation are neglected at the inner boundary. Therefore the Eddington factor $EDDININ$ is not needed so far.

At the outer boundary $XIMINUS$ comes from the boundary condition and $EDDIH1$ and $EDDIN1$ are defined in the same manner as $EDDIH2$ and $EDDIN2$. We write

$$\begin{aligned}
-A_1 &= \text{not used} \\
B_1 &= -\frac{QLF_1 * EDDIF_1}{QLH_1 * DLF_1} - GLF_1 * EDDIN - (VLF_1 + OPAK_1) * EDDIH1 \\
-C_1 &= -\frac{QLF_2 * EDDIF_2}{QLH_1 * DLF_1} \\
W_1 &= -(GLF_1 * EDDINO + VLF_1 * EDDIHO) * XJLMO_1 + OPAK_1 * \frac{XIMINUS}{2}
\end{aligned}$$

with $EDDIHO$ and $EDDINO$ beeing the old Eddington factors.

We turn now to the coefficients of equation (215) to calculate the flux H. This equation can be written as :

$$\begin{aligned}
XHLMO_i &= \\
ALH_i \frac{QLF_{i+1}EDDIF_{i+1}XJLMO_{i+1} - QLF_iEDDIF_iXJLMO_i}{QLH_i} &+ \\
(BLH_iEDDIGO_i + CLH_i)XHLMO_{\mathrm{old},i}
\end{aligned}$$

12.8. Solution of the tri-diagonal system for J

After having prepared all coefficients, we are left with a system of ND equations in the form

$$-A_lJ_{l-1} + B_lJ_l - C_lJ_{l+1} = W_l. \tag{220}$$

with coefficients A_l, B_l, and C_l being typically positive and l denoting the depth index. This can be written in the form of

$$\mathbf{T}\vec{J} = \vec{W} \tag{221}$$

with a tridiagonal matrix $\mathbf{T}$. To obtain the solution for the vector $\vec{J}$, i.e. the radiation field at all depth points (for the current CMF frequency), the straight-forward way would be to invert the matrix $\mathbf{T}$ and calculate

$$\vec{J} = \mathbf{T}^{-1}\vec{W}. \tag{222}$$

The algebra for solving this tridiagonal system is performed in `INVTRI` and follows the notation from Rybicki & Hummer (1991, Appendix A). One can introduce the two further quantities

$$D_l = \frac{C_l}{B_l - A_lD_{l-1}} \qquad \text{with } D_1 := \frac{C_1}{B_1} \tag{223}$$

$$Z_l = \frac{W_l + A_lZ_{l-1}}{B_l - A_lD_{l-1}} \qquad \text{with } Z_1 := \frac{W_1}{B_1} \tag{224}$$

which reduces the system of equations to

$$J_l = D_lJ_{l+1} + Z_l \qquad \text{with } J_{\mathrm{ND}+1} := 0, \tag{225}$$

i.e. the radiation field can now be obtained in 2ND calculation steps without the need of a full matrix inversion. This is performed in the subroutine `INVTRI` which is called from `COLIMO`.

THE FOLLOWING SECTION (FROM ANDREAS) DESCRIBES ONE VERSIO OF THE ALO AND IS NOT YET IMPLEMENTED IN THE STANDARD BRANCH 10-Jan-2019

todo: INVTRI in COLI should be replaced with a Andreas' version

12.8.1. Extraction of the Λ^*-operator

In order to obtain a suitable approximate lambda operator not only the solution of Eq. (221) is required, but also the diagonal elements of $\mathbf{T}^{-1}$ are needed.

$$J_\nu = [\Lambda_\nu] S_\nu \tag{226}$$

If we divide all terms by the opacity κ_ν, our right-hand side vector is actually the sum of the source function S_ν and other terms, hence we can write $\vec{W} := \vec{W}' + \vec{S}$, i.e.

$$\vec{J} = \mathbf{T}^{-1}\vec{W}' + \mathbf{T}^{-1}\vec{S}. \tag{227}$$

Hence we can write

$$J_{l,\nu} = (\Lambda_\nu S_\nu)_l = \sum_{j=1}^{\mathrm{ND}} T^{-1}_{l,j} S_{j,\nu} + \sum_{j=1}^{\mathrm{ND}} T^{-1}_{l,j} W'_j. \tag{228}$$

Knowing that the diagonal of the full Λ-operator is a good choice for our approximate lambda operator Λ^*, we choose the diagonal elements of the linear contribution from Λ, i.e.

$$\Lambda^*_{l,\nu} := \frac{\partial J_{l,\nu}}{\partial S_{l,\nu}} = \frac{\partial}{\partial S_{l,\nu}} \sum_{j=1}^{\mathrm{ND}} T^{-1}_{l,j} S_{j,\nu} = \sum_{j=1}^{\mathrm{ND}} T^{-1}_{l,j} \delta_{l,j} = T^{-1}_{l,l} \tag{229}$$

The consequence is that we need to obtain the diagonal elements of $\mathbf{T}^{-1}$. Since we did not explicitly invert T to obtain the radiation field, we also need a similar technique to obtain these elements. A suitable method is described in Appendix B in Rybicki & Hummer (1991) and is based on combining backward and forward Gaussian elimination. After calculating the quantities

$$D_l = \frac{C_l}{B_l - A_l D_{l-1}} \qquad \text{with } D_1 := \frac{C_1}{B_1} \tag{230}$$

$$E_l = \frac{A_l}{B_l - C_l E_{l+1}} \qquad \text{with } E_{\mathrm{ND}} := \frac{A_{\mathrm{ND}}}{B_{\mathrm{ND}}} \tag{231}$$

$$\tag{232}$$

the diagonal elements of $\mathbf{T}^{-1}$ can simply be obtained via

$$T^{-1}_{l,l} = (1 - D_l E_{l+1})^{-1} \cdot (B_l - A_l D_{l-1})^{-1}\,. \tag{233}$$

This means that also the diagonal elements of $\mathbf{T}^{-1}$ can be found in just ND steps. Since the definition of D_l is identical to the one which we required to obtain J_ν, both operations can be performed in parallel and Λ^* can be obtained together with J_ν with almost no overhead.

Using these results it is also possible to approximate the partial derivative from the radiation field at the current CMF frequency to the last one. We can split up the right hand side vector $\vec{W}$ even further and write $\vec{W}' := \vec{W}'' + \vec{X}(J_{\nu,\mathrm{old}})$. The complete entries of $\vec{X}$ would a bit more complex, but assuming that the Eddington factor g_ν does not change significantly between the depth points $l-1$ and l, they reduce to

$$X_l = \frac{GLF_l \cdot EDDIFO_l + VLF_l}{OPAK_l^{\mathrm{noth}}} J_{\nu,\mathrm{old},l}. \tag{234}$$

From the full equation

$$\vec{J}_\nu = \mathbf{T}^{-1}\vec{S}_\nu + \mathbf{T}^{-1}\vec{W}''_\nu + \mathbf{T}^{-1}\vec{X}_\nu(J_{\nu,\mathrm{old}}) \tag{235}$$

we immediately deduce

$$\frac{\partial J_{\nu,l}}{\partial J_{\nu,\mathrm{old},l}} = T^{-1}_{l,l} \frac{GLF_l \cdot EDDIFO_l + VLF_l}{OPAK_l^{\mathrm{noth}}} \tag{236}$$

Both $\frac{\partial J_\nu}{\partial S_\nu}$ and $\frac{\partial J_\nu}{\partial J_{\nu,\mathrm{old}}}$ are passed to the later frequency integration (subroutine FREQUINT).

12.9. Overall scheme for solving the radiative transfer

`PROGRAM COLI` ...

The Eddington factors are calculated from the angle-dependent radiation transfer Eq. (128). This equation is solved “ray-by-ray” by a short-characteristic integration.

The “ray-by-ray” solution is only performed from time to time (usually each 6 grand iterations) in order to update the Eddington factors. In the meantime the “Eddies” are kept fixed and applied for solving the moment equations. The latter yield the angle-averaged intensities J which are used to evaluate the radiative rates for the statistical equations.

13. Rate equations

The program `steal` (= *statistical equations with approximate lambda operators*) calculates the population numbers; the rate matrix depends on the mean radiation field $J_\nu(r)$, the electron temperature $T_\mathrm{e}(r)$, and the electron density $n_\mathrm{e}(r)$. In the course of the model iteration, the *statistical equations* (program `steal`) are solved in turn with the *radiative transfer* (program `coli`).

Naively, the statistical equations are just a system of linear equations. However, with such approach the iteration would not converge with reaonable speed to a consistent solution which fulfills the radiative transfer equations *and* the rate equations simultaneously.

The way out is the use of *approximate lambda operators* ALOs. This method, together with the use of *net radiative brackets* (NRBs), make the rate equations *non-linear*.

In the following subsection, we describe how we solve the non-linear rate equations. The ALOs will be defined in Sect. 13.2.

13.1. Solving the Rate Equations

LINPOP calculates the new non-LTE population numbers and stores them in the array POPNUM. Therefore it is the key subroutine of the STEAL program. The radiative rates are calculated with the Scharmer radation field. (...)

LINPOP loops over all depth points to calculate the new population numbers for the specific depth. This is again done by a series of iterations that splits in two branches, Newton-Raphson and Broyden.

The Newton-Raphson branch caculates the following

$$\vec{n}_{k+1} = \vec{n}_k - (\vec{n}_k \cdot P - \vec{b}) \cdot M^{-1} \tag{237}$$

P is the coefficient Matrix, M its vector derivative:

$$M_{i,j} = \frac{\partial}{\partial n_i} \left(\sum_m n_m P_{m,j} \right) \tag{238}$$

In the code P is called RATCO, M is called DM and the vector $\vec{b}$ is usually referred as V1. The direkt solution of the equation is done in the subroutine LINSOL which first tries to invert the large matrix DM and then calculates V2 by multiplying the inverse of DM with the vector V1. For improving the numerical inversion of DM a so-called “afterburner” is applied right after the direct inversion.

Usually the transitions between the different atomic levels are limited to a single element on the one hand, but the coefficient matrix (RATCO) usually contains more than one element on the other hand, RATCO and also DM have a block diagonal structure. The only exeptions are an additional column that balances the electrons and some entries that occur in some non-linearized situations if GAMMA is used. As the inverse of a block diagonal matrix is simply a diagonal matrix containing the inverse blocks the large inversion can be split up into a set of block inversions as long as we neglect the additional column and the few other entries. It turns our that this works quite well and reduces the problem of inverting large matrices so that we are able to calculate more elements with each one having as much levels as the complete system had. The block inversion changed a few lines in LINPOP (reduced the rank to neglect the additional column) and replaced LINSOL with the new subroutine LINSOL_SPLIT. To ensure the matrix block structure the complete matrix is copyied block by block before it is inverted.

In both subroutines, LINSOL and LINSOL_SPLIT the “afterburner” then calculated a correction term and subtracts it form the first solution.

$$\vec{X}_{\text{new}} := \vec{X} - \left[\vec{X} \cdot A - \vec{B}\right] \cdot A^{-1} \tag{239}$$

This can be repeated more than once by increasing the value of a variable called IMPMAX. (Default: IMPMAX = 1)

After that V2 is added to the vector NLTE population number vector EN. Now serveral criteria are checked to decide whether the interation for the current depth point has converged or if another interation should be done. In addition it is checked if the maximum number of iterations (defined by ITMAX) or any fatal problem has occurred. In both cases the iteration process is stopped.

After the iteration loop the population numbers (POPNUM), the departure coefficients (DEPART) and the (relative) electron density (RNE) are updated. The derivative Matrix DM is stored in a mass storage file named DMFILE that is saved on the local machine in the tmp_2day/*user*/ass*kn* directory.

Afterwards the subroutine checks if the number of diverged points might be so high that it makes so sense to continue the model calculations at all. This threshold is defined in NKONV_TRESH and usually set to 30 points. (Note that this value is hardcoded while the number of depth points can be adjusted. With usually 50 or 70 depth points there should be no problems, but a major increase of the number of depth points might cause problems while a major decrease will simply disable this function.) This check is only done if the CARDS parameters AUTO_MODIFY (stored in BAUTO) and ABORT_AUTO_MODIFY (stored in BAUTO_ABORT) are set.

If AUTO_MODIFY is set and the number of non-converged depth points is not zero, but below the treshhold, there will be an interpolation over the non converged depth part between the nearest converged depth points. If the diverged points are at the inner or outer boundary there will be a two-point extrapolation.

13.2. Implementation of the Accelerated Lambda Iteration (ALI)

THE FOLLOWING SECTION (FROM ANDREAS) DESCRIBES ONE VERSIO OF THE ALO AND IS NOT YET IMPLEMENTED IN THE STANDARD BRANCH 10-Jan-2019

With $\Lambda^*_\nu = \frac{\partial J_\nu}{\partial S^{\text{true}}_\nu}$ given, the quantities for the ALI procedure can now be calculated. In order to avoid calculating all terms on a fine grid, a number of frequency-integrated quantities is calculated during the frequency integration (subroutine FREQUINT) in the COLI program. Following the basic ALI concept where we approximate the new radiation field via

$$J^{\text{app}} = \Lambda S^{\text{old}} + \Lambda^* \Delta S \tag{240}$$

$$= J^{\text{FS}} + \Lambda^* \left(S^{\text{new}} - S^{\text{old}}\right) \tag{241}$$

with J^{FS} being the radiation field given by the previous COLI job, the convergence is “accelerated” by considering the effect of the changes of the population numbers (via their effect on the source function) already during the calculation of the radiative rates. However, in order to properly consider the effects for the continuum and the lines we have to extract these reactions from $\Lambda^*_{l,\nu}$. For this we write the total true source function, i.e. without Thomson term ($S^{\text{true}}_\nu \equiv S^{\text{noTh}}_\nu$ in the code), as the sum of all their contributions, i.e.

$$S^{\text{true}}_\nu = \frac{\eta^{\text{true}}_\nu}{\kappa^{\text{true}}_\nu} = \frac{\eta^{\text{L}}_\nu + \eta^{\text{C}}_\nu}{\kappa^{\text{L}}_\nu + \kappa^{\text{C}}_\nu} \tag{242}$$

$$= \frac{\kappa^{\text{L}}_\nu}{\kappa^{\text{true}}_\nu} S_{\text{L}} + \frac{\kappa^{\text{C}}_\nu}{\kappa^{\text{true}}_\nu} S^{\text{C}}_\nu \tag{243}$$

$$= \sum_{\text{IND}=1}^{\text{Lines}} \frac{\kappa^{\text{IND}}_\nu}{\kappa^{\text{true}}_\nu} S^{\text{IND}}_\nu + \frac{\kappa^{\text{C}}_\nu}{\kappa^{\text{true}}_\nu} S^{\text{C}}_\nu. \tag{244}$$

In the last step we wrote the total line source function as a weighted sum of all single line source functions. In principle we could do the same for the (true) continuum source function S^{c}, but refrained from doing it here as we do not split the various continuum contributions in our ALI approach. Assuming that the total opacity stays roughly constant and no explicit frequency derivatives occur, we can now deduce

$$\Lambda_\nu^{*,\mathrm{IND}} = \frac{\partial J_\nu}{\partial S_\nu^{\mathrm{L}}} = \frac{\partial J_\nu}{\partial S_\nu}\frac{\partial S_\nu}{\partial S_\nu^{\mathrm{L}}} = \Lambda_\nu^* \frac{\kappa_\nu^{\mathrm{L}}}{\kappa_\nu^{\mathrm{true}}} \tag{245}$$

$$\Lambda_\nu^{*,\mathrm{C}} = \frac{\partial J_\nu}{\partial S_\nu^{\mathrm{c}}} = \frac{\partial J_\nu}{\partial S_\nu}\frac{\partial S_\nu}{\partial S_\nu^{\mathrm{c}}} = \Lambda_\nu^* \frac{\kappa_\nu^{\mathrm{C}}}{\kappa_\nu^{\mathrm{true}}} \tag{246}$$

which can then be used to obtain the necessary quantities. Of course all these quantities are depth-dependent, but in order to avoid an even more overloaded notation we will refrain from denoting a depth index to all quantities from this paragraph on.

13.2.1. Lines

To avoid any fine-frequency integral for the calculation of the radiative rates, the rates use the quantity $\bar{J}$ for each line as described in chapter [...]. For the application of the accelerated lambda operator (ALO) we have to replace $\bar{J}$ with a corresponding expression following the ALI concept of

$$J_\nu \to J_\nu + \Lambda_\nu^* \Delta S_\nu. \tag{247}$$

Applying this replacement into the definition of $\bar{J}$ yields for an individual line IND:

$$\bar{J}_{\mathrm{IND}}^{\mathrm{app}} = \int\limits_{\mathrm{line}} J_\nu \,\phi(\nu - \nu_{\mathrm{IND}})\,\mathrm{d}\nu + \int\limits_{\mathrm{line}} \Lambda_{l,\nu}^* \,\phi(\nu - \nu_{\mathrm{IND}})\,\Delta S_\nu^{\mathrm{true}}\,\mathrm{d}\nu \tag{248}$$

$$= \bar{J}_{\mathrm{IND}} + \int\limits_{\mathrm{line}} \Lambda_\nu^* \,\Delta S_\nu^{\mathrm{true}}\,\phi(\nu - \nu_{\mathrm{IND}})\,\mathrm{d}\nu \tag{249}$$

Now a second term arises which does contain the difference between the new and the old source function. Note that in this approach so far the total source function appears as we want to approximate the radiation field and this could be affected by other lines and continua blending with the current line IND. To break this down, we can write

$$\Lambda_\nu^* \,\Delta S_\nu^{\mathrm{true}} = \Lambda_\nu^* \frac{\kappa_\nu^{\mathrm{L}}}{\kappa_\nu^{\mathrm{true}}} \Delta S_\nu^{\mathrm{L}} + \Lambda_\nu^* \frac{\kappa_\nu^{\mathrm{C}}}{\kappa_\nu^{\mathrm{true}}} \Delta S_\nu^{\mathrm{c}} \tag{250}$$

$$= \Lambda_\nu^{*,\mathrm{L}} \Delta S_\nu^{\mathrm{L}} + \Lambda_\nu^{*,\mathrm{c}} \Delta S_\nu^{\mathrm{c}} \tag{251}$$

If we assume coherent scattering, the frequency-dependence of the source function for a single line vanishes. Approximating that this holds also for $\Delta S^{\mathrm{L}} = \sum \Delta S^{\mathrm{IND}}$, we can write:

$$\bar{J}_{\mathrm{IND}}^{\mathrm{app}} = \bar{J}_{\mathrm{IND}} + \Delta S^{\mathrm{L}} \int\limits_{\mathrm{line}} \Lambda_\nu^* \,\frac{\kappa_\nu^{\mathrm{L}}}{\kappa_\nu^{\mathrm{true}}}\,\phi(\nu - \nu_{\mathrm{IND}})\,\mathrm{d}\nu + \int\limits_{\mathrm{line}} \Lambda_\nu^* \frac{\kappa_\nu^{\mathrm{C}}}{\kappa_\nu^{\mathrm{true}}} \Delta S_\nu^{\mathrm{c}} \phi(\nu - \nu_{\mathrm{IND}})\,\mathrm{d}\nu \tag{252}$$

$$= \bar{J}_{\mathrm{IND}} + \Delta S^{\mathrm{L}} \bar{\Lambda}_{\mathrm{IND}}^* + \int\limits_{\mathrm{line}} \Lambda_\nu^* \frac{\kappa_\nu^{\mathrm{C}}}{\kappa_\nu^{\mathrm{true}}} \Delta S_\nu^{\mathrm{c}} \phi(\nu - \nu_{\mathrm{IND}})\,\mathrm{d}\nu \tag{253}$$

Note that in the second term we have the fraction of all line opacities over $\kappa_\nu^{\mathrm{true}}$, since we want to consider the total reaction of the radiation field resulting from all lines in the integral region. With the new source function out of the integral, the quantity

$$\bar{\Lambda}_{\mathrm{IND}}^* := \int\limits_{\mathrm{line}} \Lambda_\nu^* \,\frac{\kappa_\nu^{\mathrm{L}}}{\kappa_\nu^{\mathrm{true}}}\,\phi(\nu - \nu_{\mathrm{IND}})\,\mathrm{d}\nu \tag{254}$$

can now be calculated in the COLI program in parallel with $\bar{J}_{\text{IND}}$. These values are then stored as `XJLxxxx` or `XJLxxxxx` in the `MODEL` file with `xxxx` denoting the transition index IND. The third term describing the effect of continuum source function changes is unfortunately more tricky as the frequency-dependence is not as local as for the lines here. While one could in principle make a similar approximation as for $\bar{\Lambda}^{*}_{\text{IND}}$ or even include the continua in this term, numerical experiences showed that this is unfavorable. Instead we approximate the continuum changes directly by evaluation of the approximate continuum radiation field at the rest wavelength of the considered. The complete expression for the approximate line radiation field is therefore

$$\bar{J}^{\text{app}}_{\text{IND}} = \bar{J}_{\text{IND}} + \Delta S^{\text{L}} \bar{\Lambda}^{*}_{\text{IND}} + J^{\text{app}}_{\text{c}}(\nu_{\text{IND}}) - J_{\text{c}}(\nu_{\text{IND}}). \tag{255}$$

This quantity is calculated for all non-rudimental lines in the `STEAL` subroutine `SETXJL`. The calculation of $J^{\text{app}}_{\text{c}}(\nu)$ will be explained later on.

13.2.2. Generic (iron group) lines

For transitions of the generic iron group element we have to use a different approach as the cross-sections have a complex frequency-dependence. In this case we try to extract the direct reaction of J on the change of the population numbers. Here we need to assume that the population numbers of the generic element do not have major impact on other opacities. In this case we can write:

$$\left.\frac{\partial J_\nu}{\partial n^{\text{G}}_{l}}\right|^{\text{IND}} = \frac{\partial J_\nu}{\partial S} \frac{\partial S}{\partial \kappa^{\text{true}}_\nu} \frac{\partial \kappa^{\text{G}}_\nu}{\partial n^{\text{G}}_{l}} \tag{256}$$

$$= -\Lambda^{*}_{\nu} \frac{\eta^{\text{true}}_\nu}{(\kappa^{\text{true}}_\nu)^2} \frac{\kappa^{\text{IND}}_\nu}{n_l - G_{lu} n_u} \tag{257}$$

$$\text{with} \quad G_{lu} = \frac{g_l}{g_u} \exp\left[\frac{h}{k_{\text{B}} T} (\nu_{\text{IND}} - \nu)\right]$$

The indices l and u refer to the lower and upper superlevel of the transition IND. Note that their weights g_l and g_u are already complex expressions (Ref. Graefener et al. 2002) and not simple weights as for usual levels. The analogue considerations for the derivative to the upper level yield

$$\left.\frac{\partial J_\nu}{\partial n^{\text{G}}_{u}}\right|^{\text{IND}} = \frac{\partial J_\nu}{\partial S} \left(\frac{\partial S}{\partial \kappa^{\text{true}}_\nu} \frac{\partial \kappa^{\text{G}}_\nu}{\partial n^{\text{G}}_{u}} + \frac{\partial S}{\partial \eta^{\text{true}}_\nu} \frac{\partial \eta^{\text{G}}_\nu}{\partial n^{\text{G}}_{u}}\right) \tag{258}$$

$$= \Lambda^{*}_{\nu} \left(\frac{\eta^{\text{true}}_\nu}{(\kappa^{\text{true}}_\nu)^2} \frac{\kappa^{\text{IND}}_\nu G_{lu}}{n_l - G_{lu} n_u} + \frac{1}{\kappa^{\text{true}}_\nu} \frac{\eta^{\text{IND}}_\nu}{n_u}\right) \tag{259}$$

$$\tag{260}$$

Both quantities now must be integrated over the line and weighted with the profile function. Since the individual profile function is not available, but the cross-sections $\sigma^{\text{IND}}_{\nu}$ are, we can define

$$w^{\text{G}}_{l} := \frac{\int \left.\frac{\partial J_\nu}{\partial n^{\text{G}}_{l}}\right|^{\text{IND}} \sigma^{\text{IND}}_{\nu} \text{d}\nu}{\int \sigma^{\text{IND}}_{\nu} \text{d}\nu} \tag{261}$$

$$w^{\text{G}}_{u} := \frac{\int \left.\frac{\partial J_\nu}{\partial n^{\text{G}}_{u}}\right|^{\text{IND}} \sigma^{\text{IND}}_{\nu} \text{d}\nu}{\int \sigma^{\text{IND}}_{\nu} \text{d}\nu}. \tag{262}$$

$$\tag{263}$$

These values are stored in the `MODEL` file as `WFLxxxx` and `WFUxxxx` with `xxxx` being the transition index. The approximate radiation field for these transitions is then calculated in `SETXJL` via

$$\bar{J}^{\text{app}}_{\text{IND}} = \bar{J}_{\text{IND}} + w^{\text{G}}_{l} \left(n^{\text{new}}_{l} - n^{\text{old}}_{l}\right) + w^{\text{G}}_{u} \left(n^{\text{new}}_{u} - n^{\text{old}}_{u}\right). \tag{264}$$

13.2.3. Continua

The calculation of $J_{\mathrm{c}}^{\mathrm{app}}(\nu)$, i.e. the approximated continuum radiation field, is not done on an individual transition level, but instead only performed for the total continuum. In contrast to the lines, we cannot obtain a frequency-independent form of $\Delta S_{\nu}^{\mathrm{c}}$ and thus have to store the continuum fraction of Λ_{ν}^{*} on the coarse frequency grid which is available when calculating the transition rates and the approximate radiation field. Starting from

$$\frac{\partial J_{\nu}}{\partial S_{\nu}^{\mathrm{c}}} = \frac{\partial J_{\nu}}{\partial S_{\nu}^{\mathrm{true}}} \frac{\partial S_{\nu}^{\mathrm{true}}}{\partial S_{\nu}^{\mathrm{c}}} = \Lambda_{\nu}^{*} \frac{\kappa_{\nu}^{\mathrm{c}}}{\kappa_{\nu}^{\mathrm{true}}}, \tag{265}$$

we transform the continuum ALO to the coarse grid by an elaborated integration scheme. Instead of integrating $\frac{\partial J_{\nu}}{\partial S_{\nu}^{\mathrm{c}}}$ directly, the `DIAGTAU` version of the operator first does the transformation

$$w_{J_{\mathrm{c}}}^{\tau} := -\ln\left(1 - \frac{\partial J_{\nu}}{\partial S_{\nu}^{\mathrm{c}}}\right) \tag{266}$$

and then performs the integration to the coarse grid before transforming back in the subroutine `FREQUNORM`. The quantities of $\frac{\partial J_{\nu}}{\partial S_{\nu}^{\mathrm{c}}}$ on the coarse grid are labeled $w_{J_{\mathrm{c}}}$ and termed *Scharmer weights*. They are stored in the `MODEL` file as `WJCxxxx` with `xxxx` referring to the coarse frequency index.

The approximated continuum radiation field $J_{\mathrm{c}}^{\mathrm{app}}(\nu)$ is calculated in the `STEAL`-subroutine `SETXJC` via

$$J_{\mathrm{c}}^{\mathrm{app}}(\nu) = J_{\mathrm{c}}(\nu) + w_{J_{\mathrm{c}}}(\nu)\Delta S_{\mathrm{c}}(\nu) \tag{267}$$

on the coarse frequency grid using the total continuum source functions calculated with the old and new population numbers. In the case of current temperature corrections, an additional contribution $\Delta J_{\mathrm{c}}^{T}$ is added at the inner boundary to account for the direct temperature dependency:

$$\Delta J_{\mathrm{c}}^{T} = \frac{1}{2}\left[B_{\nu}(T_{\mathrm{ND}}^{\mathrm{new}}) - B_{\nu}(T_{\mathrm{ND}}^{\mathrm{old}})\right] \tag{268}$$

13.2.4. Additional derivative terms resulting from ALI

When the derivative matrix $\mathcal{M}$ is calculated, additional terms can occur due to the appearance of the source function in the approximated radiation field.

$$\begin{aligned} \mathcal{M}_{i,j} &= \frac{\partial}{\partial n_i}\left(\sum_m n_m P_{m,j}\right) \\ &= \sum_m \frac{\partial n_m}{\partial n_i} P_{m,j} + \sum_m n_m \frac{\partial P_{m,j}}{\partial n_i} \\ &= P_{i,j} + \sum_m D_{i,m,j} \end{aligned} \tag{269}$$

Due to the first term, the matrix $\mathcal{M}$ (or `DM` in the code) is initialized with the normal rate matrix $\mathbf{P}$ (aka `RATCO`). Afterwards the terms $D_{i,m,k} := n_m \frac{\partial P_{m,j}}{\partial n_i}$ are calculated in the subroutine `DERIV` and added to the corresponding entry $\mathcal{M}_{i,j}$. It is computationally favorable to loop over all line transitions and obtain for a single transition with the lower level l and upper level u the contributions to $\mathcal{M}_{i,l}$ and $\mathcal{M}_{i,u}$ instead of trying to obtain the complete expression for the $D_{i,m,j}$ straight forward. We denote $D_{i,u}$ as the contribution to $\mathcal{M}_{i,u}$ and can use the symmetric structure to immediately get $D_{i,l} = -D_{i,u}$. For a single line transition IND written in the form of a net radiative bracket we get the following contribution:

$$D_{i,u} = -n_u A_{u,l} \frac{\partial}{\partial n_i} \frac{\bar{J}^{\text{app}}_{\text{IND}}}{S^{\text{IND}}} \tag{270}$$

$$= n_u A_{u,l} \left[\frac{\bar{J}^{\text{app}}_{\text{IND}}}{(S^{\text{IND}})^2} \frac{\partial S^{\text{IND}}}{\partial n_i} - \frac{1}{S^{\text{IND}}} \frac{\partial \bar{J}^{\text{app}}_{\text{IND}}}{\partial n_i} \right] \tag{271}$$

We do neglect all explicit derivatives of the radiation field, i.e. $\frac{\partial}{\partial n_i} \bar{J}_{\text{IND}} = 0$, which means that the second term only appears if the ALO is used. For the first term all we need to know is the derivative of the (new) line source function, i.e.

$$\frac{\partial S^{\text{IND}}}{\partial n_i} = \left[\frac{\partial \eta^{\text{IND}}}{\partial n_i} \kappa^{\text{IND}} - \eta^{\text{IND}} \frac{\partial \kappa^{\text{IND}}}{\partial n_i} \right] \frac{1}{(\kappa^{\text{IND}})^2}, \tag{272}$$

so that the first part of the bracket in Eq. (271) then yields

$$\frac{\bar{J}^{\text{app}}_{\text{IND}}}{(S^{\text{IND}})^2} \frac{\partial S^{\text{IND}}}{\partial n_i} = \left[\frac{\partial \eta^{\text{IND}}}{\partial n_i} \kappa^{\text{IND}} - \eta^{\text{IND}} \frac{\partial \kappa^{\text{IND}}}{\partial n_i} \right] \frac{\bar{J}^{\text{app}}_{\text{IND}}}{(\eta^{\text{IND}})^2}. \tag{273}$$

The necessary derivatives of the line opacity and emissivity are prepared for all transitions in the subroutine `DLIOP`. From all derivatives to n_i, of course all those to levels other than to the corresponding upper and lower level of the transition vanish. Writing the line opacity and emissivity (without profile function) as

$$\eta^{\text{IND}} = \frac{h\nu_{\text{IND}}}{4\pi} A_{u,l}\, n_u \tag{274}$$

$$\kappa^{\text{IND}} = \frac{c^2}{8\pi\nu^2_{\text{IND}}} A_{u,l} \left(\frac{g_u}{g_l} n_l - n_u \right) \tag{275}$$

the remaining derivatives are:

$$\frac{\partial \eta^{\text{IND}}}{\partial n_l} = 0 \qquad \frac{\partial \kappa^{\text{IND}}}{\partial n_l} = \frac{c^2}{8\pi\nu^2_{\text{IND}}} A_{u,l} \frac{g_u}{g_l} \tag{276}$$

$$\frac{\partial \eta^{\text{IND}}}{\partial n_u} = \frac{h\nu_{\text{IND}}}{4\pi} A_{u,l} \qquad \frac{\partial \kappa^{\text{IND}}}{\partial n_u} = -\frac{c^2}{8\pi\nu^2_{\text{IND}}} A_{u,l} \tag{277}$$

(Note that additional factors n_{tot} and $1/\Delta\nu_{\text{Dop}}$ occur in `DLIOP` due to the usage of relative population numbers in the code and the compatibility with `LIOP`, which is also used in combination with the profile function. Fortunately, due to the fact that we always calculate terms of the source function $S = \eta/\kappa$, all dependencies of $\Delta\nu_{\text{Dop}}$ eventually cancel out.)

In the case of ALI, the second term in the bracket of Eq. (271) had to be taken into account for calculating $D_{i,u}$. As previously mentioned we omit explicit derivatives of the radiation field, so that we remain with the derivative of the ALO term:

$$\frac{1}{S^{\text{IND}}} \frac{\partial \bar{J}^{\text{app}}_{\text{IND}}}{\partial n_i} = \frac{1}{S^{\text{IND}}} \frac{\partial}{\partial n_i} \left[\bar{\Lambda}^{*}_{\text{IND}} \Delta S^{\text{L}} \right] \tag{278}$$

$$= \frac{\bar{\Lambda}^{*}_{\text{IND}}}{S^{\text{IND}}} \frac{\partial}{\partial n_i} \left[S^{\text{L}}_{\text{new}} - S^{\text{L}}_{\text{old}} \right] \tag{279}$$

$$= \frac{\bar{\Lambda}^{*}_{\text{IND}}}{S^{\text{IND}}} \frac{\partial S^{\text{L}}_{\text{new}}}{\partial n_i} \tag{280}$$

Similar to the radiation field we also neglect explicit derivatives of the ALO. Furthermore the old line source function $S^{\text{L}}_{\text{old}}$ cannot depend on the new population numbers and thus the problem finally reduces to the derivative of the new line source function. Note that $S^{\textsc{l}} \neq S^{\textsc{ind}}$ as we need to account for overlapping lines. This necessity is a direct consequence from our calculation of $\bar{\Lambda}^*_{\textsc{ind}}$ in Eq. (254), where we account for the fraction of all line opacities in the region of the considered line. Dropping the "new"-index for better readability the derivative of the total line source function yields

$$\frac{\partial S^{\textsc{l}}}{\partial n_i} = \frac{\partial}{\partial n_i} \frac{\eta^{\textsc{l}}}{\kappa^{\textsc{l}}} \tag{281}$$

$$= \left(\frac{\partial \eta^{\textsc{l}}}{\partial n_i} \kappa^{\textsc{l}} - \eta^{\textsc{l}} \frac{\partial \kappa^{\textsc{l}}}{\partial n_i} \right) \frac{1}{(\kappa^{\textsc{l}})^2} \tag{282}$$

$$= \frac{1}{\kappa^{\textsc{l}}} \sum_{\textsc{ilb}=1}^{\text{Lines}} \frac{\partial \eta^{\textsc{ilb}}}{\partial n_i} - \frac{S^{\textsc{l}}}{\kappa^{\textsc{l}}} \sum_{\textsc{ilb}=1}^{\text{Lines}} \frac{\partial \kappa^{\textsc{ilb}}}{\partial n_i} \tag{283}$$

and thus the total second term becomes

$$\frac{1}{S^{\textsc{ind}}} \frac{\partial \bar{J}^{\text{app}}_{\textsc{ind}}}{\partial n_i} = \frac{\bar{\Lambda}^*_{\textsc{ind}}}{S^{\textsc{ind}}} \left[\frac{1}{\kappa^{\textsc{l}}} \sum_{\textsc{ilb}=1}^{\text{Lines}} \frac{\partial \eta^{\textsc{ilb}}}{\partial n_i} - \frac{S^{\textsc{l}}}{\kappa^{\textsc{l}}} \sum_{\textsc{ilb}=1}^{\text{Lines}} \frac{\partial \kappa^{\textsc{ilb}}}{\partial n_i} \right]. \tag{284}$$

The ILB-index ranges over all line transitions overlapping with our IND-transition and includes IND itself.

For the transitions of the generic element, we do not account for any source function changes in the derivatives. However, we can easily account for the explicit derivative of the ALO-term our calculated quantities w^{G}_l and w^{G}_u (Eqs. 261, 262) are exactly these derivatives. Writing our radiative rates as

$$R_{u,l} = 0 \qquad\qquad R_{l,u} = A_{u,l} \left[1 + \bar{J}^{\text{app}}_{\textsc{ind}} \frac{c^2}{2h\nu^3_{\textsc{ind}}} \left(1 - \frac{g_u n_l}{g_l n_u} \right) \right] \tag{285}$$

we obtain

$$D^{\mathrm{G}}_{i,u} = A_{u,l} \frac{c^2}{2h\nu^3_{\textsc{ind}}} \left(n_u - \frac{g_u}{g_l} n_l \right) \frac{\partial \bar{J}^{\text{app}}_{\textsc{ind}}}{\partial n_i} \tag{286}$$

$$\text{with } \frac{\partial \bar{J}^{\text{app}}_{\textsc{ind}}}{\partial n_i} = \begin{cases} w^{\mathrm{G}}_l & \text{for } i = l \\ w^{\mathrm{G}}_u & \text{for } i = u \\ 0 & \text{else.} \end{cases} \tag{287}$$

13.2.4.1. Continuum (bound-free) derivatives For the continuum or bound-free rates, two branches exist, one using a net radiative bracket (NRB) approach and one without. By default, the NRB is used for all continua. The net radiative bracket for bound-free transitions is more complex than the bound-bound counterparts and reads

$$Z_{u,l} = 4\pi \frac{n_{\mathrm{e}}}{T^{3/2}} C_{\mathrm{bf}} \frac{g_l}{g_u} \int_{\nu_{\textsc{kon}}}^{\infty} \frac{\sigma^{\textsc{kon}}(\nu)}{h\nu} \exp\left(-\frac{h}{k_{\mathrm{B}} T} [\nu - \nu_{\textsc{kon}}] \right) \frac{2h\nu^3}{c^2} \left(1 - \frac{J^{\text{app}}_{\mathrm{c}}(\nu)}{S^{\textsc{kon}}(\nu)} \right) \mathrm{d}\nu \tag{288}$$

for a transition KON with the upper and lower levels u and l plus $\nu_{\textsc{kon}}$ denoting the edge frequency of the transition. In Eq. (288) we used the definitions

$$S^{\textsc{kon}}(\nu) = \frac{2h\nu^3}{c^2} \frac{G^{\textsc{kon}}(\nu)\, n_u}{n_l - G^{\textsc{kon}}(\nu)\, n_u} \tag{289}$$

$$\text{and} \qquad G^{\textsc{kon}}(\nu) = \frac{g_l}{g_u} C_{\mathrm{bf}} \frac{n_{\mathrm{e}}}{T^{3/2}} \exp\left(-\frac{h}{k_{\mathrm{B}} T} [\nu - \nu_{\textsc{kon}}] \right) \tag{290}$$

$$\text{using} \qquad C_{\mathrm{bf}} = \frac{1}{2} \left[\frac{h^2}{2\pi m_{\mathrm{e}} k} \right]^{3/2} \approx 2.07 \cdot 10^{-16} [\text{cgs}]. \tag{291}$$

For calculating the derivatives, it is helpful to simplify the expression for $Z_{u,l}$ using first $G^{\text{KON}}(\nu)$ directly in Eq. (288) and then inserting the expression for $S^{\text{KON}}(\nu)$:

$$Z_{u,l} = 4\pi \int\limits_{\nu_{\text{KON}}}^{\infty} \frac{\sigma^{\text{KON}}(\nu)}{h\nu} G^{\text{KON}}(\nu) \frac{2h\nu^3}{c^2} \left(1 - \frac{J_{\text{c}}^{\text{app}}(\nu)}{S^{\text{KON}}(\nu)}\right) \text{d}\nu \tag{292}$$

$$= 4\pi \int\limits_{\nu_{\text{KON}}}^{\infty} \frac{\sigma^{\text{KON}}(\nu)}{h\nu} G^{\text{KON}}(\nu) \left(\frac{2h\nu^3}{c^2} - J_{\text{c}}^{\text{app}}(\nu) \frac{n_l - G^{\text{KON}}(\nu)\, n_u}{G^{\text{KON}}(\nu)\, n_u}\right) \text{d}\nu \tag{293}$$

$$= 4\pi \int\limits_{\nu_{\text{KON}}}^{\infty} \frac{\sigma^{\text{KON}}(\nu)}{h\nu} \left(\frac{2h\nu^3}{c^2} G^{\text{KON}}(\nu) - J_{\text{c}}^{\text{app}}(\nu) \left[\frac{n_l}{n_u} - G^{\text{KON}}(\nu)\right]\right) \text{d}\nu \tag{294}$$

$$= 4\pi \int\limits_{\nu_{\text{KON}}}^{\infty} \frac{\sigma^{\text{KON}}(\nu)}{h\nu} \left(\frac{2h\nu^3}{c^2} G^{\text{KON}}(\nu) - J_{\text{c}}^{\text{app}}(\nu) \frac{n_l}{n_u} + J_{\text{c}}^{\text{app}}(\nu)\, G^{\text{KON}}(\nu)\right) \text{d}\nu \tag{295}$$

For the derivatives we now need to calculate

$$D_{i,u} = n_u \frac{\partial Z_{u,l}}{\partial n_i}. \tag{296}$$

The only non-vanishing terms occur for $n_i \in \{n_u, n_l, n_{\text{e}}\}$. We start with the derivatives to the electron density which only come into play via $G^{\text{KON}}(\nu)$. Since $G^{\text{KON}}(\nu)$ depends linear on n_{e}, we can use the fact that

$$\frac{\partial G^{\text{KON}}(\nu)}{\partial n_{\text{e}}} = \frac{G^{\text{KON}}(\nu)}{n_{\text{e}}} \tag{297}$$

and avoid repeating lengthy calculations in the code by reusing $Z_{u,l}$ and writing

$$D_{\text{e},u} = n_u \frac{\partial Z_{u,l}}{\partial n_{\text{e}}} = n_u\, 4\pi \int\limits_{\nu_{\text{KON}}}^{\infty} \frac{\sigma^{\text{KON}}(\nu)}{h\nu} \left(\frac{2h\nu^3}{c^2} \frac{\partial G^{\text{KON}}(\nu)}{\partial n_{\text{e}}} + J_{\text{c}}^{\text{app}}(\nu)\, \frac{\partial G^{\text{KON}}(\nu)}{\partial n_{\text{e}}}\right) \text{d}\nu \tag{298}$$

$$= \frac{n_u}{n_{\text{e}}} 4\pi \int\limits_{\nu_{\text{KON}}}^{\infty} \frac{\sigma^{\text{KON}}(\nu)}{h\nu} \left(\frac{2h\nu^3}{c^2} G^{\text{KON}}(\nu) + J_{\text{c}}^{\text{app}}(\nu)\, G^{\text{KON}}(\nu)\right) \text{d}\nu \tag{299}$$

$$= \frac{n_u}{n_{\text{e}}} Z_{u,l} + \frac{n_u}{n_{\text{e}}} 4\pi \int\limits_{\nu_{\text{KON}}}^{\infty} \frac{\sigma^{\text{KON}}(\nu)}{h\nu} J_{\text{c}}^{\text{app}}(\nu) \frac{n_l}{n_u} \text{d}\nu. \tag{300}$$

The second term is essentially a correction to the error we make in the first term. In terms of programming, this approach is favorable as we can reuse the already calculated $Z_{u,l}$ and the integral in the correction term is very similar to the expression we obtain for the derivatives to n_l and n_i. Without considering the ALI contribution and the usual neglection of explicit derivatives of the radiation field $J_{\text{c}}(\nu)$, only one term remains for these derivatives, namely

$$D_{i,u}\Big|_{i\in\{u,l\}} = -n_u \frac{\partial}{\partial n_i} \left(\frac{n_l}{n_u}\right) 4\pi \int\limits_{\nu_{\text{KON}}}^{\infty} \frac{\sigma^{\text{KON}}(\nu)}{h\nu} J_{\text{c}}^{\text{app}}(\nu)\, \text{d}\nu. \tag{301}$$

Hence we can write the total expression for all continuum derivatives in the following way:

$$D_{i,u} = D^{(1)}_{i,u} + D^{(2)}_{i,u} + D^{\text{ALO}}_{i,u} \tag{302}$$

$$\text{with } D^{(1)}_{i,u} = \begin{cases} \frac{n_u}{n_\text{e}} Z_{u,l} & \text{for } n_i = n_\text{e} \\ 0 & \text{else} \end{cases} \tag{303}$$

$$\text{and } D^{(2)}_{i,u} = f(n_i) \int\limits_{\nu_\text{KON}}^{\infty} \frac{4\pi}{h\nu}\, \sigma^{\text{KON}}(\nu)\, J^{\text{app}}_{\text{c}}(\nu)\, \text{d}\nu \tag{304}$$

$$\text{with } f(n_i) = \begin{cases} -1 & \text{for } n_i = n_l \\ \frac{n_l}{n_i} & \text{else} \end{cases} \tag{305}$$

Only the ALO-contribution $D^{\text{ALO}}_{i,u}$ remains to be determined. Essentially these are terms where we have a derivative of the radiation field, hence

$$D^{\text{ALO}}_{i,u} = -n_u\, 4\pi \int\limits_{\nu_\text{KON}}^{\infty} \frac{\sigma^{\text{KON}}(\nu)}{h\nu}\, \frac{\partial J^{\text{app}}_{\text{c}}(\nu)}{\partial n_i} \left[\frac{n_l}{n_u} - G^{\text{KON}}(\nu)\right] \text{d}\nu. \tag{306}$$

So far we could avoid to actually calculate $G^{\text{KON}}(\nu)$ for the derivatives. Now we use the Saha equation

$$\frac{n^*_l}{n^*_u} = \frac{T^{3/2}}{C_\text{bf}\, n_\text{e}} \frac{g_u}{g_l} \exp\left(-\frac{h}{k_\text{B} T}\, [\nu - \nu_\text{KON}]\right) \tag{307}$$

compared with Eq. (290) to obtain

$$G^{\text{KON}}(\nu) = \frac{n^*_l}{n^*_u} \exp\left(-\frac{h\nu}{k_\text{B}T}\right), \tag{308}$$

where n^*_u and n^*_l denote the upper and lower LTE population number. With this we can express $D^{\text{ALO}}_{i,u}$ as

$$D^{\text{ALO}}_{i,u} = -n_u\, 4\pi \int\limits_{\nu_\text{KON}}^{\infty} \frac{\sigma^{\text{KON}}(\nu)}{h\nu}\, \frac{\partial J^{\text{app}}_{\text{c}}(\nu)}{\partial n_i} \left[\frac{n_l}{n_u} - \frac{n^*_l}{n^*_u} \exp\left(-\frac{h\nu}{k_\text{B}T}\right)\right] \text{d}\nu \tag{309}$$

$$= -n_u \frac{n^*_l}{n^*_u}\, 4\pi \int\limits_{\nu_\text{KON}}^{\infty} \frac{\sigma^{\text{KON}}(\nu)}{h\nu}\, \frac{\partial J^{\text{app}}_{\text{c}}(\nu)}{\partial n_i} \left[\frac{n^*_u}{n^*_l} \frac{n_l}{n_u} - \exp\left(-\frac{h\nu}{k_\text{B}T}\right)\right] \text{d}\nu \tag{310}$$

$$\tag{311}$$

What remains to be calculated is the derivative of the ALO-contribution to the radiation field. Similar as for the line transitions, we do neglect explicit derivatives of the radiation field and the ALO itself. Thus we obtain

$$\frac{\partial J^{\text{app}}_{\text{c}}(\nu)}{\partial n_i} = \frac{\partial}{\partial n_i} \left[w_{J_\text{c}}(\nu) \Delta S_\text{c}(\nu)\right] = w_{J_\text{c}}(\nu) \frac{\partial S^{\text{new}}_{\text{c}}(\nu)}{\partial n_i} \tag{312}$$

as only the new source function can depend on the new population numbers. The derivative of the source function is straight forward,

$$\frac{\partial S_\text{c}(\nu)}{\partial n_i} = \left[\frac{\partial \eta^\text{c}(\nu)}{\partial n_i} \kappa^\text{c}(\nu) - \eta^\text{c}(\nu) \frac{\partial \kappa^\text{c}(\nu)}{\partial n_i}\right] \frac{1}{(\kappa^\text{c}(\nu))^2} \tag{313}$$

$$= \left[\frac{\partial \eta^\text{c}(\nu)}{\partial n_i} - S_\text{c}(\nu) \frac{\partial \kappa^\text{c}(\nu)}{\partial n_i}\right] \frac{1}{\kappa^\text{c}(\nu)}, \tag{314}$$

and – in contrast to the lines – can be done on the total continuum opacity and emissivity since our ALO also contains the reaction of the total (true) continuum. The summed up derivatives for κ^{c} and η^{c} are prepared in the subroutine DCOOP. For the individual bound-free transitions we use the expressions

$$\eta^{\mathrm{KON}}(\nu) = \frac{2h\nu^3}{c^2}\sigma^{\mathrm{KON}}(\nu)\,G^{\mathrm{KON}}(\nu)\,n_u \tag{315}$$

$$\text{and } \kappa^{\mathrm{KON}}(\nu) = \sigma^{\mathrm{KON}}(\nu)\left[n_l - G^{\mathrm{KON}}(\nu)\,n_u\right] \tag{316}$$

and therefore the derivatives to the upper and lower level as well as to the electron density yield

$$\frac{\partial\eta^{\mathrm{KON}}(\nu)}{\partial n_l} = 0 \qquad \frac{\partial\kappa^{\mathrm{KON}}(\nu)}{\partial n_l} = \sigma^{\mathrm{KON}}(\nu) \tag{317}$$

$$\frac{\partial\eta^{\mathrm{KON}}(\nu)}{\partial n_u} = \frac{2h\nu^3}{c^2}\sigma^{\mathrm{KON}}(\nu)\,G^{\mathrm{KON}}(\nu) \qquad \frac{\partial\kappa^{\mathrm{KON}}(\nu)}{\partial n_u} = -\sigma^{\mathrm{KON}}(\nu)\,G^{\mathrm{KON}}(\nu) \tag{318}$$

$$\frac{\partial\eta^{\mathrm{KON}}(\nu)}{\partial n_{\mathrm{e}}} = \sigma^{\mathrm{KON}}(\nu)\,G^{\mathrm{KON}}(\nu)\frac{n_u}{n_{\mathrm{e}}} \qquad \frac{\partial\kappa^{\mathrm{KON}}(\nu)}{\partial n_{\mathrm{e}}} = -\sigma^{\mathrm{KON}}(\nu)\,G^{\mathrm{KON}}(\nu)\frac{n_u}{n_{\mathrm{e}}}. \tag{319}$$

Unless explicitly mentioned, all derivative calculations for $D_{i,u}$ are performed in the subroutine DERIV. The branch without NRBs, which is usually deactivated, only accounts for $D_{i,u}^{\mathrm{ALO}}$ and neglects the other terms.

14. The temperature stratification

Energy must be conserved in the stellar atmosphere. The temperature stratification $T(r)$ follows from this condition. The consistent stratification $T(r)$ is established iteratively by temperature corrections which are calculated and applied in program `steal` before new population numbers are obtained from the rate equations. Temperature corrections can be switched off, or might be automatically suppressed as long as the model iteration shows too large corretions of the population numbers. Several options in the CARDS file regulate the numerics of the temperature corrections.

Originally, the temperature corrections used in `PoWR` are derived from the condition of *radiative equilibrium*. As an alternative or complementary method, we also implemented the *thermal balance* approach which can be numerically more stable in the optically thin part of the atmosphere, as it considers the energy conservation of the free electrons. Both approaches are briefly outlined in the following subsections.

14.1. Temperature corrections from radiative equilibrium

The radiative equilibrium is usually written in the following two forms which already implicitly include the energy equation:

$$4\pi \int_0^\infty \kappa_\nu \left(S_\nu - J_\nu\right) rmd\nu = 0 \tag{320}$$

$$4\pi \int_0^\infty H_\nu \mathrm{d}\nu = \sigma_\mathrm{SB} T_\mathrm{eff}^4 \tag{321}$$

Usually the first form (320) is meant when the term “radiative equilibrium” is used while the second one (321) is referred to as “flux conversation” as H_ν is the so-called Eddington flux and T_eff is constant. In fact, both forms reflect the energy conservation. Frequency integration of the zeroth moment of the transport equation leads to the second equation for H_ν when using the first one, so it is implicitly included anyhow. The first one is therefore sometimes called the “differential form” while the second one is referred to as the “integral form” of radiative equilibrium. Note that the Eqs. (320) and (321) refer to a static, plane-parallel atmosphere and needs to be adjusted for the application in the PoWR code. This generalization to spherically expanding non-gray atmospheres of method from Unsöld (1951, 1955) and Lucy (1964) is described in detail in Hamann & Gräfener (2003). It eventually leads to the following expression for the temperature correction:

$$\begin{aligned}\Delta T(r) = \frac{\pi}{4\sigma_\mathrm{SB}} \frac{1}{T^3(r) r^2 \kappa_S(r)} \Bigg\{ & - \int_0^\infty \left(\eta_\nu - \kappa_\nu J_\nu\right) \mathrm{d}\nu \\ & + \frac{\kappa_J(r)}{(qf)_J(r)} \int_{r'=r}^{R_\mathrm{max}} (q\kappa)_H(r') \left[\tilde{H}_0(r') - \tilde{H}(r')\right] \mathrm{d}r' \\ & + \kappa_J(r) \frac{(qf)_J(R_\mathrm{max})}{(qf)_J(r)} \frac{\tilde{H}_0(R_\mathrm{max}) - \tilde{H}(R_\mathrm{max})}{h_J(R_\mathrm{max})} \Bigg\}\end{aligned} \tag{322}$$

The first line of Eq. (322) after the bracket reflects the radiative equilibrium Eq. (320) while the second and the third stem from an integral over the generalized version of Eq. (321) with the second line referring to the integral and the third one to a constant that can be fixed by the boundary values at R_max[4].

[4] All three terms can be multiplied with a weighting factor in the PoWR code in order to force specific corrections being preferred or even switched off.

In a thin atmosphere the Eq. (320) can cause numerical problems. The situation is dominated by resonance lines where κ_ν is basically zero and S_ν and J_ν are large but should cancel each other as they do not contribute the free electrons and therefore should not affect the temperature. In fact, it can happen that small numerical errors are multiplied with a large value and therefore do not cancel each other, but lead to unphysical corrections.

14.2. Temperature corrections from thermal balance

A nice way to avoid the previously mentioned problems is to consider the thermal balance of the electrons, where bound-bound line transitions do not enter, instead of the radiative equilibrium. Although this temperature correction method was not used for hot star atmospheres until the 1990s, it goes back to the ideas of Hummer & Seaton (1963) and Hummer (1963) who applied it for planetary nebulae.

Since a proper temperature structure is crucial for a successful application of the hydrodynamic routines described later in this work, the "thermal balance" method was included in PoWR as a part of this thesis. While the method is described in detail by Kubát et al. (1999) and Kubát (2001), their notation is considerably different in certain details. Therefore the main equations will be given here in the notation that follows the particular implementation in PoWR. When comparing these equations with Kubát et al. (1999), it should be noted that PoWR does not use the so-called occupation probabilities w_i, i.e. we set $w_i \equiv 1$ in their equations.

The thermal balance of electrons is written as the difference between heating (Q^{H}) and cooling (Q^{C}) terms:

$$\Delta Q := Q^{\mathrm{H}} - Q^{\mathrm{C}} \tag{323}$$

$$:= Q^{\mathrm{H}}_{\mathrm{ff}} + Q^{\mathrm{H}}_{\mathrm{bf}} + Q^{\mathrm{H}}_{\mathrm{c}} - Q^{\mathrm{C}}_{\mathrm{ff}} - Q^{\mathrm{C}}_{\mathrm{bf}} - Q^{\mathrm{C}}_{\mathrm{c}} \tag{324}$$

Both, heating and cooling terms, consist of three contributions, namely free-free and bound-free transitions as well as collisions. In an ideal situation with a perfect balance we should have $\Delta Q = 0$. In reality the situation will of course differ, but it is exactly this aim that is used to obtain a correction term for the current temperature stratification.

The straight-forward way to obtain a temperature correction is to implement a Newton-Raphson scheme, i.e. calculating the temperature derivative of ΔQ and calculate the new temperature via

$$T_{\mathrm{new}} = T_{\mathrm{old}} - \frac{\Delta Q(T)}{\frac{\partial}{\partial T}\left(\Delta Q\right)}. \tag{325}$$

An example of the corrections for the thermal balance method compared to those from radiative equilibrium is shown in Fig. 9. For such a cool model, the correction from the thermal balance method in the outer part is much smoother than those obtained by radiative equilibrium. In the innermost part, flux consistency is usually more reliable and the thermal balance corrections are switched off.

In the following paragraphs the calculation of Q_{ff}, Q_{bf}, and Q_{c} as well as their temperature derivatives are described. The choice of the Newton-Raphson method especially requires a calculation of the derivatives. For the population numbers N_j and the radiation field J_ν no analytical derivatives exist and no proper numerical derivative can be calculated within an effort that would be legitimate for the purpose of providing a temperature correction in each iteration. Therefore N_j and J_ν are treated as if they would not depend on the temperature in the following calculations. This approximation is probably not as bad as it sounds as the "thermal balance" method focuses especially on the outer parts of the stellar atmosphere where the temperature dependencies of both, N_j and J_ν, should be weak. Furthermore, the derivatives only determine the convergence radius in the Newton-Raphson method, not the actual value of the solution.

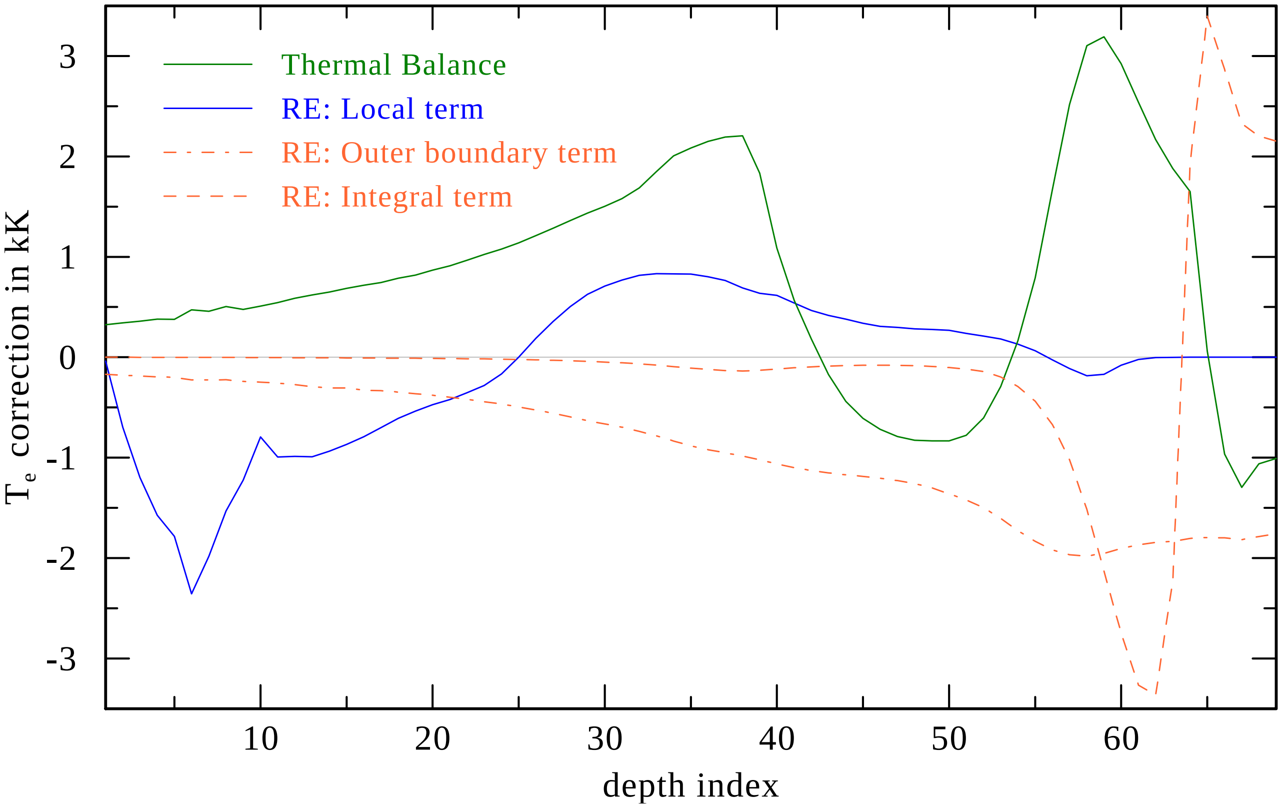


Figure 9: Electron temperature correction example for a non-converged B-star model with $T_* = 26\,\mathrm{kK}$: The corrections suggested by the thermal balance method (green curve) is compared to the three correction components from Eq. 322 based on radiative equilibrium (RE, blue and orange curves). The two orange curves together resemble the flux consistency terms.

14.2.1. Free-Free transitions

In free-free transitions all energy is transferred between the radiation field and the electrons. The heating terms describe the electron energy gains, thus $Q_{\mathrm{ff}}^{\mathrm{H}}$ consists of the energy gained by absorption, i.e.

$$Q_{\mathrm{ff}}^{\mathrm{H}} = 4\pi n_{\mathrm{e}} \sum_j N_j \int_0^\infty \alpha_{\mathrm{ff},j}(\nu, T) J_\nu \mathrm{d}\nu \tag{326}$$

with n_{e} being the electron density, and $\alpha_{\mathrm{ff},j}$ the free-free cross section. The radiation field J_ν is the radiation field on a coarse frequency grid, i.e. `XJC` in the PoWR code. For the cooling term $Q_{\mathrm{ff}}^{\mathrm{C}}$ we now have to sum up the emission processes, i.e.

$$Q_{\mathrm{ff}}^{\mathrm{C}} = 4\pi n_{\mathrm{e}} \sum_j N_j \int_0^\infty \alpha_{\mathrm{ff},j}(\nu, T) \left(J_\nu + \frac{2h\nu^3}{c^2} \right) e^{-\frac{h\nu}{kT}} \mathrm{d}\nu. \tag{327}$$

Note that both, $Q_{\mathrm{ff}}^{\mathrm{H}}$ and $Q_{\mathrm{ff}}^{\mathrm{C}}$, vanish for non-charged stages ($Z = 0$) due to the cross-section α_{ff} being proportional to Z^2:

$$\alpha_{\mathrm{ff},j}(\nu, T) = \frac{4e_0^6 Z^2}{3ch} \sqrt{\frac{2\pi}{3km_{\mathrm{e}}}} \frac{g_{\mathrm{ff}}(\nu, T)}{\nu^3 \sqrt{T}} \tag{328}$$

$$= 1.37 \cdot 10^{-23} \mathrm{cm}^5 Z^2 \left(\frac{T}{\mathrm{K}}\right)^{-\frac{1}{2}} \left(\frac{\lambda}{\mathrm{cm}}\right)^3 g_{\mathrm{ff}}(\nu, T) \tag{329}$$

Now the temperature derivative of both terms is needed. Neglecting the temperature dependencies of the population number and radiation field, only the α_{ff} coefficient remains in the heating term. Its derivative can be calculated straight forward:

$$\frac{\partial}{\partial T}\alpha_{\mathrm{ff},j}(\nu,T) = -\frac{1}{2T}\alpha_{\mathrm{ff},j}(\nu,T) + \frac{4e_0^6Z^2}{3ch}\sqrt{\frac{2\pi}{3km_{\mathrm{e}}}}\frac{\partial g_{\mathrm{ff}}(\nu,T)}{\partial T}\frac{1}{\nu^3\sqrt{T}} \tag{330}$$

For the Gaunt factor g_{ff} there is no analytic formula, but only tables depending on frequency ν and temperature T. However, this allows us to calculate $g_{\mathrm{ff}}(T)$ as well as $g_{\mathrm{ff}}(T+\delta t)$ in order to approximate the derivative of g_{ff} by a difference quotient. With all these terms given, the derivatives for the free-free heating and cooling terms can be obtained:

$$\frac{\partial}{\partial T}Q_{\mathrm{ff}}^{\mathrm{H}} = 4\pi n_{\mathrm{e}}\sum_j N_j\int_0^\infty \frac{\partial\alpha_{\mathrm{ff},j}}{\partial T}J_\nu \mathrm{d}\nu \tag{331}$$

$$\frac{\partial}{\partial T}Q_{\mathrm{ff}}^{\mathrm{C}} = 4\pi n_{\mathrm{e}}\sum_j N_j\int_0^\infty\left(\frac{\partial\alpha_{\mathrm{ff},j}}{\partial T} + \frac{h\nu}{kT^2}\alpha_{\mathrm{ff},j}\right)\cdot\left(J_\nu + \frac{2h\nu^3}{c^2}\right)e^{-\frac{h\nu}{kT}}\mathrm{d}\nu \tag{332}$$

The longer expression for the derivative of the cooling term (332) follows from the exponential factor in Eq. (327).

14.2.2. Bound-free transitions

The thermal balance of electrons is not affected by bound-bound transitions – in contrast to the radiative equilibrium – but by bound-free (BF) transitions where an electron is released or captured. Here ionizations lead to free electrons and therefore contribute excess kinetic energy to the heating. The resulting gain is

$$Q_{\mathrm{bf}}^{\mathrm{H}} = 4\pi\sum_{\mathrm{Kon=(l,u)}} N_l\int_{\nu_{lu}}^\infty \sigma_{lu}(\nu)J_\nu\left(1-\frac{\nu_{lu}}{\nu}\right)\mathrm{d}\nu, \tag{333}$$

with l denoting the lower (bound) level,u the upper level and ν_{lu} the ionization edge frequency. The term $\left(1-\frac{\nu_{lu}}{\nu}\right)$ accounts for the subtraction of the energy fraction that is transferred to the atoms, ensuring that only the energy transferred into electrons is covered here. In a similar way as for the free-free case, the corresponding term for cooling by recombination is

$$Q_{\mathrm{bf}}^{\mathrm{C}} = 4\pi\sum_{\mathrm{Kon=(l,u)}} N_u\frac{N_l^*}{N_u^*}\int_{\nu_{lu}}^\infty \sigma_{lu}(\nu)\left(J_\nu + \frac{2h\nu^3}{c^2}\right)e^{-\frac{h\nu}{kT}}\left(1-\frac{\nu_{lu}}{\nu}\right)\mathrm{d}\nu. \tag{334}$$

Population numbers marked with an asterisk, e.g. N_l^*, refer to the LTE-population number of the corresponding level. Neglecting the temperature dependence of the non-LTE populations numbers N and the radiation field J_ν, the temperature derivative of the heating term is zero. For the cooling term, there is an implicit Saha-Boltzmann factor in N_l^*/N_u^* and an explicit exponential factor that has to be taken into account:

$$\frac{\partial}{\partial T}Q_{\mathrm{bf}}^{\mathrm{H}} = 0 \tag{335}$$

$$\begin{aligned}\frac{\partial}{\partial T}Q_{\mathrm{bf}}^{\mathrm{C}} &= -\left(\frac{3}{2T} + \frac{h\nu_{lu}}{kT^2}\right)Q_{\mathrm{bf}}^{\mathrm{C}} \\ &\quad + 4\pi\frac{h}{kT^2}\sum_{\mathrm{Kon=(l,u)}} N_u\frac{N_l^*}{N_u^*}\int_{\nu_{lu}}^\infty \sigma_{lu}(\nu)\left(J_\nu + \frac{2h\nu^3}{c^2}\right)e^{-\frac{h\nu}{kT}}\left(1-\frac{\nu_{lu}}{\nu}\right)\nu\,\mathrm{d}\nu\end{aligned} \tag{336}$$

14.2.3. Collisions

Energy can also be transferred between atoms and electrons without involving the radiation field, namely by collisions. The heating component here consists of collisional recombinations and de-excitations

$$Q_{\mathrm{c}}^{\mathrm{H}} = n_{\mathrm{e}} \sum_{l,u} N_u \frac{N_l^*}{N_u^*} \, \Omega_{lu}(T) \, h\nu_{lu}, \tag{337}$$

while the cooling terms consists of ionization and excitation via collisions

$$Q_{\mathrm{c}}^{\mathrm{C}} = n_{\mathrm{e}} \sum_{l,u} N_l \, \Omega_{lu}(T) \, h\nu_{lu}. \tag{338}$$

Ω_{lu} is the so-called *collision strength*, its product with the electron density gives the collisional rates $C_{lu} = n_{\mathrm{e}}\Omega_{lu}$ which enter the rate coefficient matrix for the statistical equations. With the help of the LTE populations numbers, one can use the relation $N_l^* \, \Omega_{lu} = N_u^* \, \Omega_{ul}$. Note that the ratio between the LTE population numbers can be expressed by the (Saha-)Boltzmann equation and therefore introduces a temperature dependence. The temperature dependency of Ω_{lu} itself is not trivial as different formulae are used for each element.

In the PoWR code, the double sum over all upper and lower levels is replaced by more convenient loops. As collisions come in two different flavors, namely from line transitions (collisional excitation and deexcitation) and bound-free transitions (collisional ionization and recombination), the collisional rates are calculated differently. For the line transitions, Ω_{lu} is calculated internally, while it is Ω_{ul} for the bound-free transitions, so this has to be taken into account for the derivatives later on. This is the only part where line transitions affect the thermal balance. Additional care has to be taken for iron superlevels as it can happen that there are no radiative transitions between two levels which means that such transitions are not covered in a standard loop using the radiative transition list (`IND index`). As there can still be transitions via collisions, these contributions have to be added afterwards. However, their derivatives are not different in type and hence all equations are the same as for the other line transitions. We thus rewrite equations (337) and (338) for the line transitions in the following way:

$$Q_{\mathrm{c,IND}}^{\mathrm{H}} = n_{\mathrm{e}} \sum_{l,u} N_u \frac{N_l^*}{N_u^*} \, \Omega_{lu}(T) \, h\nu_{lu} \tag{339}$$

$$= n_{\mathrm{e}} \sum_{l,u} N_u \Omega_{ul}(T) \, h\nu_{lu} \tag{340}$$

$$= \sum_{l,u} N_u C_{ul}(T) \, h\nu_{lu} \tag{341}$$

$$Q_{\mathrm{c,IND}}^{\mathrm{C}} = \sum_{l,u} N_l \frac{N_u^*}{N_l^*} \, C_{ul}(T) \, h\nu_{lu} \tag{342}$$

This form reflects that for line transitions C_{ul} is calculated as $C_{ul} = n_{\mathrm{e}}\Omega_{ul}$ internally while C_{lu} is obtained by multiplication with the LTE population number ratio using the relation $C_{lu} = n_{\mathrm{e}} \frac{N_l^*}{N_u^*} \, \Omega_{ul}$. The temperature derivatives of the $Q_{\mathrm{c,IND}}$-terms can then be obtained by using an analytic derivative for the Boltzmann factor originating from the LTE population number ratio while the derivative of C_{ul} can be calculated numerically. Once again neglecting any temperature dependency of the non-LTE population number, the temperature derivatives are

$$\frac{\partial}{\partial T} Q_{\mathrm{c,IND}}^{\mathrm{H}} = \sum_{l,u} N_u \frac{\partial C_{ul}}{\partial T} \, h\nu_{lu} \tag{343}$$

$$\frac{\partial}{\partial T} Q_{\mathrm{c,IND}}^{\mathrm{C}} = \sum_{l,u} N_l \frac{N_u^*}{N_l^*} \left(\frac{\partial C_{ul}}{\partial T} + \frac{h\nu_{lu}}{kT^2} C_{ul} \right) h\nu_{lu}. \tag{344}$$

The collisional rates for the ionizations are calculated just the other way round. Here, $C_{lu} = n_\mathrm{e}\Omega_{lu}$ is calculated directly via Eq. (6.39) in Jefferies (1968) while C_{ul} is obtained via scaling with the corresponding Saha-Boltzmann factor. Hence, the equations are now written

$$Q^\mathrm{H}_\mathrm{c,KON} = \sum_{l,u} N_u \frac{N_l^*}{N_u^*}\, C_{lu}(T)\, h\nu_{lu} \tag{345}$$

$$Q^\mathrm{C}_\mathrm{c,KON} = \sum_{l,u} N_l\, C_{lu}(T)\, h\nu_{lu} \tag{346}$$

The corresponding temperature derivatives are:

$$\frac{\partial}{\partial T} Q^\mathrm{H}_\mathrm{c,KON} = \sum_{l,u} N_u \frac{N_l^*}{N_u^*} \left[\frac{\partial C_{lu}}{\partial T} - \left(\frac{3}{2T} + \frac{h\nu_{lu}}{kT^2}\right) C_{lu}\right] h\nu_{lu} \tag{347}$$

$$\frac{\partial}{\partial T} Q^\mathrm{C}_\mathrm{c,KON} = \sum_{l,u} N_l \frac{\partial C_{lu}}{\partial T}\, h\nu_{lu} \tag{348}$$

Note that there is a Saha-Boltzmann factor for the bound-free transitions instead of the pure Boltzmann factor as it was for the line transitions. Furthermore it might seem strange at first that ionizations enter in the heating term for bound-free transitions, while they appear in the cooling term for collisions, but one has to keep in mind that the latter ionizations are caused by another electron, so there is a net loss of electron energy in contrast to the bound-free transitions, where the ionization energy is provided by the radiation field.

14.3. Connection between radiative equilibrium and thermal balance

The two approaches described above, *radiative equilibrium* and *thermal balance*, are in fact mathematically equivalent. The previously defined Q^H and Q^C reflect the energy gains and losses for the free electrons. These values could also be obtained by the sum of the the radiative energy and the product of the ionization energies and the changes in the population numbers. Hence we can write

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = \underbrace{4\pi \int_0^\infty \kappa_\nu\,(S_\nu - J_\nu)\,\mathrm{d}\nu}_{=0 \text{ in radiative equilibrium}} + \underbrace{\sum_i h\nu_i \frac{\mathrm{d}N_i}{\mathrm{d}t}}_{=0 \text{ in statistical equilibrium}} \tag{349}$$

(see also Hillier & Miller 1998). The second term on the right hand side vanishes in statistical equilibrium, as the population numbers N_i do not change then. What remains is the first term which exactly vanishes in radiative equilibrium. Therefore in a situation with statistical and radiative equilibrium, the left hand side should be zero, but this is what we already have described as thermal balance. This illustrates that in statistical equilibrium, both correction methods should in theory lead to the same temperature structure. Numerically this is usually not the case, as both methods have different strengths and weaknesses which were previously mentioned. The proper method therefore has to be chosen depending on the specific model situation. The effective temperature or the departure from LTE in a certain part of the atmosphere can have a huge effect on how successful one or the other method exactly is.

15. Formal Integral: radiative transfer in the observer's frame

THIS SECTION IS UNDER CONSTRUCTION!

The job `formal` runs after the model iteration (`wruniq` jobs) is finally converged. Based of the thus established stratifications of temperature, density, velocity, and population numbers, the program `FORMAL` calculates the emergent spectrum.

More precisely, `FORMAL` reads the input file `FORMAL_CARDS`, which is usually composed with the help of the job `newformal_cards` (see Sect. 6.2). Each `RANGE` specified in `NEWFORMAL_CARDS_INPUT` creates a corresponding section in `FORMAL_CARDS` that starts with `BLEND` and ends with `-BLEND`. The program `formal` sequentially calculates the emergent spectra for each such *blend-block* alias `RANGE`, which are typically named `UV`, `OPT`, `K-BAND` et cetera. Each spectrum is issued in the form of plots and tables (optionally).

15.1. Wavelength and frequency grid

The formal integral is calculated in the observer's frame of reference. For each range, a grid of frequency points is defined. We introduce a dimensionless (observer's frame) frequency x by

$$\nu = \nu_0 \left(1 + \frac{\upsilon_\mathrm{D}}{c}\right)^x \tag{350}$$

or, equivalently,

$$\ln \nu = \ln \nu_0 \ + \mathtt{ALN}\ x\ . \tag{351}$$

with $\mathtt{ALN} := \ln\left(1 + \frac{\upsilon_\mathrm{D}}{c}\right)$. Hence,

$$x = \frac{\ln \nu - \ln \nu_0}{\mathtt{ALN}}\ . \tag{352}$$

In terms of wavelengths $\lambda = c/\nu$, this relation becomes

$$x = \frac{\ln \lambda_0 - \ln \lambda}{\mathtt{ALN}}\ . \tag{353}$$

since the speed of light c cancels out.

The Doppler unit is specified by υ_D, taken as the smallest Doppler-broadening velocity that occurs in the model atmosphere, accounting for thermal and microturbulent velocities.

The zero-point of the x-scale defined above is at $\nu_0 = c/\lambda_0$. We take for λ_0 the first line-center wavelength that is accidentally encountered in the atomic data for this range (named `XLAM` in the code, unfortunately). Note that $x < 0$ for wavelenths larger than λ_0.

Next, we need to define the grid points, indexed with k that runs from 1 to `NFOBS`, where $k = 1$ shall refer to the shortest wavelength (highest frequency) of the range. The corresponding dimensionless frequency is termed `FREMAX`). The step-width in the dimensionless frequency x is Δx.

Thus we define

$$x_0 = \mathtt{FREMAX} + \Delta x \tag{354}$$

and finally get the grid in terms of the dimensionless frequency

$$x_k = x_0 - k\ \Delta x \tag{355}$$

Translated into wavelengths, this corresponds to

$$\lambda_k = \lambda_0\ \exp\left(x_k\ \mathtt{ALN}\right) \tag{356}$$

(cf. beginning of k-loop in `formal`, with λ_k termed `XLAMREF` in the code).

The inverse calculation, k from a given wavelength, is needed in `LIMBDARK_PREP`:

$$k = \frac{x_0 + (\ln \lambda_0 - \ln \lambda)/\mathtt{ALN}}{\Delta x} \tag{357}$$

and gives the index k for $\lambda = \lambda_k$; for arbitrary λ one gets a fractional values for k that can be used, e.g., for interpolation.

For frequency integrals (also needed e.g. in `LIMBDARK_PREP`), note that the step-width scales like

$$\Delta\nu \propto \nu \propto [\exp(\mathtt{ALN})]^{x_k} \tag{358}$$

15.2. The emergent flux

In order to predict the spectrum as seen from an external observer, the radiative transfer must be solved in a fixed frame of reference – the *observer's frame*. This is performed in the program `formal` as a *formal solution*, i.e. on the basis of the level population numbers that have been established iteratively in the `wruniq` cycles.

The observer at distance d measures the emergent flux

$$F_\nu = \frac{R_*^2}{d^2} \int\limits_{\varphi=0}^{2\pi} \int\limits_{p=0}^{R_\mathrm{max}} I_\nu^+(p,\varphi)\; p \,\mathrm{d}p \,\mathrm{d}\varphi \tag{359}$$

where $I_\nu^+(p,\varphi)$ is the *emergent intensity* (see below). Note that $I_\nu^+(p,\varphi)$ comes in physical units. The factor $\frac{R_*^2}{d^2}$ is because p is normalized to R_*.

15.3. The emergent intensity

Calculation of the emergent intensity $I_\nu^+(p,\varphi)$ requires an integration along each ray:

$$I_\nu^+ = \int_0^{\tau_\mathrm{max}} S_\nu(z)\, e^{-\tau(z)}\,\mathrm{d}\tau \quad [+I_\nu^*\, e^{-\tau_\mathrm{max}}] \tag{360}$$

$S_\nu(z) = \eta_\nu(z)/\varkappa(\nu,z)$ is the source function at the observer's frame frequency ν and the position z along the considered ray, and

$$\tau(z) = \int_z^{z_\mathrm{max}} \varkappa(\nu,z')\,\mathrm{d}z' \tag{361}$$

the optical depth reached at z along that ray.

The term in square brackets in Eq. (360) applies only for rays which hit the inner boundary, with I_ν^* denoting the intensity of the core's radiation entering the atmosphere.

Numerically, the integral in Eq. (360) can be evaluated with the trapezoidal rule, achieving best accuracy when incorporating the $\exp(-\tau)$ factor into the quadrature weights. However, this formulation requires that $\varkappa(\nu) > 0$, i.e. it fails in laser situations. Therefore, if negative opacity is encountered in an integration step, the code switches for just this step to the alternative formulation

$$I_\nu^+ = \int_{z_\mathrm{max}}^{z_\mathrm{min}} \eta_\nu(z)\, e^{-\tau(z)}\,(-\mathrm{d}z) \tag{362}$$

TO BE CONTINUED

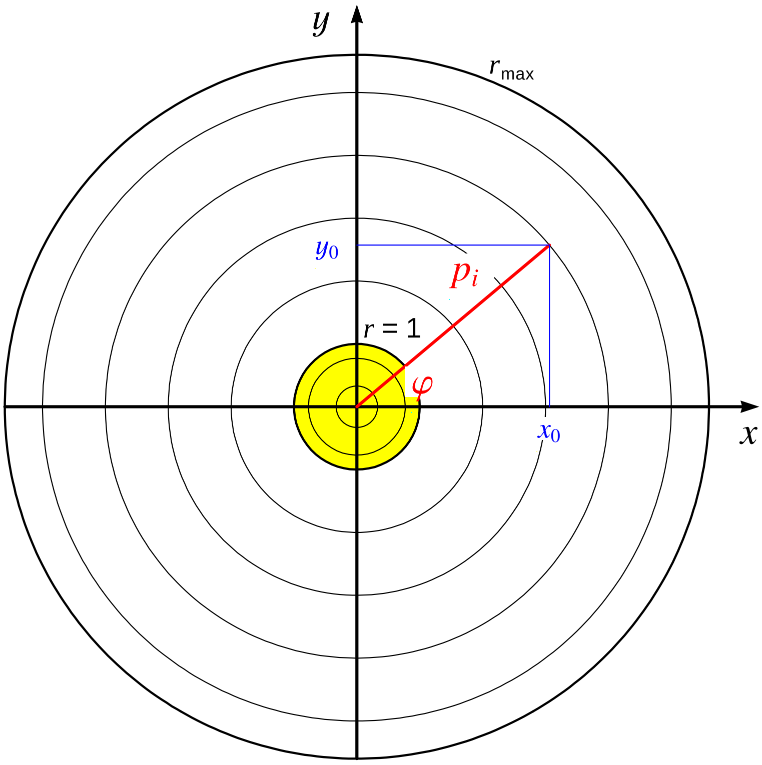


Figure 10: Integration of the emergent flux. The star is seen from the observer. In case of spherical symmetry, the emergent flux depends only on the impact-parameter p, but is independent from the azimuth angle φ. In case of wind rotation, or in case of two models being geometrically combined, this degeneracy will be lifted (see below).

15.4. Frequency redistribution by electron scattering

One contribution to the emissivity η in Eq. 360 is from photons scattered by free electrons. While this *Thomson scattering* is assumed to be *coherent* when iterating the model structure and population numbers (program COLI, the formal integral shall now account more correctly for the frequency redistribution of the scattered photons caused by their thermal motion.

Following Hummer & Mihalas (1967) – cf. also Mihalas (1978) p. 432 –, the *angle-avaraged redistribution function* for electron scattering is ... TO BE WRITTEN

Note, however, that such angle-averaging makes onlty sense in the *co-moving frame*. Therefore, in preparation for the formal integration (Eq. 360), we must first perform a full CMF radiative transfer calculation for the spectral range under consideration. This calculation is performed in the subroutine FORMCMF which is called first for each range.

A CMF frequency grid is established, spaced by 0.3 DXCMF

The continuum opacities are evaluated (subr. COOP), as are the line opacities for all alines in the current range (subr. LIOP). The line opacities are added to the total opacity, adopting gaussian profiles (i.e. only Doppler broadening) with the Doppler velocity composed of the thermal velocity (depending on atomic mass and local temperature) and the local microturbulence (if specified).

NOTE: WE SHOULD CHECK IF THIS IS TRUE. SINCE DXCMF IS MADE FOR RESOLVING VDOP FROM THE MODEL FILE, IT MIGHT BE NOT SUFFICIENT FOR RESOLVING NARROWER LINES HERE!!!

The electron-scattering contribution to the emissivity is adopted as for coherent scattering as first guess. After

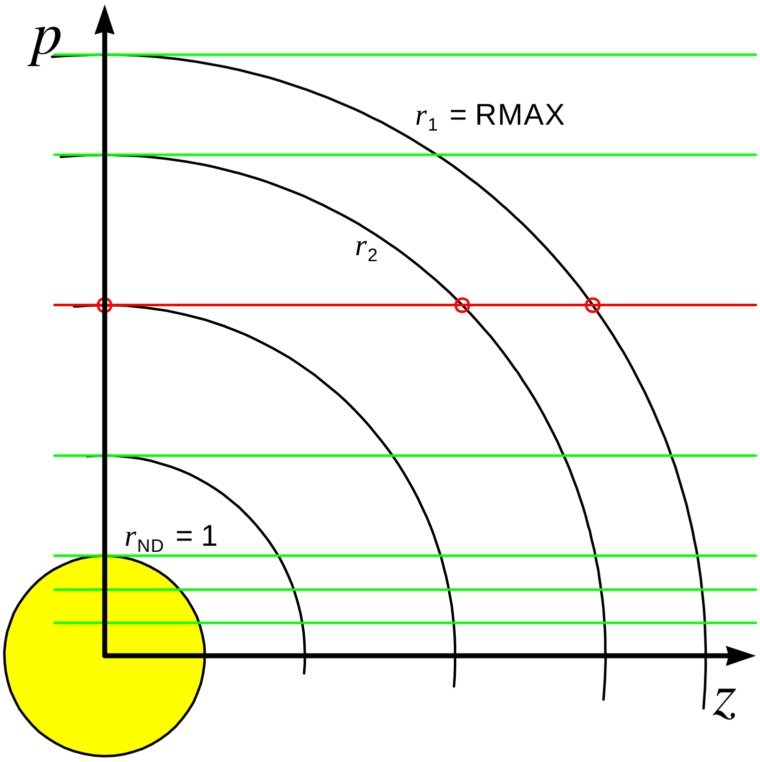


Figure 11: Integration of the emergent intensity. The observer is to the right. For each ray with impact parameter p and observer's-frame frequency ν, the integration proceeds inwards along the z coordinate. The integration starts when the ray enters the domain at $R_{\max}$ and ends either at the stellar core (if $p < 1$) or when the ray leaves the back hemisphere.

the whole frquency range is complete, the emissivity is updated, using now the electron-scattering redistribution function ...

This procedure is repeated in an iteration loop, until the maximum relative correction of η (at any frequency and depth) falls below a specified threshold $\epsilon = 0.001$.

In FORMCMF, the radiative transfer is solved ray-by-ray (i.e. angle-dependent) by a differencing scheme, which is given in detail in the Appendix of Hamann (1981).[5]

...

A starting value for the Feautrier-intensity u at the first frequency index is needed ("blue-wing boundary condition").

15.5. Line broadening

By default, the line profiles in the opacity and emissivity have Gaussian shape, i.e. they account only for thermal and microturbulent broadening (cf. specifications of VDOP and VMIC).

This is sufficient for wind lines, i.e. especially for Wolf-Rayet type spectra. Natural line broadening (radiation damping) might contribute to very thick lines, and was included in some of the older WR model calculations by specifying the VOIGT parameter on the FORMAL_CARD entries of the corresponding lines.

For O- and B-type stars, however, pressure broadening is essential. This is invoked (together with radiation damping) for *all* lines by the FORMAL_CARD option BROADENING alias ALLBROADENING.

[5] In program COLI we apply a short-characteristic integration for the same purpose; we might consider to implement the same here

Technically, the computation of line broadening proceeds in two steps:

- SUBROUTINE STARKBROAD (called for each spectral RANGE from the main program FORMAL) prepares broadening data and marks each line with a broadening keyword (vector LINPRO) that is also displayed in formal.out.
- When the formal integration is actually performed in subroutine ZONEINT, this keyword triggers for each line the evaluation of the appropriate profile function (cf. Table 2).

There are particular subroutines for the preparation of the broadening, as well as functions for the evaluation of the profile. The code distinguishes between H I lines, He I lines, He II lines, hydrogen-like ions (linear Stark effect), and other lines (quadratic Stark effect), see Table 2.

Table 2: Different kinds of line broadening

Ion	LINPRO keyword	Preparation (called from STARKBROAD)	Profile evaluation (called from ZONEINT)
H I	BRD-H	READ_H_STARKDATA FUNCTION STARK_HI_LEMKE Fallback: FUNCTION STARKHI	FUNCTION STARKPROF
He II	BRD-HeII	STARKHEIIPREP FUNCTION STARKHEII Fallback: --> L-STARK	FUNCTION STARKPROF
He I	BRD-HeI	STARKDAMP_HEI Fallback: --> Q-STARK	FUNCTION STARKVOIGT
H-like	L-STARK	LINSTARK FUNCTION KHOLTSMARK	FUNCTION STARKHOLTSMARK
Others	Q-STARK	QUADSTARK Fallback: --> VOIGT	FUNCTION STARKVOIGT
DRTR.	VOIGT		FUNCTION STARKVOIGT

The treatment differs especially between those lines for which detailed broadening tables can be used (a limited number of lines from H I and He II), and other lines for which broadening functions (Voigt, Holtzmark) are applied.

15.5.1. Radiation damping

While natural line broadening by radiation damping contributes implicitely to all broadening prescriptions, this effect remains the only broadening mechanism (in addition to thermal broadening) in case of (a) the fallback branch from Q-STARK, and (b) for transitions from auto-ionization levels (DRTRANSIT, see Sect. **??**).

Radiation damping alone leads to a Lorentz profile

$$\varphi_{\rm L} = \frac{a}{\pi} \frac{1}{(x - x_0)^2 - a^2} \tag{363}$$

where x is the dimensionless frequency in Doppler units, and x_0 its value at the line center, and a

Doppler broadening alone, caused by thermal plus microturbulent motion, leads to a Gauss profile

$$\varphi_{\rm D} = \pi^{-1/2} e^{-(x-x_0)^2} \quad . \tag{364}$$

Combination of these two broadening mechanisms correspond to a convolution of the two profile functions; the resulting *Voigt function* $H(a, x)$ cannot be written in analytic form; we use a numerical approximation from Detlef Koester (priv. comm.) in `FUNCTION VOIGTH` (see also textbooks for more details).

The Voigt parameter

$$a = \frac{\gamma}{4\pi} / \Delta\nu_{\rm D} \tag{365}$$

is depth-dependent, because the Doppler unit $\Delta\nu_{\rm D}$ is depth-dependent in any case, and the damping parameter γ can be depth-dependent as well if including pressure broadening in certain cases (see below).

Therefore we establish (in `STARKBROAD`) a two-dimensional array `AVOIGT(NBL,L)` for the Voigt parameter a, where `NBL` stands for the line index and `L` for the depth index as usual.

For He I the Voigt parameter is provided by the subroutine `STARKDAMP_HEI` for the contribution by electron impacts, and by `STARKDAMP_HEI_NEUTRAL` for collisions with neutral helium atoms. The latter contribution is probably only important for plasma that is nearly neutral (only few free electrons). In the present shape, `STARKDAMP_HEI_NEUTRAL` provides only very small damping parameters, but this might be a bug as well. Therefore, the call of `STARKDAMP_HEI_NEUTRAL` is currently commented.

For all other ions, the damping (Voigt) parameter a is provided by the subroutine `QUADSTARK`. This might not be the ultimate sophistication; there is also some inaccuracy regarding doubly excited states (see comment in `QUADSTARK`). Moreover, this subroutine is not really well-tested, since we have not yet been confronted with any spectra in which metal lines were pressure-broadened sigificantly. If necessary, this branch must be worked on.

In `FUNCTION STARKVOIGT`), the value of a is interpolated over the radius before the Voigt function `VOIGTH` is evaluated. The obtained profile function, multiplied with the line opacity, is finally used in the radiative transfer integral in `ZONEINT`.

15.5.2. Tabulated profiles: H I

For hydrogen lines, we apply tabulated profiles from Michael Lemke (available via `SIMBAD`, published in Lemke (1997) and stored for `PoWR` in the file `LEMKE_HI.DAT` (see the `FORMAL_CARDS` option `PATH_LEMKE_DAT` for its location). The Lemke tables comprise the first four line series of hydrogen, while the upper priciple quantum numbers up to 22 are covered.

This table is read by subroutine `READ_H_STARKDATA`. Subsequently, in function `STARK_HI_LEMKE`, for each radial grid point the profile is interpolated to the current temperature and electron density. The profiles in these tables account already for thermal broadening. Therefore, we convolve each requested profile only with a Gaussian for additional microturbulent broadening. For each radial point, the normalized profile is stored as a vector over frequency points in the array `PHITAB`. For each line of the current `BLEND` block that is tabulated this way in `PHITAB`, the vector IPOINTERPHITAB bookmarks its index.

For lines that are not covered by the Lemke table, the fallback is coded in function `STARKHI`. The genesis of this subroutine is obscure; following its comment lines, it has been written by Deane Peterson & Bob Kurucz.

It follows work by Griem (1960, 1967) with corrections to approximate numerical results from Vidal et al. (1973).

For those H I lines which are treated by the fallback function STARKHI, the profiles are stored in PHITAB in the same way as for the lines found in the Lemke tables. We believe that the profiles returned by STARKHI also account already for thermal broadening, and hence must be convolved only for additional microturbulence.

When the transfer equation is finally integrated along each ray in subroutine ZONEINT, the interpolation in the PHITAB tables is performed by function STARKPROF.

15.5.3. Tabulated profiles: He II

For the broadening of He II lines we use the *Vidal-Cooper-Smith-Schoening-Butler* tables published by Schoening & Butler (1989); Schoning & Butler (1989). The corresponding data file is called VCSSB.DAT (see the FORMAL_CARDS option PATH_VCSSB for its location). The source of this data file is not known anymore.

Unfortunately, these tables cover only 18 lines (principle quantum number transitions 2 – 3; 3 – 4...3 – 10; 4 – 5...4 – 15). For all other He II transitions, we use L-STARK as fallback.

The tables in VCSSB.DAT are extracted for each requested line by the subroutine STARKHEIIPREP. The function STARKHEII then interpolates in the extracted table with respect to the temperature and density at each radial point, and returns the respective profiles as vector over the frequency points in the PHITAB array.

In contrast to the H I profiles (see above), we believe that the VCSSB tables do not account yet for thermal broadening. Therefore, they are convolved with a Gaussian of width the depth-dependent Doppler width DD_VDOPDU which includes both, thermal broadening plus microturbulence.

As for H I, the evaluation of those profiles that have been prepared in the array PHITAB is performed by function STARKPROF when the transfer equation is finally integrated along each ray in subroutine ZONEINT.

15.5.4. He I

For the lines of neutral helium, Voigt profiles are assumed. The damping "constant" GAMMAHE1 for its Lorentian part is calculated in subroutine STARKDAMP_HEI for each requested line at each radial grid point. The source of this subroutine is not known. The theory follows mainly Griem et al. (1962) and further sources mentioned in the subroutine's comment lines. Wavelength shifts of the line centers are also calculated, but neglected in the rest of the PoWR code.

The subroutine STARKDAMP_HEI has hard-coded coefficients for only 16 lines of He I, all in the optical range. For transitions not covered here, STARKDAMP_HEI sets the broadening keyword to Q-STARK as fallback.

For extremely cool atmospheres, collisions with neutral atoms might contribute to pressure broadening. This shall be treated by the subroutine STARKDAMP_HEI_NEUTRAL which, however, still needs debugging and, therefore, is commented out.

The damping constant GAMMAHE1 is finally converted into the Voigt parameter a (array AVOIGT) and added to the radiation damping. The latter has been calculated already before in subroutine PREFORM from the inverse lifetimes of both involved levels (= sum of Einstein coefficients) and stored in AVOIGT.

When the transfer equation is finally integrated along each ray in subroutine ZONEINT, the evaluation of the Voigt function is performed by function STARKVOIGT.

15.5.5. Linear Stark effect

This effect applies for all hydrogen-like ions (i.e. with one electron in the outermost shell), if not already covered by tabulated broadening (H I, He II).

The preparation is done in subroutine LINSTARK which was written by Andreas Sander following the approach that is described in the Appendix B of Hubeny et al. (1994) as used in his code TLUSTY. Part of the physics is outsourced to the function KHOLTSMARK, which relies on Griem (1960). The result of LINSTARK is the parameter GRIEMPAR for each line at each depth point.

When the transfer equation is finally integrated along each ray in subroutine ZONEINT, the evaluation of the profile function is performed by function STARKHOLTSMARK. Here, the GRIEMPAR parameter is interpolated before calling the function PHIHOLTSMARK, which combines a Doppler line-core with asymptotic Holtsmark wings, again following Hubeny et al. (1994).

The L-STARK formalism requires knowledge of the principal quantum number. If this number is missing in the DATOM file, the corresponding line is calculated with Q-STARK and a corresponding warning is issued.

15.5.6. Quadratic Stark effect

The formalism in QUADSTARK follows Cowley (1971), and leads to Voigt profiles. The damping parameter Γ_{quad} is calculated from an “effective quantum number”. Since the latter diverges for upper-level energies at the ionization threshold, the quadratic Stark effect is not calculated for level energies close to or above the ionization threshold. Fallback in this case is LINPRO='VOIGT', i.e. only natural line broadening (radiatoon damping) is accounted for.

15.5.7. Bandwidth estimate

For the linebroadening it is essential to make the frequency bands for each line much wider than the few Doppler width which are sufficient for pure Gaussian profiles. Unfortunaley, we cannot allow different bandwidths for the individual lines; our administration of lines that are *active* at a given wavelength is based on the wavelength-sorted linelist: the *active* lines form a compact section in this list, and they are *checked-in* and *checked-out* when the calculation proceeds through the spectrum. Hence, we must apply the *largest* of all needed bandwidths, XMAX to all lines. Note: A more flexible administration with individual bandwidths could save substantial computing time → future work.

The largest bandwidth needed, XMAX, is establshed by SUBROUTINE BANDWIDTH. For a fixed value of the optical depth TAUCRIT we integrate the line opacity over radius as if the atmosphere was *static*, for each of the different frequencies from the line center to the wings, till the obtained optical depth falls below the TAUCRIT threshold. This wavelength difference to the line center (in Doppler units) defines XMAX for the tested line, and finally we take XMAX as the largest from all lines in the line list for the calculated RANGE alias BLEND complex.

Obviously, the parameter TAUCRIT should not be choosen too small for efficiency reasons. Since the opacity is integrated through the whole atmosphere, it can be compared to the Rosselend (continuum-only) Rosseland optical depth TAURCONT, which is usually chosen to be 20 at the inner boundary (TAUMAX=20. FIX ...). For instance, with chosing TAUCRIT=0.1, the line wing (of the broadest line) is followed till the optical depth falls below 0.5% of the (mean) continuum opacity – which should be sufficient.

The default value of TAUCRIT, here named TAUMINBROAD, is set in formal.f to 0.0001 (0.1 before Jan 2017, 0.01 before Mar 2025). This default can be overwritten by the option
TAUBROAD = x.x
in the FORMAL_CARDS.

15.6. Wind Rotation

This formalism is described in the Master Thesis by Tomer Shenar and, in less detail, in Shenar et al. (2014). The basic assumptions are:

- For a rigidly rotating sphere, the loci of constant line-of-sight projected velocity are lines parallel to the rotation axis (red stripes in Fig. 12), and these projected velocities depend only on the *product* of the rotational velocity at the eqator, $v_{\rm eq}$, and the sinus of the inclination angle i: $v_{\rm eq} \sin i$ = `VSINI` – for an elegant proof, see the book by Unsoeld (1968) on *Physik der Sternatmosphären*.
- Each radial shell rotates as if it were rigid, but with its own specific angular velocity $\omega(r)$. This assumption implies the same parameter degeneracy as known from non-extended atmospheres, namely that the effect depends only on the *product* $v_{\rm eq} \sin i$ = `VSINI`, holds here as well.
- We assume rigid rotation up to some co-rotation radius `RCOROT`. For radii larget than `RCOROT`, we assume conservation of angular momentum in the equatorial plane. The same angular velocity is then adopted for *all* latitudes, i.e. each radial shell rotates rigidly.

  Thus, the angular velocity is

  $\omega(r)$ = const. for $r <$ `RCOROT`
  $\omega(r) \propto 1/r^2$ for $r >$ `RCOROT`

In other words, we assume

- cylindrical rotation (no velocity component in polar direction
- the azimuthal velocity component (v_φ) is $\sin(\theta) \cdot f(r)$.
  The function $f(r)$ depends on the radial regime:
  $f(r) \propto r$ if $r <$ `RCOROT` (rigid-body rotation)
  $f(r) \propto 1/r$ if $r >$ `RCOROT` (angular momentum conservation)

The rotational component of the velocity field is therefore defined by specifying two parameters, `VSINI` and `RCOROT` – cf. `FORMAL_CARDS` options in Sect. 9.

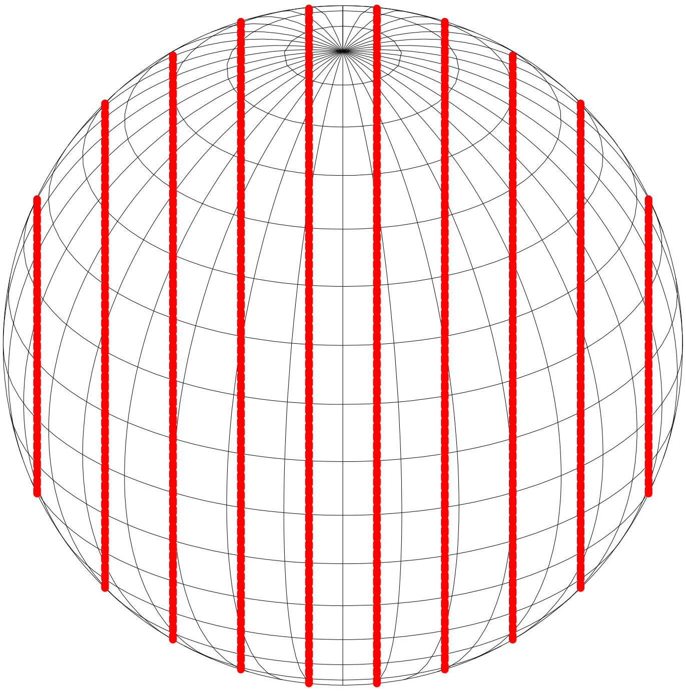


Figure 12: Rotating sphere, projected into the plane of the sky (x, y). The projected rotation axis is parallel to the y-axis. For a rigidly rotating sphere, the velocity component in the direction to the observer is constant along lines parallel to the y-axis, and depends only on $v \sin i$. The same holds for our model of a rotating wind, since each radial shell is assumed to rotate rigidly.

Thus, the velocity vector has now two components:

- radial component with velocity law $v(r)$
- azimutal component due to rotation with v_φ

This breaks the rotational symmetry of intensities over the stellar disk; instead, the flux integral becomes now 2-D (over p and over φ) – see Fig. 9 :

$$F_\nu = \frac{1}{R_{\max}^2} \int_{\varphi=0}^{2\pi} \int_{p=0}^{R_{\max}} I_\nu^+(p,\varphi)\, p \,\mathrm{d}p \,\mathrm{d}\varphi$$

15.7. Combining two models

This version of the formal integral is invoked by the option SECONDMODEL in the FORMAL_CARDS (see Sect. 9). Concept is the combination of two models – the MODEL of the current chain in wrdata*n*, and a previously calculated model that has been saved in the directory specified by the SECONDMODEL PATH option.

The formal integration now takes the population numbers, density and velocity from the *second model* in a specified part of atmosphere's volume, and from the current MODEL everywhere else.

15.7.1. Cone model

In this version, the *second model* is applied within a double-cone with (half) opening angle THETA=θ. The inclination angle between the cone axis and the line-of-sight is INCLINATION=i, where $i = 90°$ means the the cone lies in the plane of the sky (cf. Sect. 9).

The combination of inclination and opening angle is slightly restricted: i and θ are not allowed to be equal, but must differ by more then $0.1°$. Note that in case $i > \theta$ the observer sees the double-cone from the side, while if $i < \theta$ the observer looks *into* the opening of the cone.

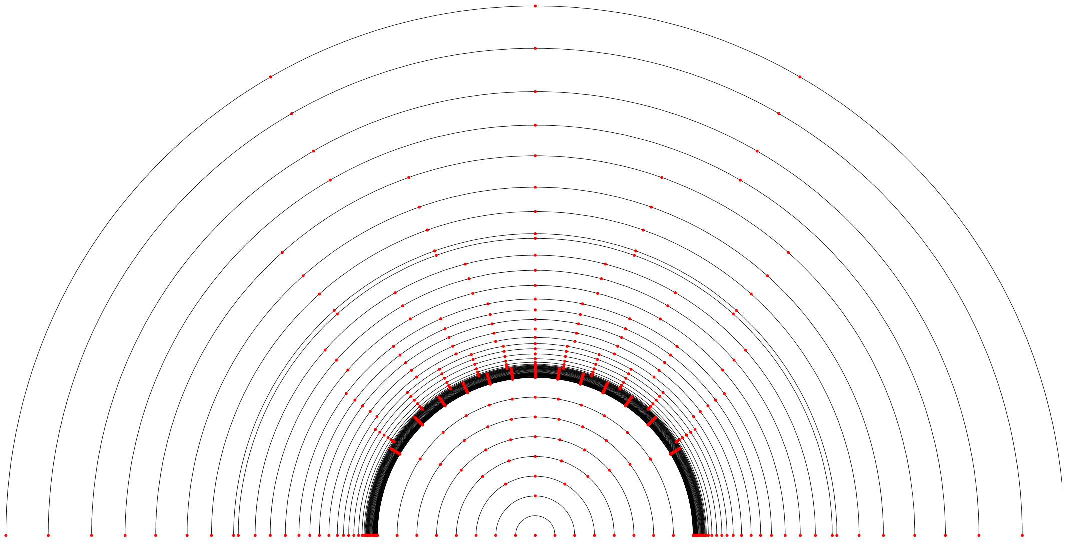


Figure 13: Distribution of rays in the plane of the sky for a rotating wind. In order to assure sufficient spectral resolution and thus avoiding numerical artefacts, the number of impact-parameter points with $p < 1$ ("core rays") is automatically increased. Azimuth-angle points φ_i are automatically defined in each impact-parameter circle. Note that wind rotation does not break the symmetry between the upper and the lower hemisphere.

The two-model option can be combined with the wind-rotation option (cf. Sect. 15.6). In case of cone geometry, the cone axis is always assumed to lie in the (y, z) plane, and there is no option to turn the cone to the side. If *not* combined with wind rotation, this is not restricting generality, but with rotation it is since the rotation axis also lies in the same plane. Albeit not being general, this complies with the plausible assumption that the cone axis and the rotation axis are identical.

The plane of the sky is spanned by the axes (x, y), and the observer looks along the z-axis from $z = \infty$. Now it is necessary to distinguish between two cases, depending on whether the inclination angle i is greater or smaller than the opening angle θ:

1. **Ellipse case**

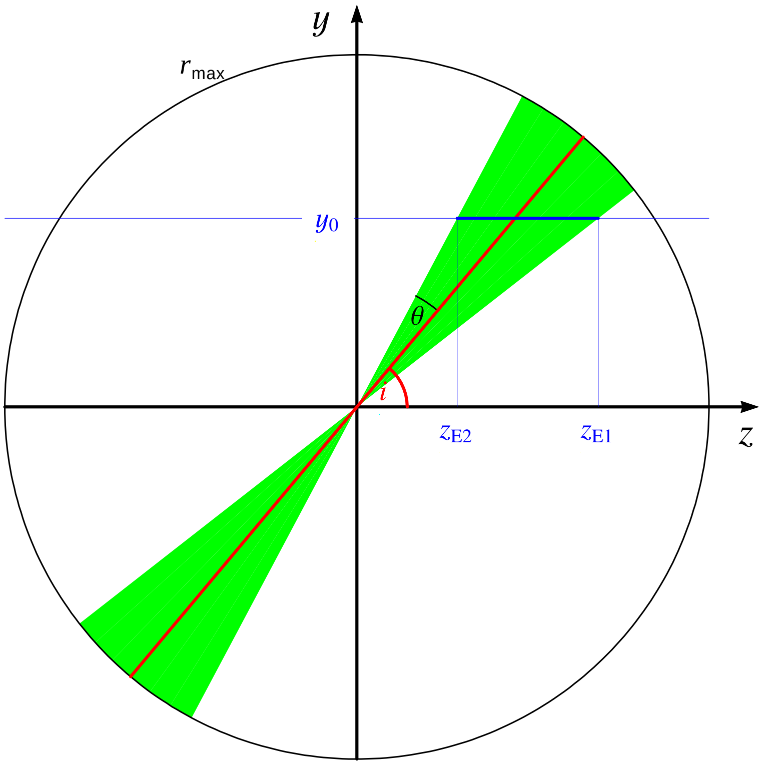


Figure 14: Cone geometry for the combination of two models: cut through the plane at $x = 0$ (note: the cone axis lies in this plane). The observer looks from right ($z \rightarrow \infty$). The cone is inclined by the angle i and has a half opening angle θ. The intersection with the plane $y = y_0$ gives an ellipse with major axis from z_{E1} to z_{E1} (thick blue line).

As obvious from Fig. 14, if $i > \theta$ the observer looks at the cone from the side. Any line-of-sight can cut through the cone only once, entering the cone domain at some itersection point $z_1 < z_{E1}$ and leaving at $z_2 > z_{E2}$. We have $y_0/z_{E1} = \sin(i - \theta) \,/\, \cos(i - \theta)$, and thus

$$z_{E1} = y_0 \; \cot(i - \theta) \tag{366}$$

$$z_{E2} = y_0 \; \cot(i + \theta) \tag{367}$$

Now we consider a cut through the cone in the plane $y = y_0$ (Fig. 15), which gives an ellipse. Its major axis lies on the x-axis; the center of the ellipse is at

$$z_M = \frac{1}{2}(z_{E1} + z_{E2}) \tag{368}$$

The major semi-axis has the length

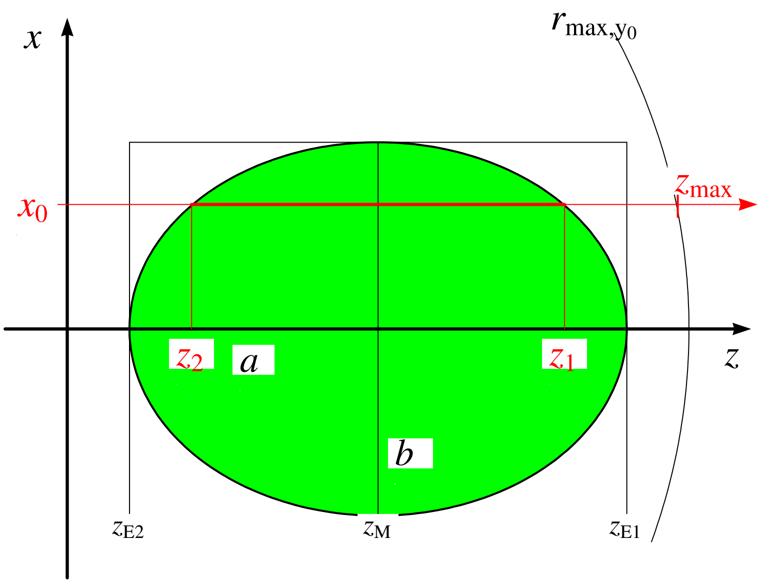


Figure 15: Cone geometry for the combination of two models: cut through the plane at $y = y_0$. The green ellipse is the cut with the inclined cone. The observer looks from right ($z \to \infty$). One specific line-of-sight at $x = x_0$ (red line) cuts the ellipse at z_1 and z_2.

$$a = \frac{1}{2}(z_{E1} - z_{E2}) \tag{369}$$

The minor semi-axis can be calculated from the opening angle and the inclination of the cone. For that we need the numeric eccentricity, which is defined by

$$\epsilon = \frac{\cos(i)}{\cos(\theta)}. \tag{370}$$

The numeric eccentricity is related to the linear eccentrizity by $\epsilon = e/a$. The minor semi-axis is defined through $a^2 = e^2 + b^2$, so we get for the minor semi-axis

$$b = a\sqrt{1 - \epsilon^2}. \tag{371}$$

The equation of the ellipse finally reads

$$\frac{(z - z_M)^2}{a^2} + \frac{x^2}{b^2} = 1 \tag{372}$$

In order to calculate the intersection points z_1 and z_2 with the line-of-sight at $x = x_0$, we have to insert this value into Eq. (372) and solve this quadratic equation. Sorting the terms for powers of z gives:

$$(z - z_M)^2 + \frac{a^2\, x_0^2}{b^2} - a^2 = 0 \tag{373}$$

which results in

$$z^2 - 2z_M\, z + z_M^2 + \frac{a^2\, x_0^2}{b^2} - a^2 = 0 \tag{374}$$

The standard form $z^2 + pz + q = 0$ has the solutions $z_{1,2} = -\frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}$. Here we have

$$p = -2\, z_{\rm M} \tag{375}$$

$$q = z_{\rm M}^2 + \frac{a^2\, x_0^2}{b^2} - a^2 \tag{376}$$

Hence the two solutions are

$$z_{1,2} = z_{\rm M} \pm \sqrt{\texttt{term2}} \tag{377}$$

with

$$\texttt{term2} = z_{\rm M}^2 - \left(z_{\rm M}^2 + \frac{a^2\, x_0^2}{b^2} - a^2 \right) \tag{378}$$

$$= a^2 - \frac{a^2\, x_0^2}{b^2} \tag{379}$$

$$= a^2 \; - \frac{x_0^2}{1-\epsilon^2} \tag{380}$$

z_1 may not lie outside of the circle with radius $r_{\rm max, y_0} = \sqrt{r_{\rm max}^2 - y_0^2}$ (cf. Fig. 15). Hence, the maximum value of z is

$$z_{\rm max} = \sqrt{r_{\rm max}^2 - x_0^2 - y_0^2} \tag{381}$$

and thus

$$z_1 = \min[z_1, z_{\rm max}] \tag{382}$$

$$z_2 = \max[z_2, -z_{\rm max}] \tag{383}$$

Note that there is no solution for $|x_0| > b$. Moreover, there is also no solution *inside* the sphere with radius $r_{\rm max}$ if $z_2 > z_{\rm max}$, and also not if $z_1 < -z_{\rm max}$.

2. **Hyperbola case**

If $i < \theta$ the observer looks *into* the cone. In priciple, any line-of-sight starts from *inside* the cone domain, leaves it at z_2, and enters the second part of the double-cone at z_3 and stays there till infinity (cf. Fig. 16). Of course, the computed domain is restricted to the sphere with radius $r_{\rm max}$, and the region behind the stellar core is obscured.

$Z_{\rm E1,\,E2}$, $Z_{\rm M}$ and a are the same as in the ellipse case (see previous item). However, the intersection line of the cone with the $x-z$-plane any given y_0 now gives a *hyperbola* with two branches (cf. Fig. 17).

A hyperbola is described by the equation

$$\frac{z^2}{a^2} - \frac{x^2}{b^2} = 1. \tag{384}$$

It is easy to see that the major axis is the same as the major axis of the ellipse. The semi axis is not the same as the ellipcal semi axes. The semi axis b is perpendicular from the apex to the asymptoic slope of the hyperbola. So we have again the linear eccentrizity

$$e^2 = b^2 + a^2. \tag{385}$$

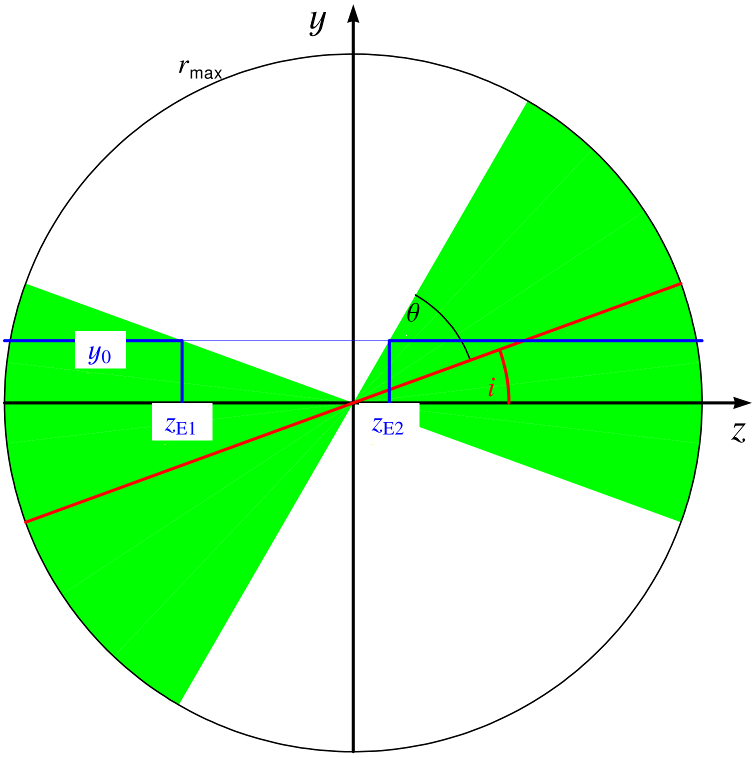


Figure 16: Cone geometry for the combination of two models: cut through the plane at $x = 0$ (note: the cone axis lies in this plane). The observer looks from right ($z \rightarrow \infty$). In the shown case, the cone inclination i is smaller than the half opening angle θ, i.e. the observar looks *into* the cone.

Hence, we get with the numerical eccentrizity Eq. (370)

$$b = a\sqrt{\epsilon^2 - 1} \tag{386}$$

As following below, we get

$$\frac{(z - z_{\mathrm{M}})^2}{a^2} - \frac{x^2}{b^2} = 1. \tag{387}$$

To calculate the intersection points z_2 and z_3 with the line-of-sight at $x = x_0$, we have to insert this value into the Eq. (387) and solve this quadratic equation with the reduced quadratic equation. Hence we get

$$z^2 - 2zz_{\mathrm{M}} + z_{\mathrm{M}}^2 - \frac{x_0^2}{b^2}a^2 - a^2 = 0. \tag{388}$$

The Variables for the reduced quadratic equation are

$$p = -2\, z_{\mathrm{M}} \tag{389}$$

$$q = z_{\mathrm{M}}^2 - \frac{x_0^2}{b^2}a^2 - a^2. \tag{390}$$

Hence the two solutions are

$$z_{2,3} = z_{\mathrm{M}} \pm \sqrt{\mathtt{term3}} \tag{391}$$

with

$$\begin{aligned} \mathtt{term3} &= z_{\mathrm{M}}^2 - z_{\mathrm{M}}^2 + \frac{x_0^2}{b^2}a^2 + a^2 && (392) \\ &= a^2 + \frac{x_0^2}{\epsilon^2 - 1} && (393) \end{aligned}$$

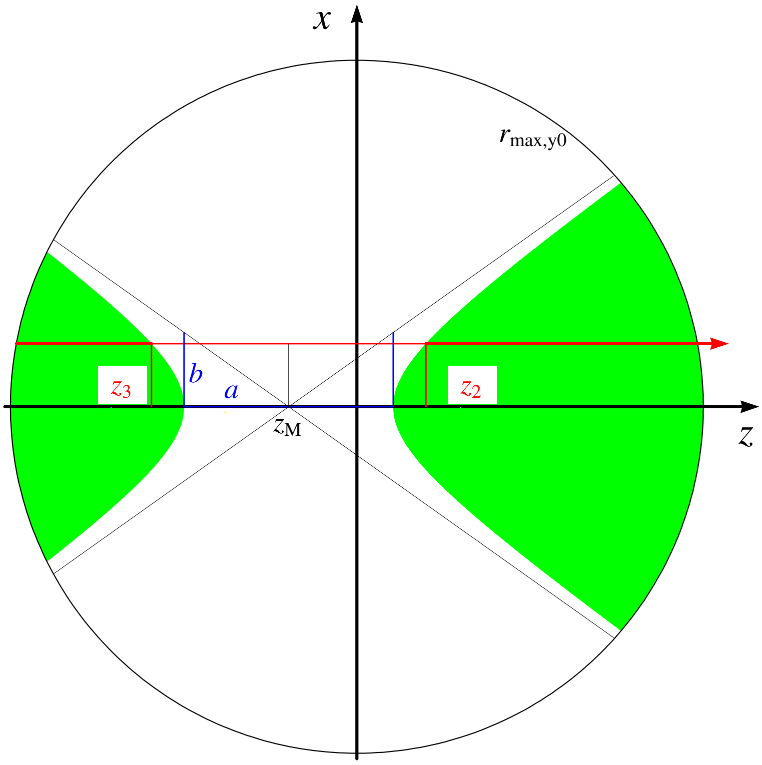


Figure 17: Cone geometry for the combination of two models: cut through the plane at $y = y_0$. The green hyperbolas are the cut with the inclined cone. The observer looks from right ($z \rightarrow \infty$). One specific line-of-sight at $x = x_0$ (red line) leaves the cone domain at z_2 and enters the second part of the double-cone again at z_3.

The intervals (z_1, z_2) as well as (z_1, z_2) are clipped at the outer radius of the stellar atmosphere, $r_{\rm max}$. For a selected ray, $z_{\rm max,y0}$ is defined by Eq. (381). As can be seen in Fig. 17), we have now two intervals, which are defined by the maximum of the radius of the star, the intersection or and the core radius. Hence we get

$$\begin{aligned} z_1 &= z_{\rm max} \\ z_2 &= \min[z_2, z_{\rm max}] \\ z_3 &= \max[z_3, -z_{\rm max}] \\ z_4 &= -z_{\rm max} \end{aligned}$$

15.7.2. Sphere model

In this version, the *second model* is applied within a sphere that is located in the wind (or at least partially overlapping). The motivation of such model is to describe the Stromgren sphere of an embedded X-ray source in a High-Mass X-ray Binary.

The plane of the sky is spanned by the axes (x, y), and the observer looks along the z-axis from $z = \infty$. The sphere and its location is specified by four mandatory parameters (cf. Sect. 9):

- RSPHERE=$r_{\rm sph}$, the radius of the sphere;
- DSPHERE=$d_{\rm sph}$, the radial distance of the sphere's center from the origen;

- `DELTASPHERE`=$\delta_{\rm sph}$, the sphere-center's elevation angle above the (x, z) plane (cf. Fig. 18);
- `ALPHASPHERE`=$\alpha_{\rm sph}$, the angle between the direction from the origin to the sphere center and the (y, z) plane (cf. Fig. 19).

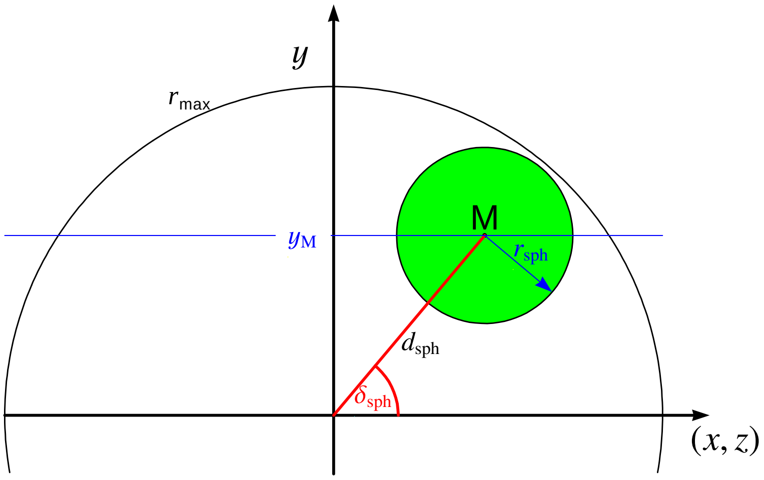


Figure 18: Sphere geometry for the combination of two models: cut through the plane containing the y-axis and the sphere center `M`.

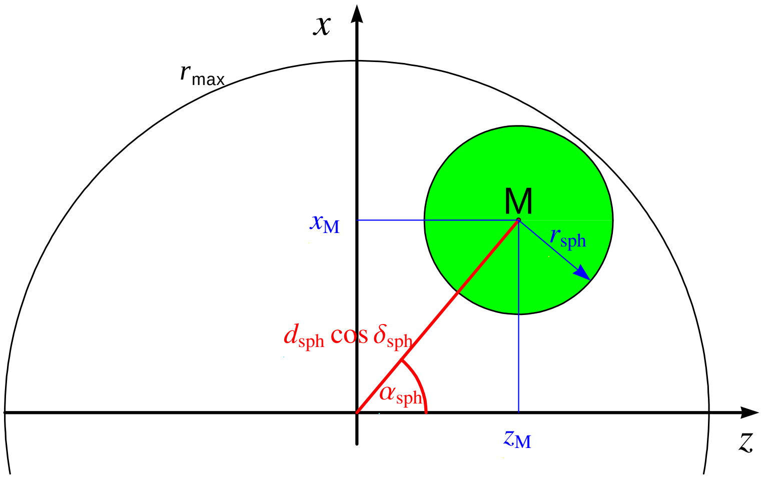


Figure 19: Sphere geometry for the combination of two models: projection into the (x, z)-plane . If this geometrical configuration rotates around the y-axis, $\alpha_{\rm sph}$ grows with the phase.

As obvious from Figs. 18 and 19, the coordinates of the sphere center $(x_{\rm M}, y_{\rm M}, z_{\rm M})$ follow from the input parameters as

$$x_{\rm M} = d_{\rm sph}\,\cos\delta_{\rm sph}\,\sin\alpha_{\rm sph} \tag{394}$$

$$y_{\rm M} = d_{\rm sph}\,\sin\delta_{\rm sph} \tag{395}$$

$$z_{\rm M} = d_{\rm sph}\,\cos\delta_{\rm sph}\,\cos\alpha_{\rm sph} \tag{396}$$

The two-model sphere geometry can be combined with the wind-rotation option (cf. Sect. 15.6). Since the sphere center can be freely located, there is no restriction of generality.

With the above center coordinates, the equation of the sphere reads

$$(x - x_\mathrm{M})^2 + (y - y_\mathrm{M})^2 + (z - z_\mathrm{M})^2 = r^2_\mathrm{sph} \tag{397}$$

Thus, for given ray with (x_0, y_0) we obtain a quadratic equation for the intersection points $z_{1,2}$:

$$(z_{1,2} - z_\mathrm{M})^2 + (x_0 - x_\mathrm{M})^2 + (y_0 - y_\mathrm{M})^2 - r^2_\mathrm{sph} = 0 \tag{398}$$

yielding

$$z^2_{1,2} - 2\, z_{1,2}\, z_\mathrm{M} + z^2_\mathrm{M} + (x_0 - x_\mathrm{M})^2 + (y_0 - y_\mathrm{M})^2 - r^2_\mathrm{sph} = 0 \tag{399}$$

The standard form $z^2 + pz + q = 0$ has the solutions $z_{1,2} = -\frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}$. Here we have

$$\begin{aligned} p &= -2\, z_\mathrm{M} & (400)\\ q &= z^2_\mathrm{M} + (x_0 - x_\mathrm{M})^2 + (y_0 - y_\mathrm{M})^2 - r^2_\mathrm{sph} & (401)\end{aligned}$$

Hence the two solutions are

$$z_{1,2} = z_\mathrm{M} \pm \sqrt{\mathtt{term2}} \tag{402}$$

with

$$\mathtt{term2} = r^2_\mathrm{sph} - (x_0 - x_\mathrm{M})^2 - (y_0 - y_\mathrm{M})^2 \tag{403}$$

Obviously, there is no intersection if $r^2_\mathrm{sph} - (x_0 - x_\mathrm{M})^2 - (y_0 - y_\mathrm{M})^2 < 0$
Moreover, the intersection points may not lie outside of the atmosphere domain (cf. Eq. 381), i.e.

$$z_\mathrm{max} = \sqrt{r^2_\mathrm{max} - x^2_0 - y^2_0} \tag{404}$$

and thus

$$\begin{aligned} z_1 &= \min[z_1, z_\mathrm{max}] & (405)\\ z_2 &= \max[z_2, -z_\mathrm{max}] & (406)\end{aligned}$$

Note that there is also no solution *inside* the sphere with radius r_max if $z_2 > z_\mathrm{max}$, and also not if $z_1 < -z_\mathrm{max}$.

15.7.3. Visualization

The *secondmodel* geometry is visualized by two plots, which are both kept in the scratch directory in which the *formal* job has been executed, e.g. `$USER/work/scratch/formal`*n*:

- `secondmodel.plot` is a WRplot script showing the plane of the sky as seen from the observer (cf. Figs. 20, 21).
- `secondmodel.dat` is a dataset that contains 3-D coordinates, and is to be viewed with the `gnuplot`-script `gnuplot ~wrh/proc.dir/view-secondmodel.gplt` (cf. Fig. 22). In the window opened by `gnuplot`, the viewing direction can be turned with the mouse.

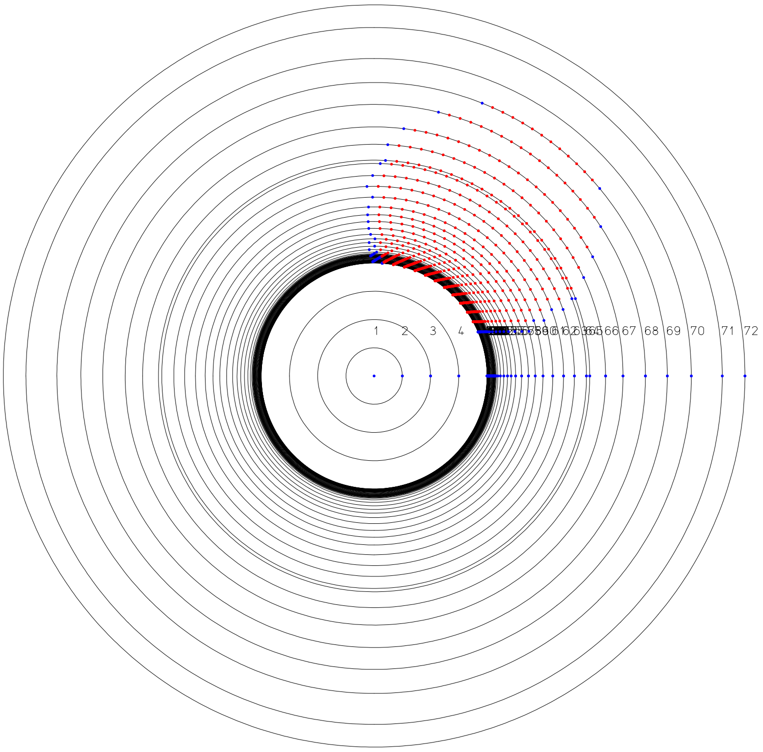


Figure 20: Positions of the rays for which the emergent intensities are calculated. Shown is the plane of the sky with spherical coordinates p (impact parameter) and azimuth-angle φ. Red dots mark rays which intersect with the second-model domain, while blue dots mark those which don't. The concentric circles correspond to the impact-parameter grid. The red dots are concentrated in the domain of the *second model* according to its specified geometry. In this example, the second-model domain is specified as SHAPE=SPHERE in a distance of DSPHERE=2 from the center and a radius of RSPHERE=1. The position angle is chosen as DELTA=45 degrees. The angle ALPHA=135 degrees is larger than 90°, i.e. the second-model domain lies in the back hemisphere and therefore is partially obscured bu the stellar disk. For better representation, the wind was cut off beyond $3.5\,R_*$ with NOWIND RADIUS=3.5 as option in FORMAL_CARDS.

15.7.4. Implementation details

In the main program FORMAL, the MODEL file is read by the subroutine FORMOSA. In case of requesting a second model, FORMOSA is called a second time, and all model quantities are stored in arrays with an additional dimension for the model index IMOD=2.

Each of the two models have their own radius grid, which is optimized for the respective model. The CMF radiative transfer calculation is performed by subroutine FORMCMF for each of the models on its respective grid, and provides the contribution to the emissivity due to electron-scattering redistribution.

In order to combine two models for formal intergration, the opacities and emissivities now must be provided on a common geometrical grid. Here it turned out to be important that the "merged" grid is suitable for both models to be combined. The subroutine MERGE_RGRID combines the radius points such that their density follows everywhere the higher density of both original grids. This procedure leads to a moderate increase in the number of depth points (as reported in the cpr file). Subsequently, also the impact-parameter grid and the z-grid is established according to the new, merged radius grid. – Note that the "core rays", i.e. the impact-parameter points, might have been increased already before because of wind rotation if requested.

With the merged radius grid established, all opacities, emissivities, nd some further arrays are interpolated onto

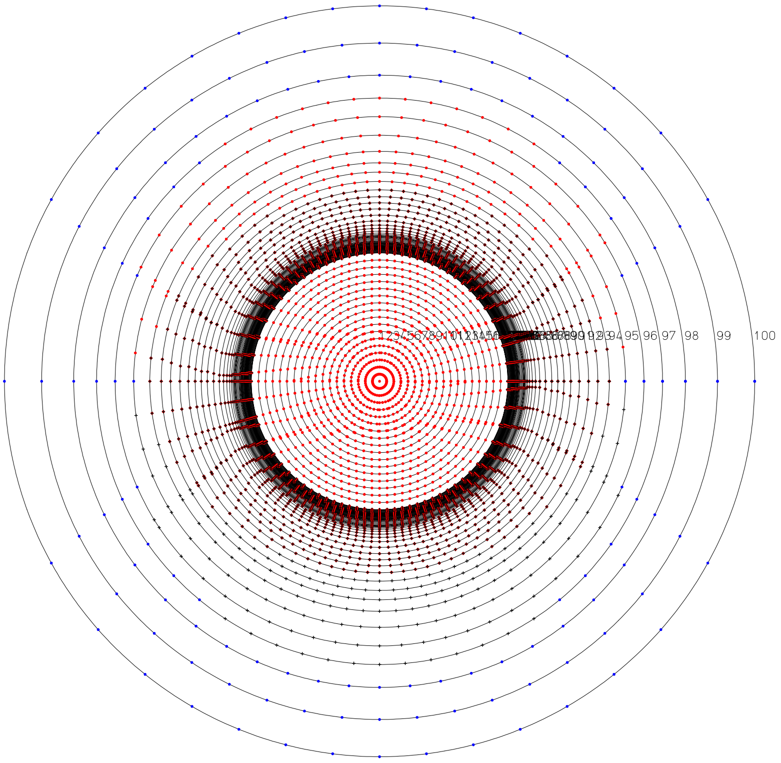


Figure 21: Same as Fig. 20, but in this example the second-model domain is specified as `SHAPE=CONE` with an inclination angle `CONEI=10` and opening half-angle `THETA=40` [degrees]. In the cone case with $i < \theta$ (the "hyperbola case"), the red dots mark rays which intersect with the front part of the double-cone, while black crosses denote rays intersecting with the back part; red dots with croses on top are rays which cut through both parts. Moreover, the calculation includes wind-rotation with `VSINI=300` [km/s] which causes additional impact-parameter and angle-points (cf. Fig. 13). For better representation, the wind was cut off beyond $3.0\,R_*$ with `NOWIND RADIUS=3.0` as option in `FORMAL_CARDS`.

that grid. The intersection points of all rays with the domain of the second model are calculated as described in Sect. 15.7. Then the integration along each ray is performed for each frequency, taking the opacities, emissivities and line profiles from the 1st or the 2nd model, respectively, acording to the domain where the current integration step is located.

The inclusion of a second model breaks the rotational symmetry (as does the wind rotation if requested). Hence, the intensities must now be calculated for several azimuthal angles φ (cf. Fig. 20 or Fig. 21) and finally integrated over those angles. *Hence, the computational effort increases significantly, possibly by a factor of ~20!* (The average number of φ angles is reported in the cpr-file.) Therefore, one should consider to restrict such calculations to the necessary wavelenth range(s).

The use of a `SECONDMODEL` is prepared in `SUBROUTINE SECONDMODEL_PREP`. For each ray (characterized by the impact-parameter index `JP` and the phi-point index `LPHI`) the points where the ray enters or exits the second-model domain (i.e. z_1, z_2, z_3, z_4) are calculated and stored in the array `ZINTER(i, JP, LPHI)`, where the first index is 1, 2, 3, or 4. If a ray does not enter the second-model domain, $z_1 = z_2 = .0$ or $z_1 = z_2 = .0$, respectively. When the integration along a specific ray is finally performed in `SUBROUTINE ZONEINT`, it is checked at each (fine) integration step: if the quadrature point lies in one of the intervalls (z_1, z_2) or (z_3, z_4), this point belongs to the `SECONDMODEL` domain.

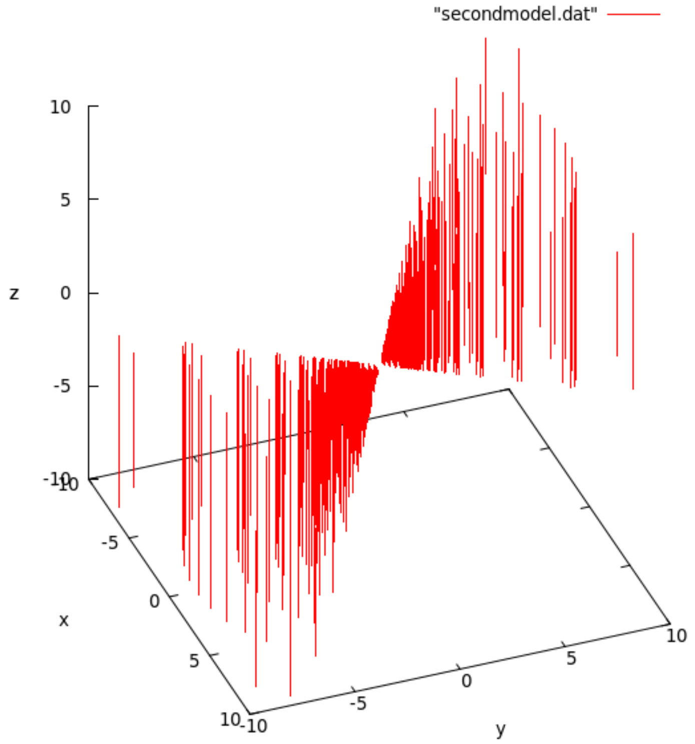


Figure 22: Example of `gnuplot ~/proc.dir/view-secondmodel.gplt`, which visualizes those parts of the rays that fall into the domain of the *second model*.

15.8. Program Structure

The input file `FORMAL_CARDS` has a double nature. Various commands specify some setting (e.g. `ALLBROADENING`); such settings overwrite their default and stay valid until being revised. In the program, these input lines are sequentially decoded by the subroutine `DECFORM`.

The bulk of lines in `FORMAL_CARDS`, however, contain atomic data. Usually, these data have been assembled with the help of the `newformal_cards` tool (see Sect. 6.2). For each spectral range (specified by a `RANGE` command in `NEWFORMAL_CARDS_INPUT`), the atomic data for all spectral lines in that range are bracketed between `BLEND` and `-BLEND`.

When subroutine `DECFORM` encounters the beginning of such `BLEND` block, reading continues by subroutine `PREFORM`. Here, the `LINE` and `MULTIPLET` data are decoded, the latter with the help by subroutine `MULTIPLE` which eventuelly appends additional `SUBLEVEL` population numbers to the `POPNUM` array.

After the `-BLEND` mark has been reached, the code returns to the main program `FORMAL` and performs the spectrum synthesis calculations for that range, especially creating the output and plot files for that range.

Subsequently, `DECFORM` continues to read the input from `FORMAL_CARDS`, eventually encountering another `BLEND` block, and so one. Thus, the program `FORMAL` works progressively through the spectral ranges.

```
1 DATOM
 2 FEDAT

1 FORMOSA
1 POPMIN_NULLING
1 PRI_PAR
```

```
1 DECFORM

1 VDOP_STRUCT

1 ROTATION_PREP
1 PLOT_WINDROT_GRID

1 SECONDMODEL_PREP
 2 PLOT_SECONDMODEL_GRID
1 COPY_SECONDMODEL
1 PREPMACROCLUMP
1 MANIPOP
1 SET_POP_ZERO

1 PREFORM
1 LIOP

1 COOP
1 PRIOPAL

1 FORMCMF
 2 CMFFEOP
 2 COOP
 2 ELIMIN
 2 BACKJC
 2 CONVOLOPAFE

1 RESCALE_SECMOD

1 STARKBROAD
1 LIMB_INFO
1 PRIDWL

1 PREPRAY
1 OBSFRAM
 2 ZONEINT
1 TRAPLO
1 TRADWL
1 PRIPRO
1 PLOTVDOP
1 PLOTLIMB
```

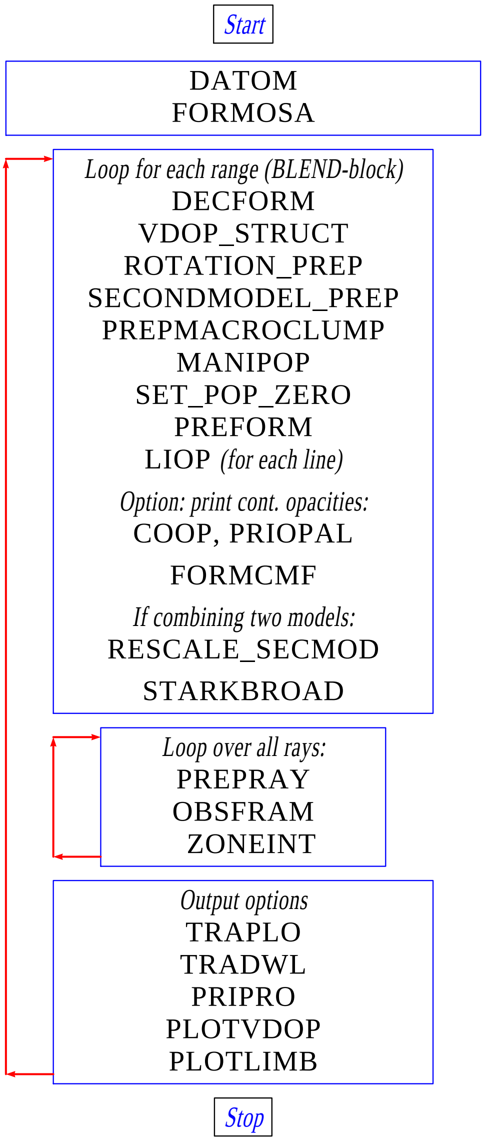


Figure 23: Flowchart of the program `FORMAL`

16. Atomic transitions

Atomic data enter the radiative transfer (via opacities and emissivities) and the rate equations (via the rate matrix). The atomic data are taken from the followind input files: `DATOM` (see Sect. 2.4), `FEDAT` (for iron-group superlevel atoms, see Sect. 16.8.1), and `FORMAL_CARDS` (see Sect. 9).

In the followowing sections we describe the different atmoc transition processes between explicitely treated non-LTE levels. The superlevel approach as employed for the "generic" iron-group elemement is outlines in Sect. 16.8.1.

16.1. Collisional bound-bound transitions

16.1.1. Introduction

Collisional transition rates R_{ij} depend on the collisional rate coefficiens C_{ij}:

$$R_{lu} = n_l \, C_{lu} \quad \text{and} \quad R_{ul} = n_u \, C_{ul} \tag{407}$$

PoWR only accounts for collisions with electrons. Therefore,

$$C_{ij} = n_{\mathrm{e}} \, \Omega_{ij} \tag{408}$$

where Ω_{lu} is called the *collision strength*.

As the electrons are assumed to have a Maxwellian distribution, collisions alone would establish LTE. In LTE, *detailed balance* must hold; therefore,

$$\Omega_{lu} = \Omega_{ul} \left(\frac{n_u}{n_l} \right)^* \tag{409}$$

In this notation (from Mihalas), $\left(\frac{n_u}{n_l}\right)^* = \left(\frac{g_u}{g_l}\right) e^{-\frac{E}{kT}}$ denotes the LTE population number ratio, i.e. the Boltzmann factor.

The number of collisions between an electron and an ion depends on the electron's speed v and the ion's cross section Q_{lu}. The Maxwellian distribution of the velocities is

$$w(v) = \frac{4}{\sqrt{\pi} \, v_{\mathrm{th}}^3} \, v^2 \, \exp\left(\frac{v^2}{v_{\mathrm{th}}^2} \right) \tag{410}$$

with the thermal velocity

$$v_{\mathrm{th}} = \sqrt{\frac{2kT}{m}} \tag{411}$$

The collision strength is an integral over all collisions of sufficiently high velocity v_0 (kinetic energy greater than the bound-bound energy difference $h\nu$):

$$\Omega_{lu} = \int_{v_0}^{\infty} Q_{lu}(v) \, v \, w(v) \, dv \tag{412}$$

Q_{lu} is in general a function of the velocity v. Ω_{ij} is a function of the electron temperature T. But even for constant Q_{lu}, Ω becomes a function of T.

For very few transitions, we can use $\Omega(T)$ from quantum mechanical calculation.

For allowed (dipole) transitions, there are approximate formulas which relate the collision strength to the radiative probability of that transition (i.e. to the oscillator strength f_{lu} or the Einstein coefficient A_{ul}.

For forbidden transitions, such relations cannot be applied.

16.1.2. Specification of the collisional bound-bound data

The file DATOM contains a line for each line transition, starting with the keyword LINE, e.g.:

```
*KEYWORD--UPPERLEVEL  LOWERLEVEL--EINSTEIN  RUD-CEY --COLLISIONAL COEFFICIENTS--
LINE      HEI 2S3..2  HEI 1S1..1 1.1300E-4   X BFK1 +7.0E-7       -4.E-13
LINE      N 32P2P4.2  N III2P2.1 2.65E2        KB24 1.0
LINE      N V 3S...3  N V 2S...1             X FCMW 0.0398 0.005220.429  1.047
LINE      N V 4S...6  N V 3P...4-0.071         KB22 0.2
```

Subroutine DATOM reads these lines.

Subroutine COLLI first calculates the *effective collision strength* Ω_{ul}. This is done either in COLLI or in one of its subroutines CBBH (for hydrogen), CBBHE (for helium), CBBN (for nitrogen), or CBBMORE (for all other elements). The subroutine CBBFE is special for generic iron, which is not described in the DATOM file, but by the binary file FEDAT.

Finally, Subr. COLLI calculates

$$C_{ul} = n_{\mathrm{e}}\, \Omega_{ul} \tag{413}$$

and with help of detailed balance:

$$C_{lu} = C_{ul} \left(\frac{n_u}{n_l}\right)^{*} \tag{414}$$

16.1.3. Formulas for collisional bound-bound collision strengths

16.1.3.1. Zero crossection Keyword: ZERO
Subroutine: COLLI

$$\Omega_{ul} = 0 \tag{415}$$

Usage: not used (only for tests)

16.1.3.2. Neutral atoms Keyword:
(empty) for hydrogen
JEFF for helium

Subroutine:
CBBH (hydrogen)
CBBHE (helium)

Source: Jefferies (1968), Eq. (6.24):

$$\Omega_{lu} = 2.16\, \alpha^{-1.68}\, e^{-\alpha}\, T^{-3/2}\, f_{lu} \tag{416}$$

converted to:

$$\Omega_{ul} = 3.24 \left(\frac{hc/k_{\mathrm{B}}}{\lambda T_{\mathrm{e}}}\right)^{-1.68} T_{\mathrm{e}}^{-3/2} \frac{A_{ul}}{\lambda^{-2}} \tag{417}$$

with $hc/k_{\mathrm{B}} = C1 = 1.44$ in cgs →

```
      OMEGA=3.24*EINST(NUP,LOW)/WN2/T32/(C1*WAVENUM/TL)**1.68
```

Usage: permitted transitions in *neutral* atoms, i.e. H I, He I

16.1.3.3. Neutral helium, forbidden (1) Keyword: BFK1 and BFK2

Subroutine: CBBHE

Source: Berrington, Fon & Kingston 1982, MNRAS 200, 347 Berrington et al. (1982)

BFK1:

```
              PBFK=COCO(1,IND)/TROOT
     +                         +COCO(2,IND)+COCO(3,IND)*TROOT
     +                         +COCO(4,IND)*T32
```

BFK2:

```
              PBFK=COCO(1,IND)/TL
     +                         +COCO(2,IND)/TROOT+COCO(3,IND)
     +                         +COCO(4,IND)*TL
              OMEGA=PBFK*WEIGHT(LOW)/WEIGHT(NUP)
```

Usage: He I forbidden transitions between $n = 2$, $n = 1$ or within $n = 2$

16.1.3.4. Neutral helium, forbidden (2) Keyword: BKMS or BKGR (are equivalent)

Subroutine: CBBHE

Source: Benson & Kulander 1972, Solar physics 27, 305 (formula 3)Benson & Kulander (1972), there in references to Mihalas & Stone 1968 Mihalas & Stone (1968)
Green, A.E.S. 1966, AIAA, J. 4, 769

$$\Omega_{lu} = AT^n \exp(-\alpha X_0) \qquad (418)$$

$$X_0 = E_0/kT \qquad (419)$$

```
C***          ATTENTION: COCO(3,IND) := 1.-ALPHA
                OMEGA=COCO(1,IND)*TL**COCO(2,IND)*
     *                 EXP(COCO(3,IND)*C1*WAVENUM/TL)*WEIGHT(LOW)/
     /                 WEIGHT(NUP)
```

Usage: forbidden transitions between $n > 2$, $n = 1$ or $n > 3$, $n \neq 1$

16.1.3.5. Neutral helium, forbidden (3) Keyword: UPS1, UPS2

Subroutine: CBBHE

Source: Formula from Mendoza (1983) ???, ϒ from Schmutz priv. comm., see Diss. Wessolowski

$$\Omega_{ul} = 8.63\ 10^{-6}\ \frac{\Upsilon}{g_u\ \sqrt{T}} \qquad (420)$$

with:
$\Upsilon = 0.05$ for keyword UPS1 (for $\Delta n = 1$)
$\Upsilon = 1.00$ for keyword UPS2 (for $\Delta n = 0$)

Usage: He I intercombination (forbidden) transitions between combined levels (traditional DATOM: $n \geq 4$)

16.1.3.6. Positive ions, allowed transitions (1)

Keyword: blank (helium), JEFF (nitrogen)
Subroutine: CBBHE, CBBN

Source: Jefferies (1968), Eq. (6.25), following van Regemorter

$$\Omega_{lu} = 3.9\,\alpha^{-1}\,e^{-\alpha}\,T^{-3/2}\,f_{lu} \tag{421}$$

with $\alpha = E/k_\mathrm{B}T$. Using the Einstein coefficient

$$A_{ul} = \frac{8\pi}{\lambda^2}\frac{g_l}{g_u}\frac{\pi e^2}{mc}f_{lu} \tag{422}$$

which, in cgs units, reads

$$f_{lu} = 1.499\left(\frac{\lambda}{\mathrm{cm}}\right)^2\,\frac{g_u}{g_l}\,A_{ul} \tag{423}$$

We convert this with help of detailed balance into

$$\Omega_{ul} = 4.06\left(\frac{\lambda}{\mathrm{cm}}\right)^3\,A_{lu}\,T^{-1/2} \tag{424}$$

```
OMEGA=4.06*EINST(NUP,LOW)/WN3/TROOT
```

16.1.3.7. Positive ions, allowed transitions (2)

Keyword: KB22
Subroutine: CBBN
Source: Diss. Wessolowski, van Regemorter (1962, APJ 136, 906)

$$\Omega_{lu} = \pi\,a_0^2\,\sqrt{\frac{8k_\mathrm{B}}{\pi m}}\,\sqrt{T}\,\left[14.5\,f_{lu}\left(\frac{E_\mathrm{H}}{h\nu}\right)^2\right]u_0\,e^{-u_0}\,\Gamma(u_0) \tag{425}$$

with $u_0 = h\nu/k_\mathrm{B}T$, $\Gamma(u_0) = \max\,[\overline{g}, 0.276\,e^{u_0}\,E_1(u_0)]$.

E_1 is the first Exponential Integral, defined as

$$E_1(x) = \int_1^\infty \frac{e^{-xt}}{t}\,\mathrm{d}t \tag{426}$$

$\overline{g} = 0.7$ is a parameter of this formula, and should be set to:

$\overline{g} = 0.7$ for $\Delta n = 0$

and $\overline{g} = 0.2$ else.

$$\pi\,a_0^2\,\sqrt{\frac{8k_\mathrm{B}}{\pi m}} = 5.465\;10^{-11} \tag{427}$$

in cgs.

With help of Einstein coefficients, we convert this into:

$$\Omega_{ul} = 20.56\,\Gamma(u_0)\,A_{ul}\left(\frac{\lambda}{\mathrm{cm}}\right)^3 / \sqrt{T} \tag{428}$$

```
OMEGA=20.56*GAMMA*EINST(NUP,LOW)/WN3/TROOT
```

Usage: allowed transitions of positive ions

16.1.3.8. Positive ions, forbidden transitions Keyword: KB24

Subroutine: CBBN

same formula as in section (16.1.3.5), but with Υ as parameter. This parameter is always set to 1.0.

16.1.3.9. N III, calculations from D. Hummer Keyword: KB23

Subroutine: CBBN
Source: calculations from Hummer (priv. comm.), polynomial fit from Butler 1984, Diss. Wessolowski
Polynomial coefficients are in DATA statements within the subroutine, the parameter gives some index to identify the transition.

$$\Omega_{ul} = \frac{1}{\sqrt{T}} \sum_{i=1}^{N=7} a_i \left(\log(\frac{T}{10^4\,\mathrm{K}}) \right)^{i-1} \tag{429}$$

16.1.3.10. N V, calculations Keyword: ACMW, FCMW

Subroutine: CBBN
Usage: for allowed (ACMW) or forbidden (FCMW) transitions between $n = 2$ and $n = 3$
Source: COCHRANE + MCWHIRTER (1983), PHYSICA SCRIPTA 28, 25-44
Fit formula with three (ACMW) or four (FCMW) coefficients given as parameters. For the allowed transitions the cross section is related to the Einstein coefficient.

ACMW:

```
      GFIT=COCO(1,IND)+COCO(2,IND)*ALOG(TL/WAVENUM/C1+
     +          COCO(3,IND))
      OMEGA=20.56*GFIT*EINST(NUP,LOW)/WN3/TROOT
```

FCMW:

```
      GFFIT=COCO(1,IND)+COCO(2,IND)*ALOG((TL/WAVENUM/C1+
     +           COCO(3,IND))/(TL/WAVENUM/C1+COCO(4,IND)))
      OMEGA=13.71/WAVENUM/TROOT*GFFIT*WEIGHT(LOW)/WEIGHT(NUP)
```

16.1.3.11. Bohr's radius Keyword: NONE

Subroutine: COLLI

$$Q(v) \equiv \pi a_0^2 = 8.8\ 10^{-17}\ \mathrm{cm}^2 \tag{430}$$

leading to the effective collision strength

$$\Omega_{ul} = \pi a_0^2 \sqrt{\frac{8 k_\mathrm{B} T}{\pi m}} \left(1 + \frac{E_0}{k_\mathrm{B} T}\right) \frac{g_l}{g_u} \tag{431}$$

```
OMEGA=5.465E-11*TROOT*(1.+C1*WAVENUM/TL)*WEIGHT(LOW)/WEIGHT(NUP)
```

Source: lecture wrh, diss. Wessolowski

Usage in our traditional DATOM:

He I intercombination transitions transitions (singulet – triplet), if no other data known

N III intercombination transitions transitions (doublet – quartet), if no other data known

N V for forbidden transitions with $\Delta n \neq 1$ and $\Delta L \neq 1$

16.1.4. Summary of Subroutines and keywords

Subroutine	Keyword	Ion	Comment
COLLI	ZERO	all	$\Omega = 0$
	NONE	all	Bohr's radius
CBBH	blank	H I	Jefferies 6.24, allowed
CBBHE	JEFF	He I	Jefferies 6.24, allowed
	BFK1	He I	forbidden, special data
	BFK2	He I	forbidden, special data
	BKMS	He I	forbidden, special data
	BKGR	He I	forbidden, special data
	UPS0		$\Omega = 0$ (obsolete)
	UPS1	He I	forbidden, $\Upsilon = 0.05$
	UPS2	He I	forbidden, $\Upsilon = 1.0$
	blank	He II	Jefferies 6.25, allowed
CBBN	JEFF	N	Jefferies 6.25, allowed
	KB22	N	van Regemorter, allowed
	KB24	N	forbidden, like UPS2, $\Upsilon = 1.0$ first parameter
	KB23	N III	Hummer fits (calculations)
	UPS0	N IV	$\Omega = 0$ (obsolete)
	UPS1	N IV	forbidden, like UPS2
	ACMW	N V	allowed (calculations)
	FCMW	N V	forbidden (calculations)
CBBMORE	JEFF	all others	Jefferies 6.25, allowed
	KB22	all others	van Regemorter, allowed
	KB24	N	forbidden, like UPS2, $\Upsilon = 1.0$ first parameter
CBBFE			Superlines, van Regemorter

WARNING: Note that some keywords have different meaning, depending on the element for which they are used!

16.1.5. Defaults and recommendations

Individually calculated data, when available, are extected to be more accurate than the general formulae. However, the representation of $\Omega(T)$ by fits is dangerous and sometimes giving nonesense! Note, that there is a special PROGRAM PLOTCBB to visualize the collisional cross sections over temperature.

Among the general formulae, there is an unclear choice; different DATOM versions exist, e.g. for allowed transitions of nitrogen:
JEFF (no parameter) or KB22 with parameter $\overline{g} = 0.7$ (if $\Delta n = 0$) or 0.2 (else),
and for the forbidden transitions of nitrogen:
NONE (Bohr's radius) or KB24 with parameter 1.0

16.1.5.1. Defaults by program opdat

The opdat-program uses the following recipes:

Allowed transitions $f \geq 0.01$
Formula from van Regemorter (1962):
CKEY = KB22
and the parameter:

- `KB22 0.7` : if $\Delta n = 0$

- `KB22 0.2` : else

Forbidden transitions $f < 0.01$
Set `X` for "rudimental", but not if one of the involved levels is the groundstate and its f-value is known.
Set CKEY and coefficient depending on the change of principle quantum number:

- if $\Delta n > 1$: `CKEY = NONE` (means: take Bohr's radius, accounting for ΔE)
- if $\Delta n = 1$: `KB24 0.05`
- if $\Delta n = 0$: `KB24 1.0`

As there is no CBB formula for forbidden transitions of positive ions, we adopted (see N V memo from 2007) the scheme from neutral He I. This scheme is from Wessolowski (1991), allegedly motivated by Mendoza (1983). Note that Wessolowski (1991) had set CBB keyword for forbidden transitions of ions to "NONE" (=Bohr's radius).

16.1.6. PLOTCBB - a plotting tool

There exists a main program PLOTCBB for creating plots of Ω_{ul} versus the logarithm of temperature. There is a job (similar to the steal job), see e.g. wrh/work/wrjobs/plotcbb1, which can be submitted in the usual way (e.g. `sub plotcbb1`). The plots are based on the current DATOM and FEDAT files in the corresponding wrdata*n* directory.

PLOTCBB can show Ω_{ul} for one or more line transitions of a given line index, or a range of line indices. Consult the output from PRINT DATOM to figure out line indices.

The input of PLOTCBB consists of only one line, which must be edited inside the plotcbb1 jobfile. This line must read
LINE *n*
or
LINE *n* - *m*
where *n* and *m* are the first and last line indices for which the data are plotted.

Note that WRplot is limited to show 100 datasets within one plot. To show more plots, insert a SET_NSETMAX statement into the WRplot source file.

16.2. Collisional bound-free transitions

Calculated by `COLLI`, only one formula (!) does not depend on element (**?**, p. 121, Eq. 6.39):

$$C_{lc} = 1.55 \times 10^{13} \, \frac{\overline{g}_i \, a(0) \, e^{-\alpha} \alpha^{-1} n_\mathrm{e}}{\sqrt{T}} \tag{432}$$

with $\alpha = E/k_\mathrm{B}T$ and $\overline{g}_i = 0.1, 0.2, 0.3$ for $Z = 1, 2, > 2$, the charge of the upper ion. The collisional b-f rate is related to $a(0)$, the photo ionization cross section at the threshold E_th. We may rewrite the collision strength for energy E in cm^{-1} with help of $hc/k_\mathrm{B} = 1.44$ cm grad, regarding, that our $a(0)$ is given in Mbarn (10^{-18} cm^2):

$$\Omega_{lc} = 1.55 \times 10^{13} \, \frac{\overline{g}_i \, a(0) \, e^{-h\nu/k_\mathrm{B}T}}{\sqrt{T}} \left(\frac{h\,c}{k_\mathrm{B}} \frac{E}{T} \right)^{-1} = 1.076 \times 10^{-5} \, \overline{g}_i \, \sqrt{T} \, \sigma_\mathrm{th} \, \frac{e^{-E/k_\mathrm{B}T}}{E_\mathrm{th}} \tag{433}$$

Therefore in `COLLI`:

```
      G=.3
      IF (NCHARG(NUP) .EQ. 2) G=.2
      IF (NCHARG(NUP) .EQ. 1) G=.1
```

Calculation of Ω:

```
     OMEGA=G*1.08E-5*TROOT*EINST(LOW,NUP)*EXPFAC/EDGE
      CRATE(LOW,NUP)=ENE*OMEGA
      CRATE(NUP,LOW)=ENE*OMEGA*ENLTE(LOW)/ENLTE(NUP)
```

where `EINST(LOW,NUP)` is `SIGMA`, the first parameter of the `CONTINUUM` card.

16.3. Radiative line transitions

Main program COLI provides the *scattering integral* $\bar{J}_L$, defined as

$$\bar{J}_L = \int_{\text{line}} J_\nu \, \phi(\nu) \, \mathrm{d}\nu \tag{434}$$

for each line, where $\phi(\nu)$ is the profile function (see Sect. **??**). Each vector $\bar{J}_L$ (over depth index) is stored in the model file under the name XJL*nnnn*, where *nnnn* is the line index. Main program STEAL reads these vectors for calculating the rate coefficients.

However, for those lines treated as *rudimental* (see ... ...), $\bar{J}_L$ is not provided by COLI. For these lines, it is assumed that $\bar{J}_L = J_C(\nu_L)$, i.e. equal to the continuum intensity at the line frequency. $J_C(\nu_L)$ is obtained by interpolation in the array of continuum intensities. The interpolation is done in terms of radiation temperature (subroutine XRUDI).

This approximation should be critically reconsidered!!

The rate coefficients are defined by

$$R_{lu} = B_{lu} \, \bar{J}_L \tag{435}$$

and

$$R_{ul} = A_{ul} + B_{ul} \, \bar{J}_L \tag{436}$$

Remember that the Einstein coefficients are related via

$$A_{ul} = \frac{2h\nu_L^3}{c^2} \, B_{ul} \quad \text{and} \quad g_l \, B_{lu} = g_u \, B_{ul} \tag{437}$$

The PoWR code makes use of the Einstein coefficient A_{ul}, stored in the array EINST. Hence the radiative line rates are

$$R_{lu} = \frac{c^2}{2h\nu_L^3} \, \frac{g_u}{g_l} \, A_{ul} \, \bar{J}_L \tag{438}$$
$$R_{ul} = A_{ul} + \frac{g_l}{g_u} \, R_{lu}$$

The above formulation of the rates is implemented in subroutine RADIO (calling tree: STEAL - POPZERO - NLTEPOP - RADIO). This branch is only used when all ALO-Gammas are zero, because than the rate equations can be kept entirely linear.

Otherwise, the rate coefficients are calculated in subroutine RADNET (Calling tree: STEAL - LINPOP - COMA - RADNET. Because with ALOs the rate equations are non-linear anyhow, *Net Radiative Brackets*

(NRBs) are employed for all line transitions which have a non-zero line core, i.e. lines for which the approximate lambda operator is zero. (For all other lines, the rates are calculated as described above.) Superline transitions always go through the NRB branch.

The NRB notation was proposed by Mihalas to achieve a better stability for optically thick transitions, by expressing the rate in terms of the ratio between intensity and source function.

In the rate equation one can group pairs of terms, referring to the transition $u \rightarrow l$ und the contrary transition $l \rightarrow u$,

$$n_u\, R_{ul} - n_l\, R_{lu} \tag{439}$$

Such pair is now re-written as

$$n_u\, (R_{ul} - \frac{n_l}{n_u}\, R_{lu}) \tag{440}$$

The term in brackets (the "Net Radiative Bracket") is now expressed with help of the non-LTE source function. First we insert Eqns. 438 and factor out the common A_{ul}:

$$A_{ul} \left(1 + J_L\, \frac{c^2}{2h\nu_L^3} \left(1 - \frac{g_u\, n_l}{g_l\, n_u} \right) \right) \tag{441}$$

Comparison with the non-LTE line source function,

$$S_L = \frac{2h\nu_L^3}{c^2} \frac{n_u}{\frac{g_u}{g_l} n_l - n_u} = \frac{2h\nu_L^3}{c^2} \left[\frac{g_u\, n_l}{g_l\, n_u} - 1 \right]^{-1} \tag{442}$$

reveals that the NRB can be written as

$$A_{ul} \left(1 - \frac{J_L}{S_L} \right) \tag{443}$$

Hence, the rate equation is recovered by defining new rate coefficients as

$$\begin{gathered} R_{lu} = 0 \\ R_{ul} = A_{ul} \left(1 - \frac{J_L}{S_L} \right) \end{gathered} \tag{444}$$

Note that the source function S_L depends on the population numbers n_i. Hence, albeit Eq. 444 looks simple, it implies that the coefficients of the rate equations now themselves depend on the unknown variables n_i, i.e. the equations are no longer linear.

16.4. Radiative bound-free transitions

Data for radiative continuum transitions (photoionization and recombination) are given explicitly in DATOM, e.g.:

```
*KEYWORD  LOWERLEVEL ----SIGMA ----ALPHA --SEXPO-- -IGAUNT- -KEYCBF- --IONLEV--
CONTINUUM C IV 2S..1     .652     1.539   2.168
CONTINUUM C 33P3DP23    0.9675    2.0510  -0.25922 PIKB12            C IV 2P.21
                    -4.7573E-2    0.        0.
CONTINUUM N 43S'P114    0.905     1.6816 -0.12313  BUTLER12          N V 2P...2
CONTINUUM N 34S'P431    1.134     1.5014    10.    DETAILN3          N IV 2P3.2
CONTINUUM H I......1     7.92                      MIHALAS
CONTINUUM H I......2    15.84                2.    SEATON
CONTINUUM HEI 1S1..1    7.43      4.3965 -0.22134  KOESTER
CONTINUUM NE7s2 1S01  2.90E-01
```

Despite the parameters and keywords for the formulas there is also the possibility to give the target level of the upper ion as the last parameter (IONLEV), otherwise the ground state of the upper ion is the target state.

These cards are read in by DATOM and the cross sections are then calculated by PHOTOCS, which is called by BFCROSS (preparation of b-f cross sections in array SIGMAKI, loop over continua and frequencies),
COOP (continuous opacities for all depth points, loop over continua and frequencies, if SIGMAKI not yet calculated),
and CMFCOOP (optimized version of COOP).

These subroutines calculate only the continuum opacity. To get the rates, the subroutine RADIO (linear) calculates the rate coefficients

$$R_{lu} = 4\pi \int_{\nu_0}^{\infty} \sigma_\nu \frac{J_\nu}{h\nu} d\nu \tag{445}$$

and from detailed balance (LTE), where $J_\nu = B_\nu$ and therefore $\frac{R_{ul}}{R_{lu}} = \left(\frac{n_l}{n_u}\right)^*$ we obtain:

$$R_{ul} = 4\pi \left(\frac{n_l}{n_u}\right)^* \int_{\nu_0}^{\infty} \frac{\sigma_\nu}{h\nu} \left(\frac{2h\nu^3}{c^2} + J_\nu\right) \exp\left(-\frac{h\nu}{k_\mathrm{B}T}\right) d\nu \tag{446}$$

Or, analogously, the subroutine RADNET calculates the rate coefficients with help of net radiative brackets, where $Z_{ul} := R_{ul} - \frac{n_l}{n_u} R_{lu}$ and $Z_{lu} = 0$:

$$Z_{ul} \;=\; 4\pi \left(\frac{n_l}{n_u}\right)^* \int_{\nu_0}^{\infty} \frac{\sigma_\nu}{h\nu} \exp\left(-\frac{h\nu}{k_\mathrm{B}T}\right) \frac{2h\nu^3}{c^2} \left(1 - \frac{J_\nu}{S_\nu^{lu}}\right) d\nu \tag{447}$$

where

$$S_{lu} \;=\; \frac{2h\nu^3}{c^2} \frac{1}{\frac{1}{G_\nu}\frac{n_l}{n_u} - 1} \tag{448}$$

$$\text{and} \quad G_\nu \;=\; \frac{g_l}{g_u} \exp\left\{\frac{h(\nu_0 - \nu)}{k_\mathrm{B}T}\right\} n_\mathrm{e} c_3 T^{-3/2} \qquad (\text{ cf. Saha Eq.}) \tag{449}$$

Following quantities are prepared within in PHOTOCS for every cross section frequency point:

```
X=EDGE/WAVENUM
XINV=1./X
```

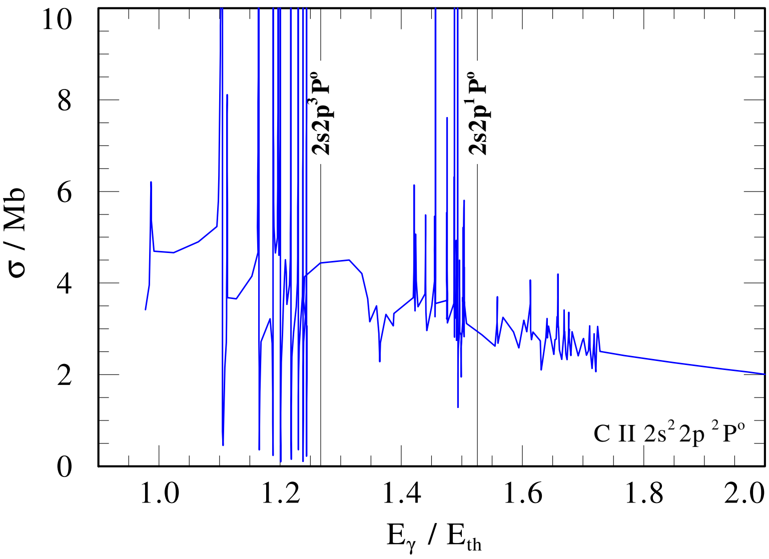
All formulas are in a form, that for $\nu_\mathrm{th} \equiv \nu_0$ the cross section $\sigma_\mathrm{th} \equiv \sigma_0$ is given (keyword SIGMA in file DATOM). The unit of σ is always Mbarn ($10^{-18}\,\mathrm{cm}^2 = 10^{-22}\,\mathrm{m}^2$).

PHOTOCS then sets all cross sections with $\sigma < 0$ or $\sigma > 10 \times \sigma_0$ to zero, in the last case, a warning is printed out:

```
      WRITE (0,*) 'WARNING : VERY HIGH PHOTOIONISATION CROSS ',
     >             'SECTION DETECTED; SET TO ZERO'
```

16.4.1. Detailed OP photoionization cross sections, e.g. C II

Radiative bound-free cross sections $\sigma(\nu)$ can have a complicated ν dependence due to resonances or dielectronic recombination (cf. Fig. 24).



C II	C III	E(C III)/Ry
2s^2 2p ^{2}P^o	2s^2 ^{1}S	0.0
2s 2p^2 ^{4}P	2s 2p ^{3}P^o	0.477
	2s 2p ^{1}P^o	0.940

Figure 24: Opacity project photo cross section for C II, summed up over all target states

In Fig. (25) are shown resonances for the $2s2p^2\ ^2S$ and $2s2p^2\ ^2D$ state. As the transition $2s2p^2 + h\nu \rightarrow 2s^2 + \mathrm{e}$ is forbidden, due to two-electron jump, one has to include interactions with other configurations, e.g.
$2s2p^2 + h\nu \rightarrow 2s2p(^3P^o)n\ell \rightarrow 2s^2 + \mathrm{e}$

Now, the problem is raised, how to treat such resonances within the code. Usually a fitting formula, smoothing over these resonances is used, which should at least recover the integral $\int \sigma_\nu d\nu$. A more detailed treatment is possible for the Low Temperature Dielectronic Recombination (LTDR, see below). However, the treatment of the sharp resonances within the radiative transfer equation remains unclear (e.g. what is width of these "spikes") and is not consistent.

In the following the fit formulas for photoionization cross sections from various sources is presented.

16.4.2. Formulas for the bound-free photo cross sections

16.4.2.1. Kramers formula - Hydrogenic cross section

Keyword: blank
Parameters: σ_0
Source: Kramers (1923), Cowan (1981):

$$\overline{Q}_n^p \equiv \overline{Q}_{n,\epsilon} \simeq \frac{64\pi\alpha a_0^2 n}{3^{3/2} Z_c^2}\left(\frac{\epsilon_n}{\epsilon}\right)^3 = 7.91 \times 10^{-18}\,\mathrm{cm}^2\,\frac{n}{Z_c^2}\left(\frac{\epsilon_n}{\epsilon}\right)^3 \tag{450}$$

where $\epsilon_n = Z_c^2/n^2$ is the threshold ionization energy for the shell n, $Z_c = Z - N + 1$ (N is total number of electrons, including the free electron, so that $V(r)_{\mathrm{Coulomb}} \simeq -2Z_c/r$), and $\alpha = \mathrm{e}^2/\hbar c$ the fine-structure constant. This is only an approximation (even for non-relativistic one-electron atoms), the accurate $\overline{Q}_n^p$ varies as $\epsilon^{-8/3}$ to $\epsilon^{-7/2}$ (Cowan 1981).

The cross section σ_0 at the threshold $(\nu/\nu_0)^{-3} = 1$ for any level with energy E w.r.t. the ground state of the

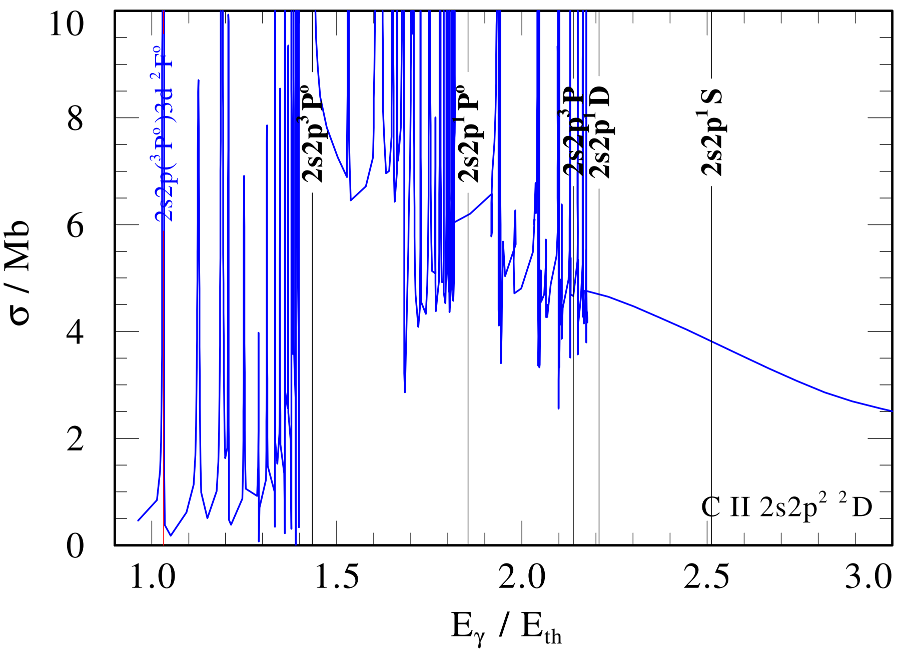


Figure 25: Photoionization cross section (total) for C II 2s 2p^2 ^{2}D. The ground state of C III can only be reached by the intermediate auto-ionization state 2s 2p(^{3}P^0)3d ^{2}F^0. For higher photon energies also other configurations of C III can be target levels, depending on which electron was removed (different spins and m_ℓ). The OP photoionization cross section is the sum of the photoionization cross sections for all individual transitions from the given level of the lower ion to the different target levels.

lower ion can be roughly estimated from the formula:

$$\sigma_0 = \frac{4.541 \times 10^8 \,\text{Mbarn}}{(z - N + 1)\sqrt{(E_\text{ion} - E)\,c}} \tag{451}$$

$$= \frac{2.623 \times 10^3 \,\text{Mbarn}}{(z - N + 1)\sqrt{(E_\text{ion} - E)}} \tag{452}$$

with ion charge $z - N$ (e.g. $z - N = 0$ for He I), the threshold energy $(E_\text{ion} - E)$ in cm^{-1} and the speed of light c in cm s^{-1} (Dissertation U. Wessolowski, Wiese, Götz).

Example:

```
CONTINUUM NE7s2 1S01 2.90E-01
```

16.4.2.2. Seaton fit-formula Keyword: blank

Parameters: σ_0, α, s

Source: Seaton (1958) This formula is used to fit the slope of detailed cross sections:

$$\sigma(\nu) = \sigma_0 \left(\alpha \left(\frac{\nu_0}{\nu} \right)^s + (1 - \alpha) \left(\frac{\nu_0}{\nu} \right)^{s+1} \right) \tag{453}$$

Example:

```
*KEYWORD  LOWERLEVEL ----SIGMA ----ALPHA --SEXPO-- -IGAUNT- -KEYCBF- --IONLEV--
CONTINUUM N V 2S...1    0.5       1.5        2.3
```

16.4.2.3. Koester fit for He I   Keyword: KOESTER

Parameters:a_0, a_1, a_2

Source: Koester et al. (1985), fit formula for calculations from Jacobs (1971, 1972, 1974) and Stewart (1978, 1979)

$$\ln g\sigma = a_0 + a_1 \ln \lambda + a_2 (\ln \lambda)^2 \tag{454}$$

with statistical weights g, coefficients a_0, a_1, a_2 in Koester et al. (1985) are for λ in Å and σ in cm^2. This may be transformed to

$$\sigma = \frac{\exp(a_0)}{g} \lambda^{a_1} \exp(a_2 (\ln \lambda)^2) \tag{455}$$

By dividing through $\sigma_0 = \frac{\exp(a_0)}{g} \lambda_0^{a_1} \exp(a_2 (\ln \lambda_0)^2)$ one obtains:

$$\sigma = \sigma_0 \left(\frac{\lambda}{\lambda_0} \right)^{a_1} \exp[a_2 (\ln^2 \lambda - \ln^2 \lambda_0)] \tag{456}$$

The coefficients are identified as follows: SIGMATH = $f(a_0)$, ALPHA = a_1, SEXPO = a_2, where $f(a_0)$ absorbs some constants, so, that in the program

```
X=EDGE/WAVENUM
XLN=ALOG(1.E8/WAVENUM)
XLN2=XLN*XLN
X0LN=ALOG(1.E8/EDGE)
X0LN2=X0LN*X0LN
SIGMA=SIGMATH*X**ALPHA(KON)*EXP(SEXPO(KON)*(XLN2-X0LN2))
```

Example:

```
CONTINUUM HEI 1S1..1    7.43      4.3965 -0.22134  KOESTER
```

16.4.2.4. Butler, modified Seaton fit for N IV   Keyword: BUTLER12

Parameters: σ_0, α, s

Source: Butler (priv. comm.) gives modified fit formula

$$\sigma = \sigma_0 \left(\frac{\nu_0}{\nu} \right)^{\alpha + s * \ln(\nu_0/\nu)} \tag{457}$$

```
SIGMA=SIGMATH*X**(ALPHA(KON)+SEXPO(KON)*ALOG(X))
```

Example:

```
CONTINUUM N IV 2S1.1    0.967     1.5013 -0.27972  BUTLER12
```

16.4.2.5. Butler, extended version of Butler12   Keyword: DETAILN3

Parameters: σ_0, α, s, a_3, a_4, a_5, a_6

Function: PHOTON3

Source: Butler (priv. comm.), the formula is half coded in the function photon3.f and half in subroutine PHOTOCS. In PHOTOCS the parameter s is only used as index for the function PHOTON3. If $s = [4; 6]$, the cross section is set to zero for $\frac{\nu_0}{\nu} \leq 0.05$ ($\lambda \leq 30$ Å).

$$\sigma = \sigma_0 \left(\frac{\nu_0}{\nu} \right)^{\alpha + \Sigma} \tag{458}$$

$$\text{function photon3: } \Sigma = \sum_{i=3}^{6} a_i \left(\ln \frac{\nu_0}{\nu} \right)^{i-2} \tag{459}$$

Σ is calculated in photon3.f

```
      XLN=ALOG(X)
      PHOTON3=XLN*(AI(3,NO)+XLN*(AI(4,NO)+XLN*(AI(5,NO)+AI(6,NO)*XLN)))
```

Example:

```
*KEYWORD  LOWERLEVEL ----SIGMA ----ALPHA --SEXPO-- -IGAUNT- -KEYCBF- --IONLEV--
CONTINUUM N 32P2P4.2    3.871     2.1302     1.    DETAILN3           N IV 2P3.2
```

16.4.2.6. Obsolete PIKB12 Keyword: PIKB12

Parameters: σ_0 (,α, s, a_1, a_2, a_3)
Source: unknown, formula with $x = \nu_0/\nu$

$$\sigma = \sigma_0 x^{\alpha+\ln x\,(s+\ln x\,(a_1+\ln x\,(a_2+\ln x a_3)))} \tag{460}$$

$$= \sigma_0 \exp\left(\alpha \ln x + s(\ln x)^2 + a_1(\ln x)^3 + a_2(\ln x)^4 + a_3(\ln x)^5\right) \tag{461}$$

As this expression diverges for small x (high energies), the formula is replaced by the hydrogenic cross section and a warning is printed out:

```
 *** WARNING issued from PHOTOCS:
 *** Obsolete Formula PIKB12 replaced by hydrogenic slope ***
```

Note: This formula works with a continuation line, containing three further coefficients. Therefore in subroutine DATOM the keyword PIKB12 signalized to read in another line and convert this into coefficients ADD1, ADD2, ADD3, or ADDCON1(KONT), ADDCON2(KONT), ADDCON3(KONT) respectively.

Example:

```
CONTINUUM C 33S3PP13    0.4110    1.4742  -0.19453 PIKB12            C IV 2P.21
                    -1.4499E-2    0.        0.
```

16.4.2.7. OP fits Keyword: OPAPROIX

Parameters:σ_0, α, s, a_1
Source: unknown
Opacity project data fit (The lithium isoelectronic sequence; Peach, Saraph & Seaton 1988, J. Phys. B: At. Mol. Opt. Phys. 21, 3669)

$$\sigma = \sigma_0 \left(\frac{\nu}{\nu_0}\right)^{\alpha+s\ \log(\nu/\nu_0)+a_1\ \log^2(\nu/\nu_0)} \tag{462}$$

```
          XLOG = ALOG10(XINV)
          SIGMA = SIGMATH*XINV**(ALPHA(KON)+XLOG*(SEXPO(KON)+
     +                                  XLOG*ADDCON1(KON)))
```

Note: Not used.

16.4.2.8. Mihalas Keyword: MIHALAS

Parameters: σ_0 (, a_0, a_1, a_2, a_3, a_{-1}, a_{-2}, a_{-3} DATA in photocs)
Source: (Mihalas 1967, p. 187)

Modification of Kramers formula with help of Gaunt factors, interpolation of Gaunt factors from threshold up to 50 Å:

$$g_{II}(n,x) = a_0 + a_1\,x + a_2\,x^2 + a_3\,x^3 + a_{-1}\,x^{-1} + a_{-2}\,x^{-2} + a_{-3}\,x^{-3} \tag{463}$$

```
IF (X .GT. 0.055) THEN
   GAUNT=A0+X*(AM1+X*AM2)+(A1+(A2+A3*XINV)*XINV)*XINV
ELSE
   GAUNT=1.
ENDIF
SIGMA=SIGMA*GAUNT
```

Example:

```
CONTINUUM HE II....1     1.981                    MIHALAS
CONTINUUM H I......1     7.92                     MIHALAS
```

16.4.2.9. Hydrogen-like levels – Keyword: `SEATON`

Parameters: σ_0, $s(= n)$
Source: Seaton (1960)
Main quantum number n is stored in s:

$$g = 1 + 0.17238\left(\frac{\nu}{\nu_0} - 2\right)\left(\frac{\nu_0}{\nu\,n}\right)^{2/3} - 0.0496\left(\left[\frac{\nu}{\nu_0} - 1\right]\left(\frac{\nu}{\nu_0} + \frac{1}{3}\right) + 1\right)\left(\frac{\nu_0}{\nu\,n}\right)^{4/3} \tag{464}$$

```
      U=XINV - 1.
      DEN=(X/SEXPO(KON))**0.666666666666
      GAUNT=1. + 0.1728 * (U-1.) * DEN -
     -      0.0496 * (U*(U+1.333333333333)+1.) * DEN * DEN
      SIGMA=SIGMA*GAUNT
```

Example:

```
CONTINUUM N V 4F...9    1.263                    4.    SEATON
```

16.4.2.10. Summary The different formulas which are implmented for the radiative bound-free transitions are listed in Table 3.

16.4.2.11. Recommendation If available the exact OP cross sections should be fitted, e.g. with Seaton fit formula. Due to the complex structure, this may be done manually and a smoothing of the curve should be performed first. Furthermore, the integral $\int \sigma_\nu d\nu$ should be recovered.

If no exact cross section is available, the hydrogenic approximation with effective quantum number for calculation of the threshold cross section is used, e.g. by the program `opdat`.

16.5. K-shell ionization

Despite ionization via removal of the outer (photo) electron, there is also the possibility of removing the inner (K-shell, 1s-) electron via X-rays.

Analogously to `PHOTOCS`, the subroutine `KSIGMA` (called by `COOP`, `CMFCOOP`, `COOPFRQ`, `DCOOP`, `RADIO`, and `RADNET`) calculates the K-shell cross section:

Table 3: Summary of the radiative bound-free formulas

Keyword	Coefficients	Ion	Comment
blank	σ_0	all	hydrogenic
blank	$\sigma_0,\ \alpha,\ s$	all	Seaton fit
`KOESTER`	$f(a_0),\ a_1,\ a_2$	He I	Koester fit to calculations
`BUTLER12`	$\sigma_0,\ \alpha,\ s$	N IV	Butler, priv. comm.
`DETAILN3`	$\sigma_0,\ \alpha,\ s, + 3$	N III	Butler, priv. comm., hard coded coeff.
`PIKB12`	σ_0	C III, II	*replaced by hydrogenic cross section*
`OPAPROIX`			not used
`MIHALAS`	σ_0	H (n = 1), He II (n = 1)	gaunt factors coded in photocs
`SEATON`	$\sigma_0,\ s = n$	H (n > 1), He II (n > 1), N V	for hydrogenic levels

```
      X=EDGEK/WAVENUM
      SIGMAK = SIGMATHK * 1.E-18 * X ** SEXPOK
```

Data for K-Shell cross section can be given optional in `DATOM`, e.g. old style with averaged opacities:

```
*KEYWORD--*****SY***<-K-SIGMA><-K-SEXPO>***<-K-EION->
K-SHELL         C         1.0       2.50        2400000.
K-SHELL         N         0.71      2.54        3333333.3
K-SHELL         O         0.483     2.605       4928045.
```

or new style (`~wrh/work/wrdata/DATOM.K-SHELL-by-ions`) with data for every ion:

```
*KEYWORD--*****SY*I*<-K-SIGMA><-K-SEXPO>***<-K-EION->
K-SHELL         C  1      0.742     2.47           291.EV
K-SHELL         C  2      0.665     2.48           308.EV
K-SHELL         C  3      0.807     2.49           329.EV
K-SHELL         C  4      0.811     2.51           352.EV
```

With the cross section at the threshold `K-SIGMA` im Mbarn, the exponent of the fitting formula `K-SEXPO` and the threshold energy `K-EION`, which can be given in cm^{-1} or by using the trailing `EV` in eV.

The exponent `K-SEXPO` is from Daltabuit & Cox (1972), energies and threshold cross section are taken from Verner et al. (1993); Verner & Yakovlev (1995), e.g.:
`C       I:  Z= 6, N= 6, K-SIGMA= 0.938 Mb, E_th=   291.0 eV`
Note: The recent `K-SIGMA` *are confused with the σ_0 from Verner & Yakovlev (1995).*
There, the fit formula is

$$\sigma_{nl}(E) = \sigma_0 F(E/E_0) \tag{465}$$

$$F(y) = \left[(y-1)^2 + y_{\rm w}^2\right]^{-Q}\left(1 + \sqrt{y/y_{\rm a}}\right)^{-P} , \tag{466}$$

where n is the principal quantum number of the shell, $l = 0, 1, 2$ is the subshell orbital quantum number, E is the photon energy in eV, $y = E/E_0$, $Q = 5.5 + l - 0.5P$; σ_0, E_0, $y_{\rm w}$, $y_{\rm a}$, and P are the fit parameters. It must be point out that in this case, σ_0 and E_0 are not $\sigma_{\rm th}$ and $E_{\rm th}$, respectively.

Note: If cross sections are not given for every ion explicitly, the same given mean data are used for every ion.

We assume that K-shell absorption is always followed by auto-ionization, e.g. Auger effect (inner photo effect):

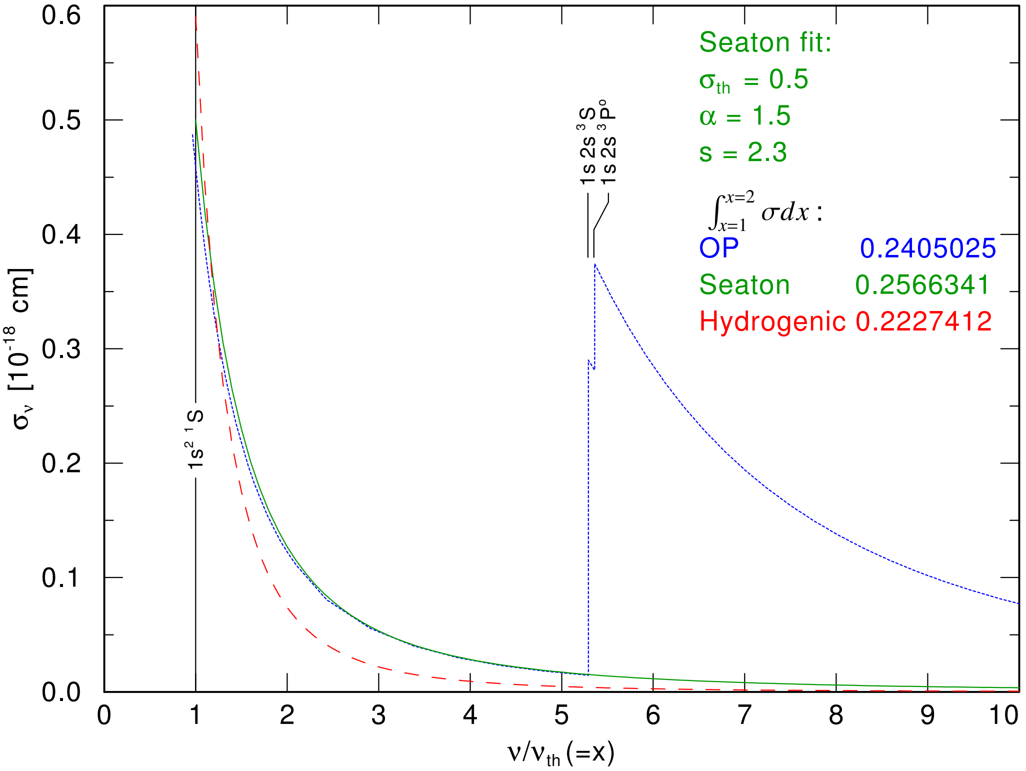


Figure 26: OP photo ionization cross section (blue dotted line) for N v ground state $1s^2 2s\ ^2S$ vs. Seaton fit (green solid line) and Hydrogenic fit (red dashed line). Shown are also the values of the integrals for the curves in the interval $\nu/\nu_{th} = [1;2]$. Also identified are the threshold energies for the levels of the upper ion N vi

.

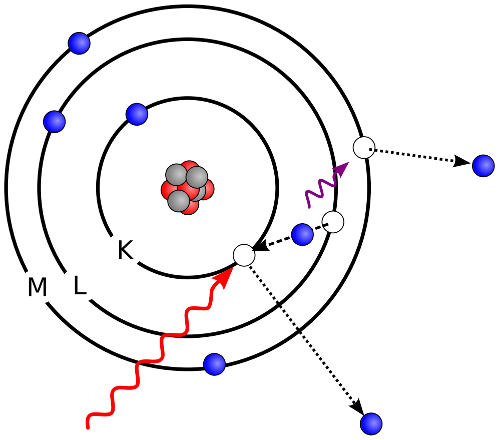


Figure 27: (Ref.: Wikimedia commons)

Through Auger effect, one X-ray photon with energy above E_{th} removes two electrons, therefore ionization stage changes by +2. Auger-ionization needs at least 4 electrons. The competing process, emission of an X-ray photon (X-ray fluorescence), is dominant for heavy elements with $Z > 33$.

Note that the upper level of the Auger process should be the first excited state (of the helium-like ion?). As for C v, N vi, O vii usually only the ground state is included in the model, the ground level is taken instead.

If there are only 3 electrons or, in the case of more than three electrons, the upper level is not included in the rate matrix, then K-shell ionization occurs without Auger effect.

K-shell ionization with only two electrons is equivalent to normal photoionization.

16.6. Dielectronic recombination

Close to the ionization threshold often exist *doubly* excited states, possibly with (positive) energies above the ionization threshold. These states may be reached by absorption of a photon from lower ion or recombination with an electron from upper ion. Then two competing processes are possible, either the excited atom makes auto-ionization, thus an electron is ejected and an excited ion may be left, or a stabilizing transition back to the ground state via photon emission occurs. This process is called *Low Temperature Dielectronic Recombination* (LTDR).

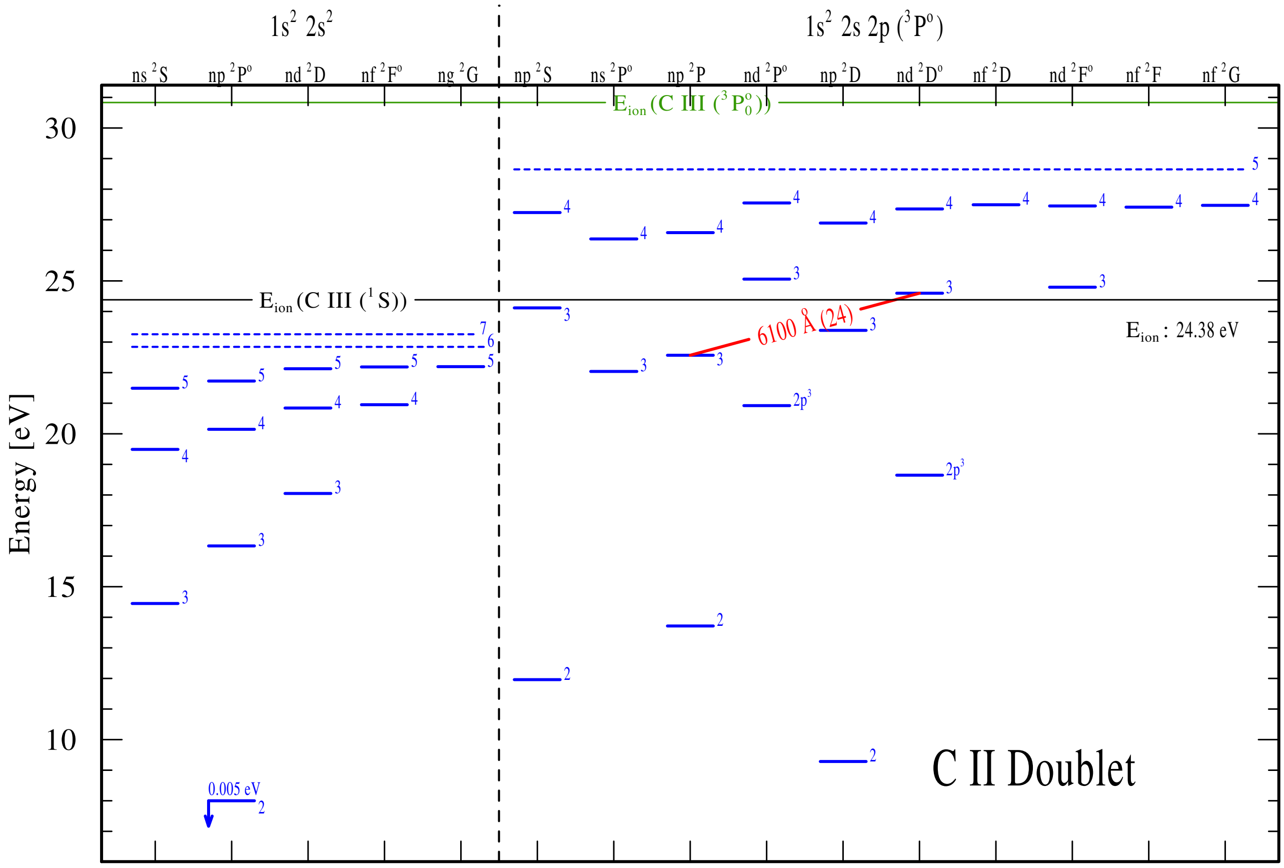


Figure 28: Grotrian diagram adopted from Leuenhagen & Hamann (1994) for C II doublets with $2s^2$ singly (left) and 2s 2p doubly (right) excited states. Triply excited states are indicated by $2p^3$. The levels above the ionization threshold are considered to be in LTE with the C III ground state. None of these LTE levels can auto-ionize (to the C III ground state $2s^2\,{}^1S$) without radiation, as the inner $n = 2$ electrons have parallel spins: $1s^2\,2s2p(^3P^o)$. Therefore, their lifetimes are comparable to normal levels and they can de-excite by a stabilizing line transition, e.g. $2s\,2p\,3d\,{}^2D^o$ - $2s\,2p\,3p\,{}^2P$ at 6100 Å.

By use of the DATOM entries for DRTRANSIT, e.g.:

```
*KEYWORD  LOWERLEVEL  UPPERLEVEL --G- --ENERGY-- -EINSTEIN-  PARENT-ION RUD
DRTRANSIT C 22P4PP.2  C 23D4PP34  12       2193    -0.13830
DRTRANSIT C 22P4PP.2  C 24S4PP37  12      12931    -0.02158
```

we assume LTE for auto-ionization levels (upper levels) w.r.t. ground state of upper ion. Auto-ionization levels are inserted in the rate matrix with rates of stabilizing lines with background radiation field (optically thin), mostly auto-ionization occurs within $\approx 10^{-14}$ s. Then due to energy-time uncertainty principle (Heisenberg) there energies are unsharp, therefore no (sharp) lines must be considered in RTE.

However, there are exception for some levels (see e.g. Fig. 28), for which auto-ionization is prohibited by selection rules, thus they also have sharp level energies. The assumption of optically thin lines may then violate energy conservation, and the stabilizing lines must be treated in the RTE like normal lines (see Fig. 29).

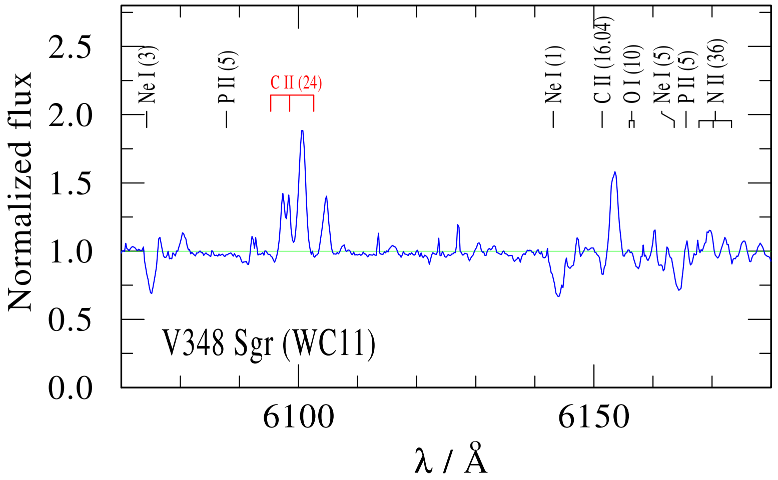


Figure 29: V348 with visible lines due to dielectronic recombination.

UNDER CONSTRUCTION:

Explanation:

Auto-ionizing levels should have a short life-time; therefore, their energies are supposedly unsharp (Heisenberg's uncertainty principle), making the stabilizing lines much broader that normal spectral lines. For that reason, by default we treat the stabilizing lines as *rudimental lines*, i.e. their opacities/emissivities are neglected in the radiative transfer, and their radiative rates are calculated with the background radiation field.

However, this is not the full truth: certain doubly-excited states cannot decay easily because of selection rules, albeit their energy lies above the ionization threshold. Then, the life-time of such states becomes “normal”, and the stabilizing lines are as narrow as normal spectral lines (see above; note, however, that the calculation of such stabilizing lines in the formal integral is completely independent from the MODEL calculation).

Therefore, one can specify in the CARDS file:

DRLINES ALL

meaning that ALL stabilizing transitions are treated in the radiative transfer like normal "narrow" lines.

If levels below the ionization threshold are included in the DRTRANSIT data, the corresponding levels are presumably sharp, and the corresponding stabilizing lines are treated as narrow lines. However, the inclusion of such levels is no longer default, and discouraged (see above).

There is also the CARDS option DRLINES NONE meaning that none of the stabilizing lines is treated as being sharp. If levels below the ionization threshold are not included, as is now the recommended default, DRLINES NONE has no effect compared to the default.

Future work: One could go through all DRTRANSIT data in our database, identify the terms of the auto-ionizing levels, and figure out whether they can auto-ionize via any dipole transition. It is already foreseen in the DRTRANSIT data line to mark explicitely stabilizing lines as *rudimental* by an `X` in the corresponding column, or by `NO` if the upper levelis definitely sharp. Such entries would supersede the global `DRLINES` setting.

16.7. Radiative free-free transitions (Bremsstrahlung)

Interaction of free electrons with ions of charge Z where we assume a Maxwellian distribution of the momentums of the electrons (thermalization of the electron gas), i.e. free-free emission and free-free absorption always in LTE (cf. script "Physik der Sternatmosphären").

$$S_\nu^{\mathrm{ff}}(\nu) = \frac{\eta_\nu^{\mathrm{ff}}(\nu)}{\kappa_\nu^{\mathrm{ff}}(\nu)} = B_\nu(\nu,T) = \frac{2h\nu^3}{c^2}\frac{1}{\exp\left(\frac{h\nu}{k_\mathrm{B}T}\right)-1} \qquad \left[\frac{\mathrm{erg}}{\mathrm{cm^2\,Hz\,s\,sr}} = \frac{\mathrm{erg}}{\mathrm{cm^2\,sr}}\right] \tag{467}$$

Due to the argument of detailed balancing, the opacity and emissivity are always (also in NLTE):

$$\eta^{\mathrm{ff}}(\nu) = \alpha_{\mathrm{ff}}(\nu,T,Z)\,n_\mathrm{e}\,n_k\,\frac{2h\nu^3}{c^2}\exp\left(-\frac{h\nu}{k_\mathrm{B}T}\right) \qquad \left[\frac{\mathrm{erg}}{\mathrm{cm^3\,Hz\,s\,sr}}\right] \tag{468}$$

$$\kappa^{\mathrm{ff}}(\nu) = \alpha_{\mathrm{ff}}(\nu,T,Z)\,n_\mathrm{e}\,n_k\left[1-\exp\left(-\frac{h\nu}{k_\mathrm{B}T}\right)\right] \qquad \left[\frac{1}{\mathrm{cm}}\right] \tag{469}$$

Note that Eq. (468) divided by Eq. (469) recovers the Planck function Eq. (467).

The cross section $\alpha_{\mathrm{ff}}(\nu,T,Z)\cdot n_\mathrm{e}$ for the free-free absorption is proportional to n_e and depends due to the Maxwellian distribution of the electrons on T. n_k denotes the population number of the ion k with charge Z. In a semi-classic approach the free-free coefficient is a generalized from the Kramers formula

$$\alpha_{\mathrm{ff}}(\nu,T,Z) = \frac{4e_0^6Z^2}{3ch}\left[\frac{2\pi}{3km_\mathrm{e}^3}\right]^{-1/2} T^{-1/2}\,\nu^{-3}\,g_{\mathrm{ff}}(\nu,T,n_\mathrm{e}) \qquad \left[\mathrm{cm^5}\right] \tag{470}$$

$$= 3.692\times10^8\,\mathrm{cm^5}\,Z^2\left(\frac{T}{\mathrm{K}}\right)^{-1/2}\left(\frac{\nu}{\mathrm{Hz}}\right)^{-3} g_{\mathrm{ff}}(\nu,T,n_\mathrm{e}) \tag{471}$$

$$= 1.370\times10^{-23}\,\mathrm{cm^5}\,Z^2\left(\frac{T}{\mathrm{K}}\right)^{-1/2}\left(\frac{\lambda}{\mathrm{cm}}\right)^{3} g_{\mathrm{ff}}(\nu,T,n_\mathrm{e}) \tag{472}$$

The free-free Gaunt factor g_{ff} describes the departure from Kramer's theory. It is close to $g_{\mathrm{ff}}\approx 1$ in visible and near ultra-violet spectrum. For radio waves the gaunt factor is (Chambe & Lantos 1971; Allen 1973) ca.

$$g_{\mathrm{ff}} = 10.6 + 1.90\log_{10}T - 1.26\log_{10}(Z\nu) \tag{473}$$

16.7.1. Implementation

The free-free emissivity and opacities are calculated in `cmfcoop.f` (and also in `coop.f`) with help of the subroutine `gauntff.f`. The subroutine `gauntff.f` uses the values from Berger (1956) and Karzas & Latter (1961). The tabulated ranges for $Z = 1$ are $T = 1577\ldots157\,700\,\mathrm{K}$ and $\lambda = 120\ldots1.2\times10^6\,\text{Å}$, bicubic spline interpolation is performed (up to $Z = 17$). For $\lambda < 120\,\text{Å}$ g_{ff} is set to 1. With these g_{ff}s the opacity and emissivity is calculated.

16.7.2. Additional X-rays for superionization

In PoWR it is possible to add X-rays, which do not emerge from the photosphere, but are rather created in the wind.

We treat only (continuum) free-free emission and absorption (Bremsstrahlung, see Sect. 16.7), as line contributions can be neglected for photon energies above 0.9 keV ($\approx 14\,\text{Å}$) and the considered parameter range (see Baum et al. 1992). Therefore we assume a 2nd, hot plasma component in the wind, which is assumed to be distributed within the cool wind in form of optically thin filaments of temperature T_X.

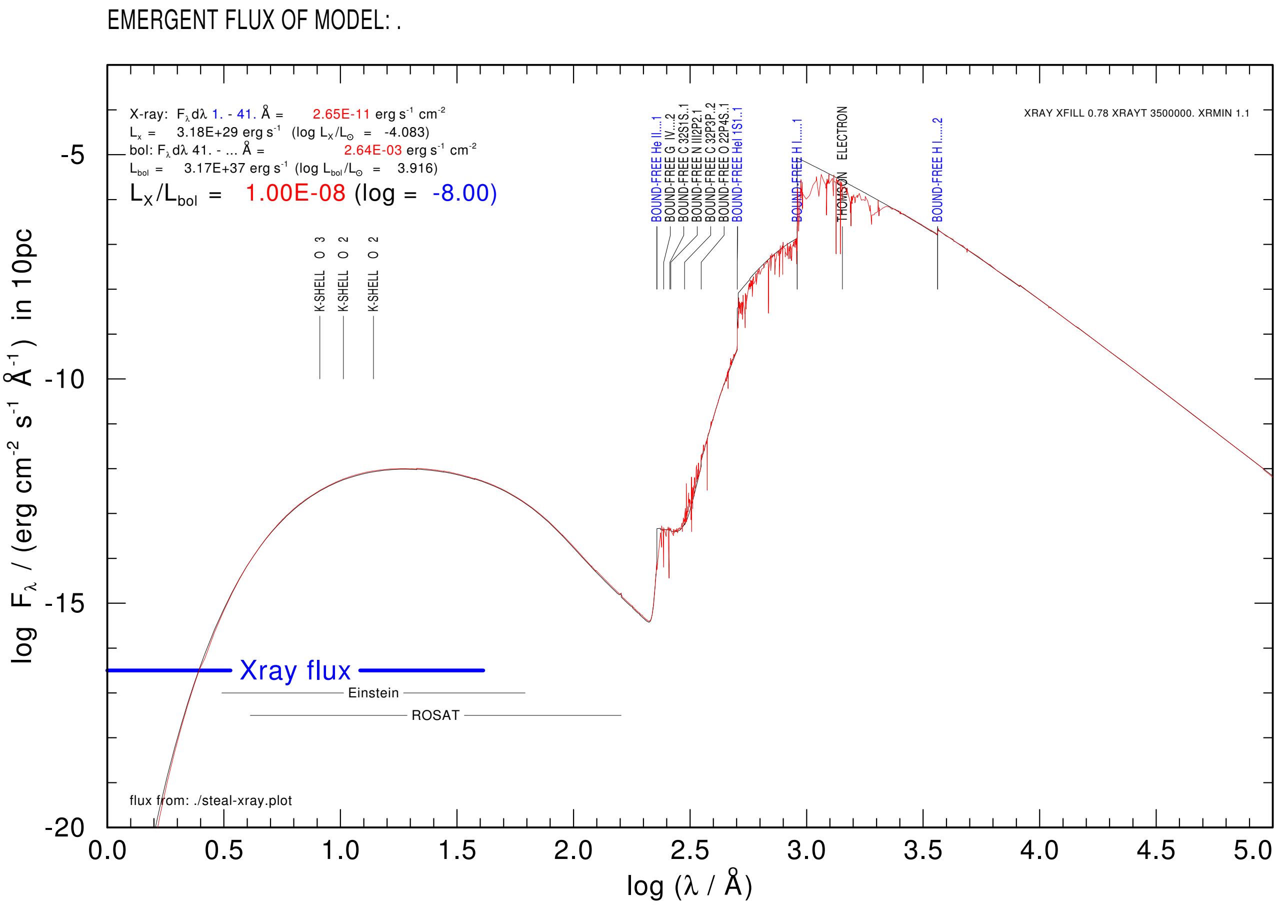


Figure 30: Example for the emergent flux from `steal.plot` with additional X-rays included (between $\log(\lambda) = 0.2$ and $\log(\lambda) = 2.3$). This special plot has been generated by `xrcheck.com` for a B star of $T_{\rm eff} = 23\,$kK.

They are described by three parameters. The first one, `XFILL`, is the X-ray filling factor $X_{\rm fill} = {\rm EM}_{\rm hot}/{\rm EM}_{\rm cool}$, which is related to the emission measure EM. The temperature (only one temperature!) of the hot material, $T_{\rm X}$ is set by the parameter `XRAYT` and is typically in the order of one or a few MK. The filaments are homogeneously distributed ($X_{\rm fill}$ = const.) in the atmosphere, starting at radius `XRMIN` (in R_*). E.g.

```
XRAY XFILL 1.3E-1 XRAYT 1500000. XRMIN 1.1
```

The emissivity η and opacity κ are calculated in `cmfcoop.f` and `coop.f`.
For calculations in `STEAL` → `LINPOP` → `COMA` and `WRSTART` → `GREY` → `OPAGREY`, i.e. in the absence of NLTE population numbers it is assumed that all material is He III.
So in both cases it is $\eta_{\rm ff} \sim n_i\, n_{\rm e}\, X_{\rm fill}$. Therefore `XFILL` is referenced to as the "fraction" (which can be larger than one) of electrons that make additional Bremsstrahlung. Also, the luminosity in X-rays of the hot filaments depends roughly linear on `XFILL`.

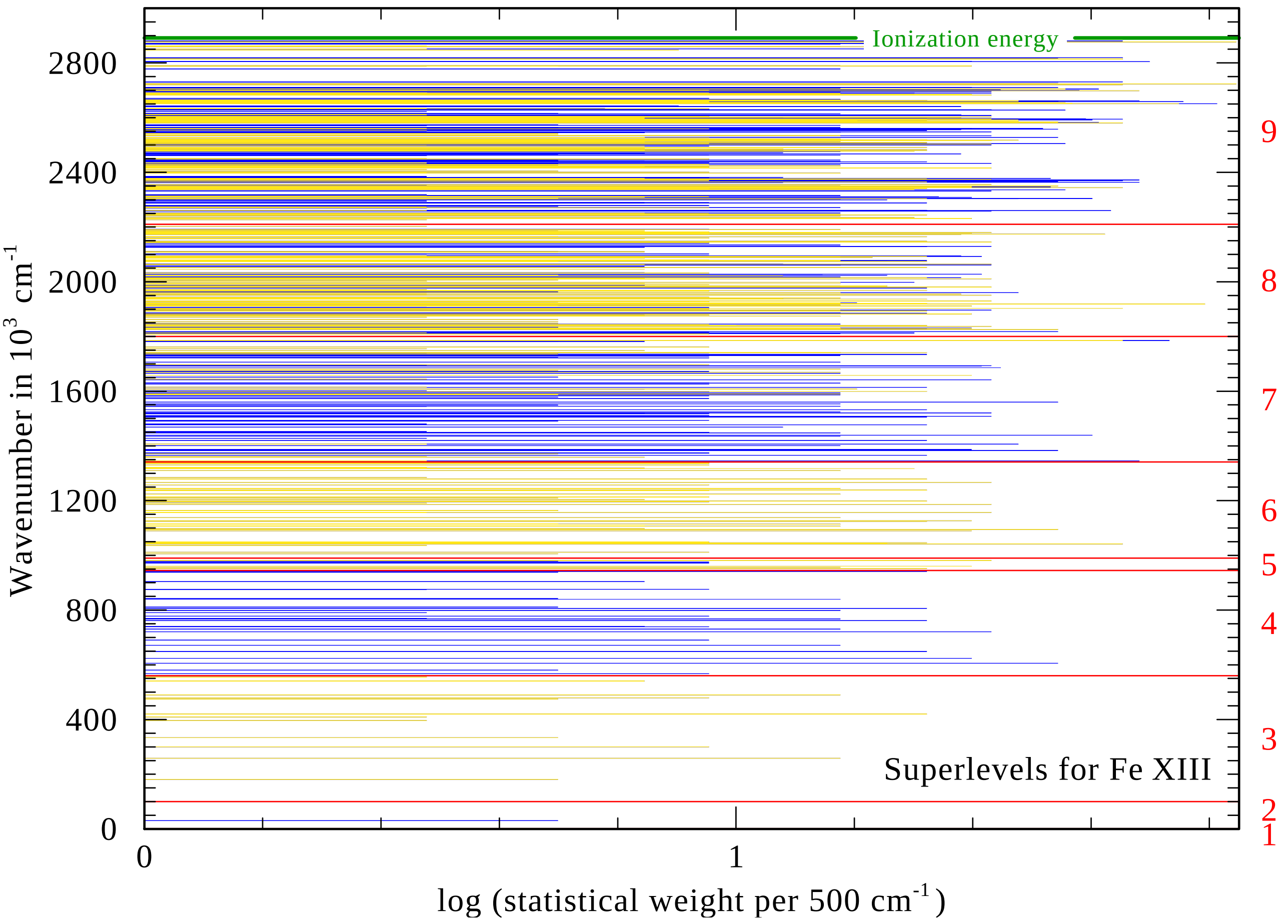


Figure 31: Superlevel assignment for Fe XIII: The sum of the statistical weights of all considered levels is shown in bins of 500 kayser. The yellow lines refer to the contributions of levels with an even parity while the blue color denotes those with odd parity. The red horizontal lines indicate the sorting in a certain superlevel and the green line marks the ionization energy of the ion. The corresponding superlevel indices are given by the red numbers on the right.

16.8. Iron-group elements

16.8.1. Superlevel approach

Iron and other iron group elements are treated in form of one *generic* element, labeled as `ELEMENT G`. The iron-group elements possess many electrons, which leads to thousands of levels and millions of line transitions that cannot be treated in full detail within a stellar atmosphere code. Nevertheless the elements are extremely important as their huge number of lines have a significant "line-blanketing" effect on the radiative transfer and atmospheric structure, even though the combined abundance of these elements is only on the order of $X_{\rm G} \approx 0.1\%$, even for solar metallicity. For PoWR models this has been demonstrated in Gräfener et al. (2002).

This section introduces the basic concepts of the superlevel approach and shows the basics of their their implementation in the PoWR code. For ionization stages below G X, it contains all elements with atomic numbers from 21 to 28, i.e. scandium, titanium, vanadium, chromium, manganese, iron, cobalt, and nickel. The required atomic data is taken from the Kurucz database. For higher ions, where no Kurucz data are available, only iron is accounted for, taking all the required data from TOPbase, the Opacity Project database (Mendoza 1992). The relative abundances of the elements are listed in Gräfener et al. (2002).

In order to summarize levels, which will now be called sublevels here, into superlevels, energy bands have to be defined for each ionization stage. An example for such a superlevel grouping can be seen in Fig. All sublevels that inside each energy band are assumed to have an occupation probability based on an LTE-description using a characteristic temperature T_{exc} for this ion. This temperature is called the *excitation temperature* and has to be given[6]. This allows the define the following quantities, where capital indices always refer to superlevels, while lowercase indices refer to the sublevels which are added up to a superlevel:

$$E_L = \frac{\sum\limits_{i,l} E_l \, a_i \, g_l \exp\left(-\frac{E_l}{kT_{\mathrm{exc}}}\right)}{\sum\limits_{i,l} a_i \, g_l \exp\left(-\frac{E_l}{kT_{\mathrm{exc}}}\right)} \tag{474}$$

$$g_{i,l} = a_i \, g_l \exp\left(\frac{E_L - E_l}{kT_{\mathrm{exc}}}\right) \tag{475}$$

$$G_L := \sum_{i,l} g_{i,l} = \sum_{i,l} a_i \, g_l \exp\left(\frac{E_L - E_l}{kT_{\mathrm{exc}}}\right) \tag{476}$$

The term E_L then denotes the energy of a superlevel, while E_l refers to the sublevel energy. G_L is the weight of the superlevel. It is actually the sum of the generalized sublevel weights g_i, l, which in turn contains the actual weight g_l of the sublevel and the occupation probability. Note that even though PoWR does not use occupation probabilities for normal levels, these factors cannot be neglected for superlevels. Only in the limit of $T_{\mathrm{exc}} \to \infty$ this factor vanishes and the total weight turns into a simple addition of individual weights. The factor a_i describes the relative abundance of a real element in the generic element.

The population number of a sublevel $n_{i,l}$ can be obtained from the corresponding superlevel population number n_L via

$$n_{i,l} = n_L \frac{g_{i,l}}{G_L}. \tag{477}$$

To obtain the transition rates as well as the emissivity and opacity for the superlevels, the particular values have to be summed. In a first step, the individual cross-sections σ_{lu} are added up to the superlevel cross-sections σ_{LU} such that the product of the superlevel weight and the superlevel cross-section matches the sum of the weighted individual cross-sections, i.e.

$$\sigma_{LU} := \frac{1}{G_L} \sum_{i,l,u} g_{i,l} \sigma_{lu}. \tag{478}$$

An example for the complex, wavelength-dependent structure of the resulting superlevel cross sections can be seen in Fig. 32. Due to the exponential terms in the definition of the superlevel weight (476), the equation

[6] In practice, a significant values for the T_{exc} of a particular ion can be determined iteratively by calculating a stellar atmosphere model and check at which electron temperature this ion is the leading ion.

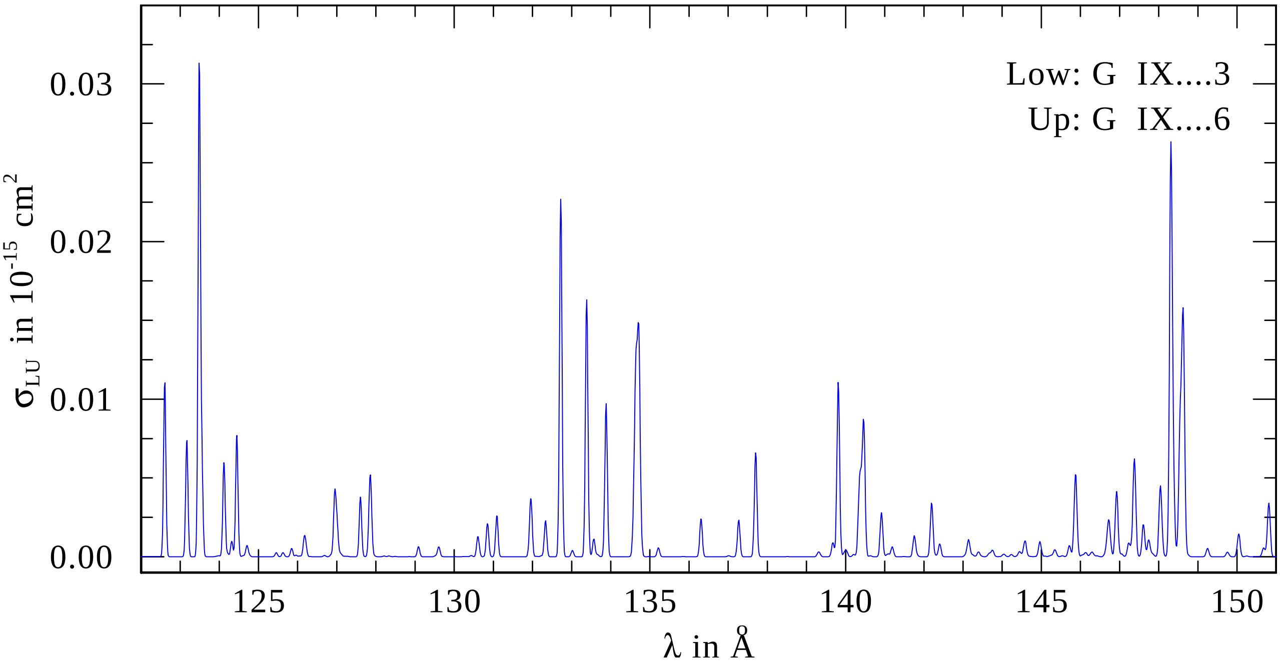


Figure 32: An excerpt of the superlevel cross section σ_{LU} for a transition of the G IX state of the generic element representing the iron group elements.

$\sigma_{lu} g_l = \sigma_{ul} g_u$ is not automatically fulfilled for the superlevels:

$$\sigma_{UL} = \frac{1}{G_U} \sum_{i,l,u} g_{i,u} \sigma_{ul} \tag{479}$$

$$= \frac{1}{G_U} \sum_{i,l,u} a_i g_u \exp\left(\frac{E_U - E_u}{kT_{\mathrm{exc}}}\right) \sigma_{ul} \tag{480}$$

$$= \frac{G_L}{G_U} \frac{1}{G_L} \sum_{i,l,u} a_i \frac{g_l}{g_u} g_l \frac{\exp\left(\frac{E_U - E_u}{kT_{\mathrm{exc}}}\right)}{\exp\left(\frac{E_L - E_l}{kT_{\mathrm{exc}}}\right)} \exp\left(\frac{E_L - E_l}{kT_{\mathrm{exc}}}\right) \sigma_{ul} \tag{481}$$

$$= \frac{G_L}{G_U} \frac{1}{G_L} \sum_{i,l,u} a_i g_l \frac{\exp\left(\frac{E_U - E_L}{kT_{\mathrm{exc}}}\right)}{\exp\left(\frac{E_u - E_l}{kT_{\mathrm{exc}}}\right)} \exp\left(\frac{E_L - E_l}{kT_{\mathrm{exc}}}\right) \sigma_{lu} \tag{482}$$

$$= \frac{G_L}{G_U} \frac{1}{G_L} \sum_{i,l,u} g_{i,l} \exp\left[\frac{h}{kT_{\mathrm{exc}}} (\nu_{UL} - \nu_{ul})\right] \sigma_{lu} \tag{483}$$

It is precisely the exponential term with the brackets that prevents a straight-forward transformation from σ_{LU} to σ_{UL}. The exponential term vanishes only in the limit of an infinite excitation temperature T_{exc} or if $\nu_{ul} = \nu_{UL}$. The latter is approximately true if the superlevels are small enough. However, a look at Fig. 31 illustrates, that for the important ionization stages, this is not possible without losing the advantage of superlevel, i.e. a significant reduction of the total number of levels. Thus, the fact that σ_{UL} cannot be calculated from σ_{LU} as simple as for normal levels has significant consequences and requires a special treatment for the radiative rates, opacities and emissivities in the PoWR code.

In the current blanketing approach in PoWR, which is described in Gräfener et al. (2002), only σ_{LU} is pre-calculated and stored in the iron data file, while σ_{UL} has to be calculated. Based on the fact that only active transitions are considered in the comoving frame, i.e. those where the current integration frequency ν is in a certain range around ν_{UL}, the Eq. (483) is further approximated by replacing ν_{ul} with ν. This allows to get the exponential term out of the sum. Furthermore the excitation temperature T_{exc} is replaced by the current electron temperature T_e, so the expression used in the code is:

$$\sigma_{UL} = \frac{G_L}{G_U}\frac{1}{G_L}\exp\left[\frac{h}{kT_\mathrm{e}}\left(\nu_{UL}-\nu\right)\right]\sum_{i,l,u} g_{i,l}\sigma_{lu} \tag{484}$$

$$= \frac{G_L}{G_U}\exp\left[\frac{h}{kT_\mathrm{e}}\left(\nu_{UL}-\nu\right)\right]\sigma_{LU} \tag{485}$$

This approach has proven to be quite successful in normal models which usually use ions only up to G x. In this regime, the difference between T_e and T_exc is usually not large in the regions where a line is active. The usage of T_e is also motivated in order to regain the LTE limit at the inner boundary. However, this approach seems to have problems when using the higher ions, especially if combined with stellar temperatures above $\approx$ 175 kK. Already below this limit the high ions produce insufficiently high transitions rates, but this could be neglected by implementing a more sophisticated “switch-off” for levels which are not significantly populated. For higher temperatures however, the temperature correction method is also affected and the coupling of the electron temperature T_e with η, κ, and the rates seems to produce situations where the corrections might produce a non-converging or even diverging situation. This is one of the reasons why WO models are hard to calculate, which will later become important when discussing a hydrodynamically consistent WO model in Sect. **??**.

Regardless of how accurate the terms of σ_{UL} are produced, the way of calculating the superlevel rates as well as the line opacities and emissivities are always the same. The line opacities of the superlevels can than be calculated by simply adding up their sublevel line opacities:

$$\kappa_{lu} = n_l\sigma_{lu}\left(1-\frac{n_u}{n_l}\frac{g_l}{g_u}\right) = n_l\sigma_{lu} - n_u\sigma_{ul} \tag{486}$$

$$\kappa_{LU} := \sum_{l,u}\kappa_{lu} = \sum_{i,l,u} n_{i,l}\sigma_{lu}\left(1-\frac{n_{i,u}}{n_{i,l}}\frac{g_l}{g_u}\right) \tag{487}$$

$$= \frac{n_L}{G_L}\sum_{i,l,u} g_{i,l}\sigma_{lu}\left(1-\frac{n_U}{n_L}\frac{g_{i,u}}{G_U}\frac{G_L}{g_{i,l}}\frac{g_l}{g_u}\right) \tag{488}$$

$$= \frac{n_L}{G_L}\sum_{i,l,u} g_{i,l}\sigma_{lu} - \frac{n_U}{G_U}\sum_{i,l,u} g_{i,u}\sigma_{lu}\frac{g_l}{g_u} \tag{489}$$

$$= \frac{n_L}{G_L}\sum_{i,l,u} g_{i,l}\sigma_{lu} - \frac{n_U}{G_U}\sum_{i,l,u} g_{i,u}\sigma_{ul} \tag{490}$$

$$= n_L\sigma_{LU} - n_U\sigma_{UL} \tag{491}$$

In a similar way the superlevel line emissivities can be calculated:

$$\eta_{lu} = \frac{2h\nu_{lu}^3}{c^2} n_u \frac{g_l}{g_u}\sigma_{lu} \tag{492}$$

$$\eta_{LU} = \sum_{l,u}\eta_{lu} = \sum_{i,l,u}\frac{2h\nu_{lu}^3}{c^2} n_{i,u}\frac{g_l}{g_u}\sigma_{lu} \tag{493}$$

$$= \frac{2h}{c^2}\frac{n_U}{G_U}\sum_{i,l,u}\nu_{lu}^3 g_{i,u}\frac{g_l}{g_u}\sigma_{lu} \tag{494}$$

The last expression can be reduced further, if one assumes that $\nu_{lu} \approx \nu_{LU}$ and hence

$$\eta_{LU} \approx \frac{2h\nu_{LU}^3}{c^2}\frac{n_U}{G_U}\sum_{i,l,u} g_{i,u}\sigma_{ul} \tag{495}$$

$$= \frac{2h\nu_{LU}^3}{c^2} n_U\sigma_{UL} \tag{496}$$

The to total radiative transition rate is the sum of all subrates. Using this requirement and inserting Eq. (477) leads to

$$n_L R_{LU} = \sum_{i,l,u} n_{i,l} R_{lu} = n_L \sum_{i,l,u} \frac{g_{i,l}}{G_L} R_{lu} \tag{497}$$

which immediately provides the formula how to add up the rate coefficients of the sublevels:

$$R_{lu} = 4\pi \int\limits_0^\infty \frac{\sigma_{lu}}{h\nu} J_\nu \mathrm{d}\nu \tag{498}$$

$$R_{LU} = \sum_{i,l,u} \frac{g_{i,l}}{G_L} R_{lu} \tag{499}$$

$$= 4\pi \sum_{i,l,u} \frac{g_{i,l}}{G_L} \int\limits_0^\infty \frac{\sigma_{lu}}{h\nu} J_\nu \mathrm{d}\nu \tag{500}$$

$$= 4\pi \int\limits_0^\infty \frac{J_\nu}{h\nu} \sum_{i,l,u} \frac{g_{i,l}}{G_L} \sigma_{lu} \mathrm{d}\nu \tag{501}$$

$$= 4\pi \int\limits_0^\infty \frac{J_\nu}{h\nu} \sigma_{LU} \mathrm{d}\nu \tag{502}$$

The same scheme can be used for slightly more complex R_{UL}:

$$R_{ul} = 4\pi \frac{g_l}{g_u} \int\limits_0^\infty \frac{\sigma_{lu}}{h\nu} \left[\frac{2h\nu^3}{c^2} + J_\nu \right] \mathrm{d}\nu \tag{503}$$

$$R_{UL} = \sum_{i,l,u} \frac{g_{i,u}}{G_U} R_{ul} \tag{504}$$

$$= 4\pi \sum_{i,l,u} \frac{g_{i,u}}{G_U} \frac{g_l}{g_u} \int\limits_0^\infty \frac{\sigma_{lu}}{h\nu} \left[\frac{2h\nu^3}{c^2} + J_\nu \right] \mathrm{d}\nu \tag{505}$$

$$= 4\pi \int\limits_0^\infty \frac{1}{h\nu} \left[\frac{2h\nu^3}{c^2} + J_\nu \right] \sum_{i,l,u} \frac{g_{i,u}}{G_U} \sigma_{ul} \mathrm{d}\nu \tag{506}$$

$$= 4\pi \int\limits_0^\infty \frac{1}{h\nu} \left[\frac{2h\nu^3}{c^2} + J_\nu \right] \sigma_{UL} \mathrm{d}\nu \tag{507}$$

The Eqs. (502) and (507) show that the radiative transition rates for the superlevels have the same form as those for normal levels, except that the cross-sections σ_{ul} and σ_{lu} are replaced by their superlevel counterparts σ_{UL} and σ_{LU}, which means no additional assumptions or approximations have to be made beside those entering the calculation of the superlevel cross sections. However, for the normal rates the $(h\nu)^{-1}$-term is often approximated to $(h\nu_{ul})^{-1}$ and thus can be put in front of the frequency integral. This is also the case in the PoWR code, where the "normal" radiative rates are calculated using the coarse frequency grid. For the rates of the generic element however, the $(h\nu)^{-1}$ is kept and the rates are calculated on the fine frequency grid, in parallel with the CMF radiative transfer.

For a start approximation which has to be available before the first radiative transfer calculation, the rates of the generic element are also calculated in the approximated form as it is done for the other rates. This is also done during the solution of the statistical equations, where the have to be recalculated based on updated population numbers. Here the more accurate rates from the CMF calculations are used, but afterwards updated by adding the differences between the approximated rates. By using only the differences of the rates and not their absolute value, the error should be kept small.

A special features for iron line transitions accounts for the fact that the temperature might have been updated since the radiation field has been calculated.

HERE: INPUT FROM ANDREAS on the TCORR FERAT factor!

16.8.2. The FEDAT files

The iron-group elements are treated as one generic model atom with superlevels and superlines. The corresponding atomic data are prepared with the *Blanketing* program (see Sect. 16.8.1). The data are provided to the PoWR code in form of the name-indexed mass-storage file FEDAT or FEDAT_FORMAL. This file is decoded by the subroutine FEDAT which is called from DATOM in each of the PoWR main programs.

In the FEDAT file, the lower and upper superlevel index is encoded in strings named A*nn*aNAM, where *nn* is the ion charge. The level names are stored in vectors (one vector per ion) LEVN*nn*a, only since parity splitting is taken into account. The number of superline transitions in each of the ions are stored in the FEDAT-file in vector NTRA_A (Fortran variable: NTRA)

The bound-bound cross sections are stored in the FEDAT file in variables with names of the kind A*nn*a*mmm*, where *nn* is the ion charge and *mmm* is the line transition index (counted within this ion). The cross-sections are tabulated over a logarithmic frequency grid which is consistently defined in the *Blanketing* program. The grid index k translates into wavelength λ_k as

$$\lambda_k = \lambda_0 \;\cdot\; 10^{k*\mathtt{XLOGSTEP}} \quad \text{with } \mathtt{XLOGSTEP} = \log\left(1 + \frac{v_\mathrm{Dop}}{c}\,\mathtt{DX}\right)$$

The parameters defining the wavelength grid are communicated via the FEDAT file.

Variable	Mass-storage name	Fortran variable	comment
v_Dop	VDOPP	VDOPFE	
DX	FSTEP	DXFE	spacing in Doppler units
λ_0	XLAMNULL	XLAM0FE	
ASCII-coded level indices	A*nn*aNAM		ion charge *nn*; 1 entry per line
level names	LEVN*nn*a	LEVNAMES	new since parity splitting
mean level energies	ELEV*nn*a	ELEVEL	vector; cf. Eq. (2.65) in Diss. Sander
superlevel stat. weight	WEIG*nn*a	WEIGHT	vector; cf. Eq. (2.67) in Diss. Sander
No. of superlines	NTRA_A	NTRA	vector, 1 entry per ion
$\sigma_{\ell u}$	A*nn*a*mmm*	SIGMAFE	ion charge *nn*, line index *mmm*
1st freq. index	A*nn*a*mmm*(2)	IFRBSTA(IND)	neg. index k_min
last freq. index	A*nn*a*mmm*(1)	IFRBEND(IND)	neg. index k_max
No. of freq. in $\sigma_{\ell u}$	N*nn*a_A	NX_A	vector

The first entry in A*nn*a*mmm* gives (with a minus sign) the frequency index k_max of the first cross-section entry; the second entry gives the frequency index k_min of the last cross-section entry. FEDAT stores these numbers in the variables (with negative sign!!) IFRBEND(IND) and IFRBSTA(IND), respectively. The following NX_A(IND) elements of vector A*nn*a*mmm* contain the values of the superline cross section. Note that the first element refers to the largest wavelength (lowest frequency), i.e. the values correspond to a falling sequence of frequency index k.

Redundantly, the number of cross section data points is also stored in the FEDAT vector `N`*nn*`a_A` (Fortran variable name: `NX_A`(*i*), for the current ion).

In the Fortran program (subroutine FEDAT), all bound-bound cross sections are stored in a single big vector SIGMAFE, now in inverse order (i.e. index increases with wavelength). The first index for each particular line transition `IND` inside the vector `SIGMAFE` is stored in the vector element `INDRB(IND)`.

Subroutine `CMFFEOP` is called for a specific wavelength `XLAMK` and returns the iron bound-bound opacity and emissivity vectors (over radius points). Internally, it loops over all radius points. For each radius index, the cross-sections `SIGMAFE` of all lines that are "active" at this wavelength are multiplied with the respective population numbers and added up into the vectors `OPA` and `ETA`. The "active" lines have been identified before by the subroutine `FECHECK`.

In order to get the cross section for a given transition at a particular wavelength `XLAMK`, the corresponding dimensionless wavelength x = `XINDF` (real number!) on the iron-wavelength scale is calculated. For each (active) superline, this index `XINDF` is compared with its start index `IFRBSTA` and the relative index of the next-lower and next-higher grid point is determined. A linear interpolation is then performed to obtain the cross section `SIGMA`.

Note that the program `COLI` calls the subroutine `CMFFEOP` for a each wavelength separately. In contrast, the formal integral (program `FORMAL`) requires the opacities and emissivities over the whole wavelength range instantaneously, in order to calculate the electron-scattering redistribution. Therefore, the subroutine `FORMCMF` calls `CMFFEOP` in a frequency loop and stores the opacities and emissivities in two-dimensional arrays.

16.8.2.1. The effect of microturbulence on iron-line blanketing Iron data (`FEDAT`-Files ara available for different values of the Doppler-broadening velocity VDOP (usually encoded in their filename). For the interation of the MODEL, it makes sense to chose the same VDOP for the FEDAT version as for all other lines (`VDOP` parameter in the `CARDS` file).

For the spectrum synthesis (program `formal`) the microturbulence is usually chosen much smaller and dictated by the width of observed absorption-line profiles (talking here about OB-star models, not abaout WR stars). If the FEDAT file was established for smaller VDOP than the minimum Doppler-broadening required by the thermal and microturbulent velocities, the iron lines are degraded automatically. In the opposite case, i.e. if the iron data have a too large Doppler broadening, the `formal` program can only issue a warning. However, in this case the "iron forest" might be over-estimated, s demonstrated in Fig. 33.

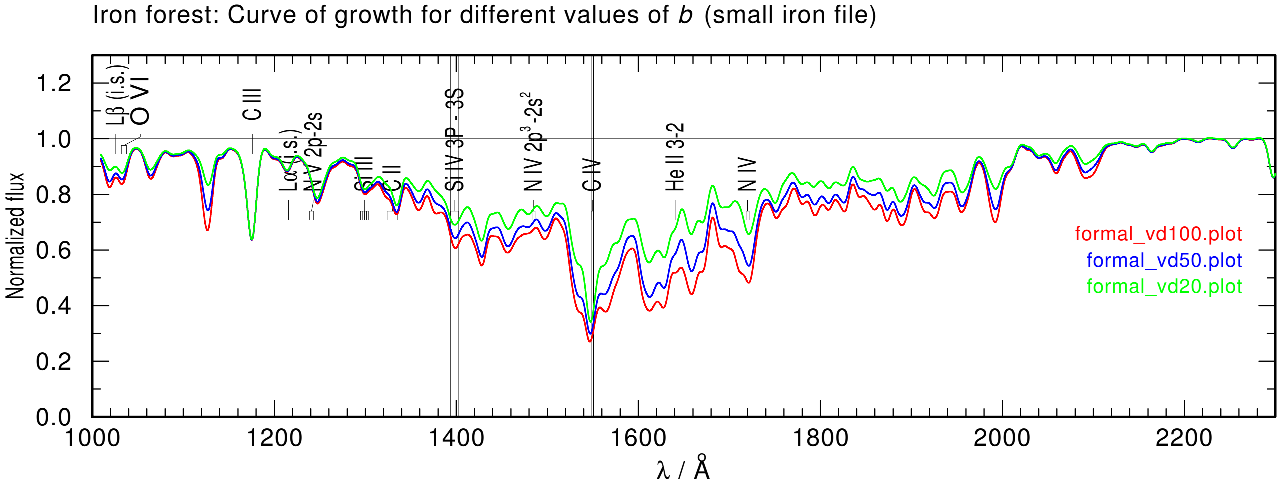


Figure 33: The influence of different values of VDOP (only in iron data) on the iron forest. The model shown is B31_MDOT-10_LOGL4.8_LOGG3.97_Si0.1 and was calculated with VDOP=100 km/s in the wruniq iteration, while the formal was calculated with VDOP=30 km/s. Only the FEDAT_FORMAL is different according to the key in the figure. In each case the so-called SMALL version was used. This version contains only observationally confirmed iron lines (i.e. with correct wavelengths), while the BIG version also includes theoretically predicted iron lines without observational confirmation. The BIG file is used in the wruniq iteration. The differences between BIG and SMALL version increase towards shorter wavelengths, especially below the 1240 Å the BIG file provides much more lines than the SMALL version. The model spectra were convolved with a Gaussian of FWHM of 10 Å to emphasize the effect on the equivalent widths.

Part III.
Organization of the Code

UNDER CONSTRUCTION!

In PoWR, the MODEL is established by iterating the radiative transfer (program COLI) with the statistical equations (program STEAL).

17. The MODEL file

In a PoWR run, a chain of main programms is executed. The information is passed from one program to the next with the help of the file `MODEL`, which is a name-indexed mass-storage file. Such binary file cannot be read with an editor, but we have a couple of tools to extract content from such files, see Sect. 4. FORTRAN does not allow to open random-acess files under a filename; the file `MODEL` is assigned to channel 3 (`fort.3`).

In particular, file `MODEL` contains the following records:

`POPNUM` ...

`XJL`*n* meaning $\bar{J}_{\mathrm{IND}}$, where *n* is the 4-digit or 5-digit line index `IND`, see Sect. 18.

18. Reading atomic data

Each main program calls `SUBROUTINE DATOM`, which reads the atomic-data input. For the "normal" elements these data are provided by the ASCII file also called `DATOM`, while the data for the "generic element" (iron group) are to be decoded from the random mass-storage file `FEDAT` alias `fort.21`.

The following variables are established for the "normal" elements:

- An index `I` is assigned to each atomic energy level, specified by a `LEVEL` entry, in the sequence of its occurence in the `DATOM` file. The levels must be grouped per `ELEMENT`, and be sorted by energy within each ion.

  The following vectors are filled for `I=1 ... N`:
  `LEVEL(I)` 10-character name of the level
  `NCHARG(I)` charge of the ion to which this level belongs
  `WEIGHT(I)` statistical weight of this level
  `ELEVEL(I)` energy of the level above the gound level (in Kayser = wavenumber per cm)
  `EION(I)` ionization energy (in Kayser) from the ground state to which this level belongs
  `MAINQN(I)` pricipal quantum number of this this level
  `NA(I)` index of the element

- A line-index `IND` is assigned to each bound-bound transitions in the sequence of the occurence of a `LINE` entry in the `DATOM` file. Note that *all* possible bound-bound transitions need to be covered with a `LINE` entry. *Rudimental* lines are included in this index list.

  The following vectors are filled for `IND=1 ... LASTIND`:
  `INDNUP(IND)` index of the upper level of this line transition
  `INDLOW(IND)` index of the lower level of this line transition
  `EINST(NUP,LOW)` Einstein coeff. A_u, ℓ
  rudimental lines are marked by -2. in the transposed matrix element `EINST(LOW,NUP)`
  `KEYCBB(IND)` the keyword specifying the formula for collisional bound-bound (CBB) cross section (see Sect. 16.1)
  `COCO(k,IND)` (k=1...4) up to four coefficients to supply the respective CBB formula

- A continuum-index `KONT` is assigned to each bound-free transitions in the sequence of the occurence of a `CONTINUUM` entry in the `DATOM` file.

The following vectors are filled for KONT=1 ... LASTKON:
KONTLOW(KONT) index of the lower level of this bound-free transition
KONTNUP(KONT) index of the upper level of this bound-free transition; if not specified explicitely in the DATOM input file, this is the groundlevel of the next-higher ion
EINST(KONTLOW,KONTNUP) photo-cross section at the threshold in 10^{-18} cm^2
IGAUNT(KONT) keyword specifying which of the different formulas for the radiative bound-free (RBF) cross sections (see Sect. 16.4) is applied
ALPHA(KONT), SEXPO(KONT) parameters to supply the applied formula for the radiative bound-free cross sections
KEYCBF(KONT) keyword specifying which of the different formulas for the collisional bound-free (CBF) cross sections (see Sect. 16.2) is applied
ADDCON1(KONT), ADDCON2(KONT), ADDCON3(KONT) additional parameters for certain CBF formulas

- For K-SHELL ionization (if specified in file DATOM by K-SHELL entries), the following arrays are filled (cf. Sect. 16.5):
SIGMATHK(NATOM,ISTAGE) threshold photo cross-section in 10^{-18} cm^2, where NATOM is the index of the element, and ISTAGE the ionization stage (1 = neutral)
SEXPOK(NATOM,ISTAGE) exponent for wavelength-dependence
EDGEK(NATOM,ISTAGE) energy of the K-shell edge in Kayser

- Stabilizing lines of dielectronic recombination channels connect an autoionizing level (doubly excited state above the ionization threshold) with a normal bound state.

  They are specified by DRTRANSIT entries in the DATOM file; an index IAUTO=1 ... NAUTO is assigned to each of these entries describing a stabilizing line transition. The following vectors are filled:

  LOWAUTO(IAUTO) index of the lower level
  LEVUPAUTO(IAUTO) name of the upper (auto-ionizing) level
  AAUTO(IAUTO) Einstein coeff. $A_{u,\ell}$
  KRUDAUTO(IAUTO) value 1 if line flagged as *rudimental*, 0 else (default);
  IONAUTO(IAUTO) index of the level reached by auto-ionization; if not specified explicitely, this is the ground level of the next-higher ion
  WAUTO(IAUTO) statistical weight of the upper (auto-ionizing) level
  EAUTO(IAUTO) energy [Kayser] of the upper (auto-ionizing) level, relative to the ionization threshold

  Note that a specific auto-ionizing level can appear as upper level in many DRTRANSITs; therefore, entries with the same upper-level name LEVUPAUTO *must* agree in the given statistical weight WAUTO and its energy EAUTO. This is checked in wrstart.

  For the dielectronic transitions, the indices for the lower level and the upper level (after autoionization) are appended to the line indices in the range LASTIND+1 ... LASTINDAUTO:

```
DO IND=1, NAUTO
  INDLOW(LASTIND+IND) = LOWAUTO(IND)
  INDNUP(LASTIND+IND) = IONAUTO(IND)
ENDDO
```

  Moreover, SUBROUTINE APPEND_AUTOLEVELS which is called by subr. DATOM appends the auto-ionizing levels to those vectors that specify the energy levels (index range N+1 ... N_WITH_DRLEVELS). This includes all attributes that are assigned to each energy level (see above). An autoionizing level which occurs repeatedly in DRTRANSIT entries leads to only one entry in the level list.

  The huge amount of atomic data describing the *generic element*for the iron group, which is treated in the *superlevel approach* (cf. Sect. 16.8.1), are provided by the mass-storage file FEDAT alias fort.21. This file is decoded by SUBROUTINE FEDAT which is called from subr. DATOM when the entry ELEMENT

GENERIC is encountered in file DATOM towards its end. For all ions in the range which is requested on this ELEMENT GENERIC entry, the following arrays are filled:

ELEVEL (appended to the vector behind LASTIND) WEIGHT (appended to the vector behind LASTIND) LEVNAMES SIGMAFE (compact vector with all superline bound-free cross-sections) IFENUP IFELOW superline indices

subr. FEDAT appends analogues LASTFE entries to the vectors INDNUP(IND), INDLOW(IND) as well as the matrix EINST(NUP,LOW). –> max. index LASTIND is increased: LASTIND = LASTIND + LASTFE

19. Initialization

in WRSTART :

in WRSTART - JSTART: initial rad. field written to MODEL file; for lines: XJLnnnn with nnnn (I4) = IND; if IND > 9999 : XJLnnnnn (I5)

RUD lines *are* skipped (no XJL entry written): IF (EINST(I,J) .EQ. -2.) GOTO 99 jumps behind CALL WRITMS (3,XJL ...

Since the loop runs over all line indices IND=1, LASTIND starting values for XJL are written including all iron superlines

20. Radiative transfer

Main program COLI

COLI - DECCOLI:

reads an obsolete option LINE ... in many old CARDS files: LINES: ALL which means IND1=0, IND2=LASTIND default: also like ALL –> this option might be deleted !! at the end of deccoli: effectively LINE(IND) = IND

COLI - SEQLINECL sorts the lines by increasing wavelengths

Rud. lines are skipped here!!! NLINE (input) becomes reduced by the number of rud.lines: NLINE=NLINE-NUMRUD

COLI has a big frequency loop (index K)

Before thet loop:

COLI - PREPK initialized (among others): K = 0 LINECHECK = 1 ILINECHECK = LINE(ILINECHECK)

Note: for iron this initalization is done directly in COLI: INDFEACT(1) = 1

Insside thie loop: COLI - CHECK_LINES : For the current K, those lines wich are "active" at this frequency are asembled in vector LIND(NL), NL= 1 ... MAXLIN

MAXLIN is a dimension parameter set in the main program COLI. Unused entries in the LIND vector are set to zero. The total number of lines that are active at any given frequency (NACT) thus cannot exceed MAXLIN (else: error stop).

A line is checked out when the loop wavelenth has passed its BANDWIDTH. If this happens for the line with index LACT=LIND(NL), the mean intensity XJL is written to the model file:

WRITE (NAME,'(A3,I4,A1)') 'XJL',LACT,' ' CALL WRITMS > (3, XJLMEAN(1,NL), ND, NAME, -1, IDUMMY, IERR)

where LIND(NL) is the original (from DATOM) line index

After beeing checked-out, LIND(NL) is set to zero (i.e. this entry is now free and can be re-used for another line to be checked-in.

In effect, XJL.... entries for rud. lines are *not* written by COLI

After the last XJL-entry for lines of explicit atoms (index LASTIND), the next LASTFE entries contain the corresponding mean intensities XJFEMEAN for the superlines.

These XJL entries for iron bands are written to MODEL by subr. WMODCOLI

Part IV.
Installation Manual

21. PoWR-Code-Skripte

21.1. wrstart8

```
#!/bin/ksh
echo $HOSTNAME > $HOME/work/scratch/wruniq8/fwhere
.  $HOME/work/wrjobs/wrstart_helge 8 $1
```

21.2. wruniq8

```
#!/bin/ksh
.  $HOME/work/wrjobs/wruniq_helge 8 $1
```

21.3. set_repeat8

```
# Set_Repeat8
#
cd $HOME/work/wrdata8
echo 'REPEAT' > next_job
echo 'REPEAT' > next_jobz
```

21.4. njn8

```
#!/bin/ksh
.  $HOME/work/wrjobs/njn_neu_helge 8
```

22. List of routines

FLAG_ZERORATES returns a logical vector with index over all levels, indicating whether the level is switched off due to POPMIN criterion. Flagging is only allowed if the level above is already flagged.

NLTEPOP called by ... for solving the linear rate equations. The rate equations are only linear in the case of

- first `steal` in wrstart-job
- all `GAMMA`s off, `PRINT RATES` enabled

23. Compilation of PoWR code

You need the libraries, the scripts and the FORTRAN compiler for generating new exe-Files.

Usually you will find in ~wrh/libraries.dir the libraries

- libcr_cl.tlb and libcr_cl.a containing the .f and .o files resp.
- libcr_add.tlb and libcr_add.a

You can also have your own library, e.g. libcr_user.tlb and libcr_user.a.

When using the scripts in ~wrh/proc.dir you have to omit all suffixes after the dot.

23.1. The way it works - Example

- create your own ~/libraries.dir
- in this directory you can create your own library
- you need the file loadlibs for compiling (s. below)
- create your own ~/proc.dir and copy therein the following scripts (chmod ux *.com+):
  - work.com
  - index.com
  - get.com
  - replace.com
  - link.com
  - linkopt.com

  and alias them to the shortend forms work, index, etc.
- pay attention, that you have the adequate compiler, e.g. the f90 on the alphas or intel on

A. Atomic Theory

A.1. Levels – energy order

$1s$	$2s$	$3s$	$4s$	$5s$	$6s$	$7s$	$8s$	$9s$	$10s$
	$2p$	$3p$	$4p$	$5p$	$6p$	$7p$	$8p$	$9p$	$10p$
		$3d$	$4d$	$5d$	$6d$	$7d$	$8d$	$9d$	$10d$
			$4f$	$5f$	$6f$	$7f$	$8f$	$9f$	$10f$
				$5g$	$6g$	$7g$	$8g$	$9g$	$10g$
					$6h$	$7h$	$8h$	$9h$	$10h$
						$7i$	$8i$	$9i$	$10i$
							$8k$	$9k$	$10k$
								$9l$	$10l$
									$10m$

Table 4: For determining the energy order of terms of a one-electron system, just go along the diagonals, e.g. 1s 2s 2p 3s 3p 4s 3d 4p 5s.

A.2. L-S coupling

$$\mathbf{J} = \mathbf{L} + \mathbf{S} \tag{508}$$

$$\mathbf{L} = \sum_i l_i \tag{509}$$

A.2.1. Addition of angular momenta

$\mathbf{l_1}$	$\mathbf{m_{l_1}}$	$\mathbf{l_2}$	$\mathbf{m_{l_2}}$	$\mathbf{m_l}$	$L = 3$	2	1
2	2	1	1	3	x		
	2		0	2	x	x	
	1		1	2	x	x	
	2		-1	1	x	x	x
	1		0	1	x	x	x
	0		1	1	x	x	x
	1		-1	0	x	x	x
	0		0	0	x	x	x
	-1		1	0	x	x	x
	0		-1	-1	x	x	x
	-1		0	-1	x	x	x
	-2		1	-1	x	x	x
	-1		-1	-2	x	x	
	-2		0	-2	x	x	
	-2		-1	-3	x		

Table 5: Magnetic quantum numbers of an $l = 2$ (d) electron plus an $l = 1$ (p) electron, from (Cowan 1981, p. 52)

A.2.2. Selection rules for dipole radiation

1. In general:

   $\Delta J = 0,\ \pm 1$ but no $(J = 0) \rightarrow (J = 0)$
   $\Delta m_J = 0,\ \pm 1$ but no $(m_J = 0) \rightarrow (m_J = 0)$, if $\Delta J = 0$

2. LS-coupling:

   $\Delta S = 0$
   $\Delta L = 0,\ \pm 1$
   $\Delta \ell = \pm 1$ for the transient electron (so $\Delta L = \pm 1$ for one-electron transitions)

3. jj-coupling:

   $\Delta j = 0,\ \pm 1$ for one electron
   $\Delta j = 0$ for all others

4. In general – change of parity:

   for all dipole transitions: $\Delta P = \pm 1$

B. Quadrature sums

B.1. General

Integrals are numerically solved as quadrature sums, i.e.

$$I = \int_a^b f(x)\,\mathrm{d}x \tag{510}$$

is evaluated as

$$I = \sum_i w_i\, f_i \tag{511}$$

where $f_i = f(x_i)$, and w_i are the quadrature weights. Let us for the following consider only a small interval (a, b) between just two points; for a sum over many intervalls, at each inner point two weights from both adjacent intervalls apply to the same f_i and can be added.

The trapezoidal rule yields weights $w_a = \frac{1}{2}(b-a)$ and $w_b = \frac{1}{2}(b-a)$. The quadrature sum with these weights gives the exact integral for a linear function $f(x)$.

Now we want to achive the same accuracy for the integral

$$I = \int_a^b f(x)\, g(x)\,\mathrm{d}x \tag{512}$$

where the integral contains a weight function (or "kernel") $g(x)$. For this purpose, the weight function is accounted for in the quadrature weights. Before considering specific weight functions, we first derive the general formalism.

We define

$$\begin{aligned}
\Delta &= b-a \\
m &= 1/2\,(b+a) \\
x &= m\; +\; r\Delta/2 \\
dx &= \Delta/2\,dr
\end{aligned}$$

A linear interpolation of $f(x)$ between $f_a = f(a)$ and $f_b = f(b)$ is then the function

$$f(r) = f^{\mathrm{e}}\; +\; f^{\mathrm{o}} \tag{513}$$

with the even (with respect to r) term

$$f^{\mathrm{e}} = \frac{f_b + f_a}{2} \tag{514}$$

and the odd term

$$f^{\mathrm{o}} = r\,\frac{f_b - f_a}{2} \tag{515}$$

The weight function is also split into an even and an odd (with respect to r) part

$$\begin{aligned}
g(x) &= g^{\mathrm{e}}(r) + g^{\mathrm{o}}(r) \\
g^{\mathrm{e}}(r) &= \frac{1}{2}\left[g(m + \frac{\Delta}{2}r)\; +\; g(m - \frac{\Delta}{2}r)\right] \\
g^{\mathrm{o}}(r) &= \frac{1}{2}\left[g(m + \frac{\Delta}{2}r)\; -\; g(m - \frac{\Delta}{2}r)\right]
\end{aligned}$$

The integral is now

$$I = \frac{\Delta}{2}\int_{-1}^{1} (f^{\mathrm{e}} \; + \; f^{\mathrm{o}}(r))\,(g^{\mathrm{e}}(r) + g^{\mathrm{o}}(r))\,\mathrm{d}r \tag{516}$$

Because of the symmetric interval, only the even-even and the odd-odd products contribute to the integral, i.e.

$$I = \frac{\Delta}{2}\int_{-1}^{1}\left[\frac{f_b + f_a}{2}\,g^{\mathrm{e}}(r) \; + \; \frac{f_b - f_a}{2}\,r\,g^{\mathrm{o}}(r)\right]\mathrm{d}r \tag{517}$$

Therefore we define

$$\begin{aligned} G &\equiv \int_0^1 g^{\mathrm{e}}(r)\,\mathrm{d}r \\ H &\equiv \int_0^1 r\,g^{\mathrm{o}}(r)\,\mathrm{d}r \end{aligned}$$

This makes the integral to

$$I = \frac{\Delta}{2}\left[(f_b + f_a)\,G \; + \; (f_b - f_a)\,H\right] = w_a f_a + w_b f_b \tag{518}$$

with the quadrature weights being

$$w_a \; = \; \frac{\Delta}{2}(G - H) \quad \text{and} \quad w_b \; = \; \frac{\Delta}{2}(G + H) \tag{519}$$

B.2. Linear weight function

$$\begin{aligned} g(x) \; &= \; x \quad \Longrightarrow \quad G \; = \; m, \quad H \; = \; \frac{\Delta}{6} \\ w_a \; &= \; \frac{\Delta}{2}\left(m - \frac{\Delta}{6}\right), \quad w_b \; = \; \frac{\Delta}{2}\left(m + \frac{\Delta}{6}\right) \end{aligned}$$

B.3. Quadratic weight function

$$g(x) \; = \; x^2 \quad \Longrightarrow \quad G \; = \; m^2 + \frac{\Delta^2}{12}, \quad H \; = \; m\frac{\Delta}{3} \tag{520}$$

$$w_a = \frac{\Delta}{2}\left(m^2 + \frac{\Delta^2}{12} - m\frac{\Delta}{3}\right), \quad w_b = \frac{\Delta}{2}\left(m^2 + \frac{\Delta^2}{12} + m\frac{\Delta}{3}\right), \tag{521}$$

B.4. Cubic weight function

$$g(x) \; = \; x^3 \quad \Longrightarrow \quad G \; = \; m^3 + \frac{1}{4}m\Delta^2, \quad H \; = \; \frac{1}{2}m^2\Delta + \frac{\Delta^3}{40} \tag{522}$$

$$w_a \; = \; \frac{\Delta}{2}\left(m^3 + \frac{1}{4}m\Delta^2 - \frac{1}{2}m^2\Delta - \frac{\Delta^3}{40}\right), \quad w_b \; = \; \frac{\Delta}{2}\left(m^3 + \frac{1}{4}m\Delta^2 + \frac{1}{2}m^2\Delta + \frac{\Delta^3}{40}\right) \tag{523}$$

B.5. Inverse-x weight function

$$g(x) = x^{-1} \quad \Longrightarrow \quad G = \frac{1}{\Delta} \ln \left| \frac{b}{a} \right| \quad , \quad H = \frac{2}{\Delta} \left[1 + \frac{m}{\Delta} \ln \left| \frac{a}{b} \right| \right] \tag{524}$$

$$w_a = \frac{b}{b-a} \ln \left| \frac{b}{a} \right| \; - \; 1 \quad , \quad w_b = \frac{-a}{b-a} \ln \left| \frac{b}{a} \right| \; + \; 1 \tag{525}$$

B.6. Exponential weight function

$$g(x) = e^{-x} \quad \Longrightarrow \quad G = \frac{1}{\Delta} \left(e^{-a} - e^{-b} \right) \quad H = \frac{1}{\Delta} \left(-e^{-a} - e^{-b} + \frac{2}{\Delta} \left(e^{-a} - e^{-b} \right) \right) \tag{526}$$

$$w_a = e^{-a} + \frac{1}{\Delta} \left(e^{-b} - e^{-a} \right), \qquad w_b = -e^{-b} - \frac{1}{\Delta} \left(e^{-b} - e^{-a} \right) \tag{527}$$

TO BE SORTED !!!!!!!!!!!

We describe in the following the integration weights used for the calculation of J, H, K and N.

J: To solve the integral $\int_{-1}^{1} J \, d\mu$ we prepare the integration weights W_0 along each ray (JP) for all valid depth points ($\mathrm{L} = 1 \ldots \mathrm{L}_{\max}$). In this special case (weight function is 1) $W_a = W_b = 1/2$. W_0 is then defined by

JP=1 $\Longrightarrow \mathrm{W}_0(\mathrm{L}) = \frac{Z(\mathrm{L},1)-Z(\mathrm{L},2)}{2R(\mathrm{L})}$

intemediate steps $\Longrightarrow \mathrm{W}_0(\mathrm{L}) = \frac{Z(\mathrm{L,JP}-1)-Z(\mathrm{L,JP}+1)}{2R(\mathrm{L})}$

Last step (non-core) $\Longrightarrow \mathrm{W}_0(\mathrm{L} = \mathrm{L}_{\max}) = \frac{Z(\mathrm{L,JP}-1)}{2R(\mathrm{L})}$
The integrations weights are then $W_a = W_b = 1/2$

H: The integral $\int_{-1}^{1} J \mu \, d\mu$ can be written as $\frac{p}{r^2} dp$. Therefore the weights are equal for all depths. We introduce the abbreviations $A = P_{\mathrm{J}-1}$, $B = P_{\mathrm{J}}$ and $C = P_{\mathrm{J}+1}$ Each integration weight has contributions from the left and the right intervall W_b of the left interval is

$$W_b \;=\; \frac{\Delta}{2r^2} \left(m + \frac{\Delta}{6} \right) \;=\; \frac{B-A}{2r^2} \left(\frac{B+A}{2} + \frac{B-A}{6} \right) \;=\; \frac{1}{6r^2} (2B^2 - AB - A^2)$$

while W_a of the right interval is

$$W_a \;=\; \frac{\Delta}{2r^2} \left(m - \frac{\Delta}{6} \right) \;=\; \frac{C-B}{2r^2} \left(\frac{C+B}{2} - \frac{C-B}{6} \right) \;=\; \frac{1}{6r^2} (C^2 + BC - 2B^2)$$

both terms together yield

$$W \;=\; \frac{1}{6r^2} (C^2 + BC - AB - A^2) \;=\; \frac{1}{r^2} (A + B + C)(C - A)$$

At the first (last) point only W_a (W_b) contributes

$$W \;=\; W_a \;=\; (C - B)(C + 2B), \quad W \;=\; W_b \;=\; (B - A)(B + 2A) \quad \text{, respectively}$$

Special attention is essential to the end of an integration if non-core rays are considered. W_a of the right interval has then to be calculated for the half interval [JMAX − 1, JMAX], i.e $C \;=\; 1/2(P_{\mathrm{J}} + P_{\mathrm{J}+1})$. This integration weight is written into WP1Last. The complete sum to integrate an interstice shell is then

$$H \;=\; \left(\sum_{\mathrm{P}=1}^{\mathrm{JMAX}-1} v_j \,\cdot\, \mathrm{WP}_j \right) + \, v_{j=\mathrm{JMAX}} \,\cdot\, \mathrm{WP1LAST}_{j=\mathrm{JMAX}}$$

K: The same as for J but with different weights. $\int_{-1}^{1} \mu^2\, d\mu = \frac{1}{r^3} \int z^2\, dz$.

$$\frac{\Delta}{2}\left(m^2 + \frac{\Delta^2}{12}\right) = \frac{1}{6}(B^3 - A^3) \quad \text{and} \quad \frac{\Delta}{2} m \frac{\Delta}{3} = \frac{1}{12}(B^3 - A^2B - AB^2 + A^3)$$

$$\begin{aligned} \rightsquigarrow\ r^3 W_b &= \frac{1}{6}(B^3 - A^3) + \frac{1}{12}(B^3 - A^2B - AB^2 + A^3) \quad \text{and} \\ r^3 W_a &= \frac{1}{6}(C^3 - B^3) - \frac{1}{12}(C^3 - B^2C - BC^2 + B^3) \\ W_a + W_b &= \frac{1}{12r^3}\left[B(C^2 - A^2) + C(C^2 + B^2) - A(A^2 + B^2)\right] \end{aligned}$$

N: The integral $\int_{-1}^{1} \mu^3\, d\mu$ can be written as $\int \left(\frac{p}{r^2} - \frac{p^3}{r^4}\right) dp$. The weights are principally in the same manner as the weights for H. For the second term ($\mu^3\, d\mu$) the weights are the following

$$W_a = \frac{\Delta}{2}\left(m^3 + \frac{1}{4} m\Delta^2 - \frac{1}{2} m^2 \Delta - \frac{\Delta^3}{40}\right) \quad \text{and}$$

$$W_b = \frac{\Delta}{2}\left(m^3 + \frac{1}{4} m\Delta^2 + \frac{1}{2} m^2 \Delta + \frac{\Delta^3}{40}\right)$$

$$\begin{aligned} W_a &= \frac{C - B}{16}\left((C + B) + (C + B)(C - B)^2 - (C + B)^2(C - B) - \frac{C - B}{5}\right) \quad \text{and} \\ W_b &= \frac{B - A}{16}\left((B + A) + (B + A)(B - A)^2 + (B + A)^2(B - A) + \frac{B - A}{5}\right) \end{aligned}$$

Together with the first term ($\mu\, d\mu$) we derive

$$W_a = (C - B)(C + 2B) + \frac{C - B}{16}\left((C + B) + (C + B)(C - B)^2 - (C + B)^2(C - B) - \frac{C - B}{5}\right)$$

and

$$W_b = (B - A)(B + 2A) + \frac{B - A}{16}\left((B + A) + (B + A)(B - A)^2 + (B + A)^2(B - A) + \frac{B - A}{5}\right)$$

Because the calculation of the weights is not time-critical, we do not simplyfy these formulae in order to avoid errors.

C. Interpolation with cubic splines

D. Characteristics of the Moment Equations

In this Appendix we want to derive the consitions for the coefficients in the moment equations which must be fulfilled to be of *hyperbolic type*. This will yield constraints for the choice of the Eddimix-Parameter ϵ.

This notation in this Appendix, which is copied from the work of Christian Friedl, is slightly different from the one in the main Sections. E.g., we omit the tilde, i.e. $J = \tilde{J} = r^2 J, H = \tilde{H} = r^2 H, K = \tilde{K} = r^2 K, N = \tilde{N} = r^2 N$. Therefore we first briefly repeat the equations.

$$\frac{\partial H(r,x)}{-\partial r} + \left(v'(r) - \frac{v(r)}{r}\right)\frac{\partial K(r,x)}{\partial x} + \frac{v(r)}{r}\frac{\partial J(r,x)}{\partial x}$$
$$= (J(r,x) - S(r,x))\,\kappa(r,x),$$

$$\frac{\partial(qK)(r,x)}{-q(r,x)\partial r} + \left(v'(r) - \frac{v(r)}{r}\right)\frac{\partial N(r,x)}{\partial x} + \frac{v(r)}{r}\frac{\partial H(r,x)}{\partial x}$$
$$= \kappa(r,x)H(r,x),$$

with
$J(r,x)$: 0. moment
$H(r,x)$: 1. moment
$K(r,x)$: 2. moment
$N(r,x)$: 3. moment
$S(r,x)$: sourcefunction
$v(r)$: velocity > 0
$v'(r)$:velocity gradient > 0
$q(r,x)$: sphericity factor > 0
$f(r,x)$: $\frac{K}{J}$ Eddingtonfactor $0 < f < 1$
$g(r,x)$: $\frac{N}{H+\epsilon J}$ another Eddington factor
$\epsilon(r)$: Eddimix parameter.

Respectively in a form which only contains the moments J and H:

$$\frac{\partial H(r,x)}{-\partial r} + \left(v'(r) - \frac{v(r)}{r}\right)\frac{\partial(fJ)(r,x)}{\partial x} + \frac{v(r)}{r}\frac{\partial J(r,x)}{\partial x}$$
$$= (J(r,x) - S(r,x))\,\kappa(r,x),$$

$$\frac{\partial(qfJ)(r,x)}{-q(r,x)\partial r} + \left(v'(r) - \frac{v(r)}{r}\right)\frac{\partial[g(H+\epsilon J)](r,x)}{\partial x} + \frac{v(r)}{r}\frac{\partial H(r,x)}{\partial x}$$
$$= \kappa(r,x)H(r,x),$$

The first equation is called the 0.moment equation, and the second one is called the 1.moment equation. Building the linear combination (1) + λ(0) gives

$$\begin{aligned}\frac{\partial(qfJ)}{q\partial r} + \left(\frac{v}{r} - \frac{\mathrm{d}v}{\mathrm{d}r}\right)\frac{\partial}{\partial x}\Big[g(H+\epsilon J)\Big] - \frac{v}{r}\frac{\partial H}{\partial x} \\ +\lambda\left\{\frac{\partial H}{\partial r} + \left(\frac{v}{r} - \frac{\mathrm{d}v}{\mathrm{d}r}\right)\frac{\partial(fJ)}{\partial x} - \frac{v}{r}\frac{\partial J}{\partial x}\right\} \\ = -\kappa H + \lambda\kappa(S-J).\end{aligned}$$

This can be written as

$$\begin{aligned}\frac{\partial(qf)}{q\partial r}J + f\frac{\partial J}{\partial r} \\ +\left(\frac{v}{r} - \frac{\mathrm{d}v}{\mathrm{d}r}\right)\times\left[\frac{\partial g}{\partial x}H + g\frac{\partial H}{\partial x} + \epsilon J\frac{\partial g}{\partial x} + \epsilon g\frac{\partial J}{\partial x}\right] - \frac{v}{r}\frac{\partial H}{\partial x} \\ +\lambda\left\{\frac{\partial H}{\partial r} + \left(\frac{v}{r} - \frac{\mathrm{d}v}{\mathrm{d}r}\right)\left(\frac{\partial f}{\partial x}J + f\frac{\partial J}{\partial x}\right) - \frac{v}{r}\frac{\partial J}{\partial x}\right\} \\ = -\kappa H + \lambda\kappa(S-J).\end{aligned} \tag{528}$$

Note, that

$$\kappa(J-S) = \kappa J - \eta = (\kappa_{\mathrm{noth}} + \kappa_{\mathrm{Th}})J - \eta_{\mathrm{Th}} - \eta_{\mathrm{noth}} = \kappa_{\mathrm{noth}}J - \eta_{\mathrm{noth}}.$$

Sorting by derivatives we obtain

$$\begin{aligned}fJ_r + \lambda H_r + \left[\left(\frac{v}{r} - \frac{\mathrm{d}v}{\mathrm{d}r}\right)g - \frac{v}{r}\right]H_x \\ +\left[\left(\frac{v}{r} - \frac{\mathrm{d}v}{\mathrm{d}r}\right)\epsilon g + \left(\frac{v}{r} - \frac{\mathrm{d}v}{\mathrm{d}r}\right)\lambda f - \lambda\frac{v}{r}\right]J_x \\ +\left[\frac{\partial(qf)}{q\partial r} + \lambda\left(\frac{v}{r} - \frac{\mathrm{d}v}{\mathrm{d}r}\right)\frac{\partial f}{\partial x} + \lambda\kappa_{\mathrm{noth}} + \left(\frac{v}{r} - \frac{\mathrm{d}v}{\mathrm{d}r}\right)\epsilon\frac{\partial g}{\partial x}\right]J \\ +\left[\left(\frac{v}{r} - \frac{\mathrm{d}v}{\mathrm{d}r}\right)\frac{\partial g}{\partial x} + \kappa\right]H - \lambda\eta_{\mathrm{noth}} = 0,\end{aligned} \tag{529}$$

and with the abbreviations
$V := \frac{v}{r} > 0$
$G := \frac{\mathrm{d}v}{\mathrm{d}r} - \frac{v}{r}$
$F := fG + V > 0$
$D := gG + V > 0$

(F is greater than zero because $0 < f < 1$, D is greater than zero because of the "Eddireset condition"), we achieve the following result

$$\begin{aligned}fJ_r + \lambda H_r - DH_x - [G\epsilon g + \lambda F]\,J_x \\ +\left[\frac{\partial(qf)}{q\partial r} + \lambda\left(\kappa_{\mathrm{noth}} - G\frac{\partial f}{\partial x}\right) - G\epsilon\frac{\partial g}{\partial x}\right]J \\ +\left[\kappa - G\frac{\partial g}{\partial x}\right]H - \lambda\eta_{\mathrm{noth}} = 0.\end{aligned} \tag{530}$$

It's possible to simplify the term with the sphericity factor q:

$$\frac{\partial(qf)}{q\partial r} = \frac{\partial f}{\partial r} + \frac{f}{q}\frac{\partial q}{\partial r}, \tag{531}$$

the sphericity factor is implicitly defined as follows

$$\frac{\partial \ln(\mathrm{r^2 q})}{\partial r} = \frac{3f-1}{fr}.$$

The lhs gives

$$\frac{1}{r^2 q}\left[2rq + r^2\frac{\partial q}{\partial r}\right] = \frac{2}{r} + \frac{1}{q}\frac{\partial q}{\partial r},$$

this leads to the relation

$$\frac{f}{q}\frac{\partial q}{\partial r} = \frac{f-1}{r}.$$

So we obtain the final representation

$$\begin{aligned} & fJ_r + \lambda H_r - DH_x - \left[G\epsilon g + \lambda F\right] J_x \\ +& \left[\frac{\partial f}{\partial r} + \frac{f-1}{r} + \lambda\left(\kappa_{\mathrm{noth}} - G\frac{\partial f}{\partial x}\right) - G\epsilon\frac{\partial g}{\partial x}\right] J \\ +& \left[\kappa - G\frac{\partial g}{\partial x}\right] H - \lambda\eta_{\mathrm{noth}} = 0. \end{aligned} \tag{532}$$

To get the value of of the scaling factor λ and something more, a short insertion:
Consider a function u, which depends of 2 variables r and x, which are parametrized by a parameter σ, i. e.

$$u = u(r(\sigma), x(\sigma)). \tag{533}$$

The derivative gives

$$\frac{\mathrm{d}u}{\mathrm{d}\sigma} = u_r\frac{\mathrm{d}r}{\mathrm{d}\sigma} + u_x\frac{\mathrm{d}x}{\mathrm{d}\sigma}. \tag{534}$$

So, every linearcombination $au_r + bu_x$ can be interpreted as differentating u to σ along the curve $(r(\sigma), x(\sigma))$. The slope of this curve in the $x - r$ plane is given by

$$\frac{\mathrm{d}r}{\mathrm{d}x} = \frac{\frac{\mathrm{d}r}{\mathrm{d}\sigma}}{\frac{\mathrm{d}x}{\mathrm{d}\sigma}} = \frac{\dot{r}}{\dot{x}} = \frac{a}{b}. \tag{535}$$

Characteristic curves of 2 functions $u_1(r, x)$ and $u_2(r, x)$ have the property, that the slope $\frac{\mathrm{d}r}{\mathrm{d}x}$ is the **same** (this slope gives a so called characteristic direction).
The following equation delivers the slope of the characteristics

$$\frac{\mathrm{d}r}{\mathrm{d}x} = \frac{a_1}{b_1} = \frac{a_2}{b_2}. \tag{536}$$

In our case, we set $u_1 = J$, $u_2 = H$ and from (532) we can read off the slope of our characteristic curves

$$\frac{\mathrm{d}r}{\mathrm{d}x} = \frac{f}{-G\epsilon g - \lambda F} = \frac{\lambda}{-D}. \tag{537}$$

This leads to an quadratic equation for the scaling factor λ

$$\lambda^2 F + \lambda G\epsilon g - Df = 0, \tag{538}$$

with the solutions

$$\lambda_+ = \frac{-G\epsilon g + \sqrt{X}}{2F} > 0, \tag{539}$$

$$\lambda_- = \frac{-G\epsilon g - \sqrt{X}}{2F} < 0, \tag{540}$$

and

$$X = (G\epsilon g)^2 + 4DFf \ . \tag{541}$$

The system of differential equations is called of *hyperbolic type*, if there real characteristics exist.

Obviously it is sufficient for the discriminant X to be real if D, F and f are positive. Since $J > 0$ and $H < J$, this is sure for $f = H/J$ and $F = fV' + (1 - f)V$.

For $D = gG + V$ being always positiv, however, we must demand a large enough ϵ. This is exactly the same condition that we have already derived in order to avoid singular coefficients in the moment equations, see Eq. (204).

E. Conversion, constants, formulae

E.1. Constants and conversions

$$1\,\mathrm{cm} = 10^8\,\mathrm{\AA} \tag{542}$$

$$1\mu = 10^5\,\mathrm{\AA} \tag{543}$$

$$1\,\mathrm{\AA} = \frac{1.239842\times10^4}{1\,\mathrm{eV}} \tag{544}$$

$$1\,\mathrm{pc} = 3.085677\times10^{16}\,\mathrm{m} = 3.09\times10^{18}\,\mathrm{cm} = \exp(18.4894)\,\mathrm{cm} \tag{545}$$

$$1\,\mathrm{Ryd} = 13.6\,\mathrm{eV}\cdot 1.6022\times10^{-19}\,\mathrm{C}/6.626\times10^{-34}\,\mathrm{Js}/2.9979\times10^{10}\,\mathrm{cm\,s^{-1}} \quad [\,\mathrm{cm^{-1}}] \tag{546}$$

$$= 1.097375\times10^5\,\mathrm{cm^{-1}} \tag{547}$$

$$1\,\mathrm{W\,m^{-2}} = 10^3\,\mathrm{erg\,s^{-1}\,cm^{-2}} \quad | \quad 1\,\mathrm{erg\,s^{-1}\,cm^{-2}\,\mathrm{\AA}} = 10^{-2}\,\mathrm{W\,m^{-2}\,nm^{-1}} \tag{548}$$

$$1\,\mathrm{W\,m^{-2}}\,\mu^{-1} = 10^{-2}\,\mathrm{erg\,s^{-1}\,cm^{-2}\,\mathrm{\AA}^{-1}} \tag{549}$$

$$1\,\mathrm{Jy} = 10^{-26}\ \mathrm{W\ Hz^{-1}\ m^{-2}} = 10^{-23}\ \mathrm{erg\ s^{-1}\ cm^{-2}\ Hz^{-1}} = 10^{-23}\ \frac{\lambda^2}{c}\ \mathrm{erg\ s^{-1}\ cm^{-2}\ \mathrm{\AA}^{-1}} \tag{550}$$

$$E_{\mathrm{B-V}} = 0.826\,E_{\mathrm{b-v}} = 0.77\,c(H\beta) \quad (R = 3.1) \tag{551}$$

$$L_\odot = 3.846\times10^{26}\,\mathrm{W} = 3.846\times10^{33}\,\mathrm{erg\,s^{-1}} = 33.585\ \log(\,\mathrm{erg\,s^{-1}}) \tag{552}$$

$$R_\odot = 69.57\times10^9\,\mathrm{cm} = 10.842\ \log(\,\mathrm{cm}) \tag{553}$$

$$M_\odot = 1.989\times10^{33}\ \mathrm{g} = 33.299\ \log(\,\mathrm{g}) \tag{554}$$

$$1\,M_\odot\,\mathrm{a^{-1}} = 6.303\times10^{25}\,\mathrm{g\,s^{-1}} = 25.80\ \log(\,\mathrm{g\,s^{-1}}) \tag{555}$$

$$h = 6.6237\times10^{-27}\,\mathrm{erg\,s} = -26.179\ \log(\,\mathrm{erg\,s}) \tag{556}$$

$$k_\mathrm{B} = 1.3803\times10^{-16}\,\mathrm{erg\,grad} = 8.6173303\times10^{-5}\,\mathrm{eV/\,K} = -15.860\ \log(\,\mathrm{erg\,grad}) \tag{557}$$

$$hc = 1.98648\times10^{-8}\,\mathrm{erg\,\mathrm{\AA}} \tag{558}$$

$$\frac{hc}{k_\mathrm{B}} = 1.4385\,\mathrm{cm\,grad} = 0.1579\ \log(\,\mathrm{cm\,grad}) \tag{559}$$

$$\sigma_\mathrm{SB} = 5.671\times10^{-5}\,\mathrm{erg\,cm^{-2}\,s^{-1}\,grad^{-4}} = -4.2463\ \log(\,\mathrm{erg\,cm^{-2}\,s^{-1}\,grad^{-4}}) \tag{560}$$

$$\sigma_0 = \frac{\pi\,e_0^2}{m_\mathrm{e}c} = 0.026537\,\mathrm{cm^2\,Hz} \tag{561}$$

$$G = 6.67384\times10^{-8}\,\mathrm{cm^3\,g^{-1}\,s^{-2}} \tag{562}$$

$$\tag{563}$$

E.2. Formulae

$$\mathrm{DM} = m - M = 5\log_{10} d[\,\mathrm{pc}] - 5 \quad \Leftrightarrow \quad d[\mathrm{pc}] = 10^{\mathrm{DM}/5+1} \tag{564}$$

$$R_\mathrm{t} = R_*\left[\frac{\upsilon_\infty}{2500\,\mathrm{km}s^{-1}}\bigg/\frac{\dot{M}\sqrt{D}}{10^{-4}\,\mathrm{M_\odot\,a^{-1}}}\right]^{2/3} \tag{565}$$

$$\upsilon_\mathrm{esc} = 618\,\mathrm{km\,s^{-1}}\sqrt{\frac{(M/\mathrm{M_\odot})}{(R/\mathrm{R_\odot})}} \tag{566}$$

$$= 618\,\mathrm{km\,s^{-1}}\sqrt[4]{M/\mathrm{M_\odot}}\cdot 10^{\frac{1}{4}(\log g - 4.4371)} \tag{567}$$

$$= 618\,\mathrm{km\,s^{-1}}\sqrt{10^{\log g - 4.4371}\,R/R_\odot} \tag{568}$$

$$\log g = 4.4382\,\mathrm{cm\,s^{-2}} + \log\left(\frac{M/M_\odot}{(R/R_\odot)^2}\right) \tag{569}$$

$$\frac{M}{M_\odot} = 10^{\log g - 4.4371 + 2\log(R/R_\odot)} \tag{570}$$

$$\tag{571}$$

$$\frac{R}{R_\odot} = 10^{\frac{1}{2}(\log(L/L_\odot) - 4\log T_{\rm eff} + 15.05)} \tag{572}$$

$$= 10^{\frac{1}{2}(\log(M/M_\odot) - \log g + 4.4371)} \tag{573}$$

$$T_{\rm eff} = \frac{\sqrt[4]{L/L_\odot}}{\sqrt{R/R_\odot}}\, 10^{3.7625} = \frac{10^{\frac{1}{4}\log L/L_\odot}}{\sqrt{R/R_\odot}}\, 10^{3.7625} \tag{574}$$

$$v_{\rm th} = \sqrt{\frac{2k_{\rm B}T}{m_{\rm A}}} = 0.1285\ \mathrm{km\ s^{-1}} \sqrt{\frac{T/\mathrm{K}}{A}} = \sqrt{2}\,a \quad \left(a = \text{isothermal sound speed} = \sqrt{\frac{5}{3}\frac{p}{\varrho}}\right) \tag{575}$$

$$\text{e.g. O star: } T = 50\,\mathrm{kK}, A = 1 \text{ (hydrogen)} \Rightarrow v_{\rm th} = 29\ \mathrm{km\ s^{-1}} \tag{576}$$

Re-scaling grid models for a different luminosity L_*, with $\Delta\log L_* = \log L_* - \log L'_*$:

$$R_* \sim L^{1/2} \quad \Rightarrow \quad R_* = R_*{}' \left(10^{\Delta\log L_*}\right)^{1/2} \tag{577}$$

$$\dot{M} \sim L^{3/4} \quad \Rightarrow \quad \log\dot{M} = \log\dot{M}' + \frac{3}{4}\Delta\log L_* \tag{578}$$

If you want to convert our T_* to the often used $T_{2/3}$, i.e. the temperature at the radius $R_{2/3}$, where $\tau_{\rm Ross} = 2/3$:

$$T_{2/3} = T_* \sqrt{\frac{R_*}{R_{2/3}}} \tag{579}$$

F. Contour plots for model grids

This is a more or less complete description how to generate a contour plot over a given grid of models, where T_* is on the x-axis, R_t on the y-axis, and the iso-lines (z-axis) indicate one of the following quantities:

- the equivalent width of a spectral line, e.g. He I 5875.7, for every model of the grid as a function of (T_*, R_t),
- or the ratio of the equivalent widths of two different spectral lines, e.g. EW(He II 4686) / EW(He I 5875.7) as a possible indicator for T_*, for every model of the grid as a function of (T_*, R_t),
- or the peak value of a spectral line in the continuums normalized form, e.g. the peak value of He I 5875.7, for every model of the grid as a function of (T_*, R_t),
- or the ratio of the peak values for two different spectral lines, e.g. EW(He II 4686) / EW(He I 5875.7) as a possible indicator for T_*, for every model of the grd as a function of (T_*, R_t).

F.0.1. How to create contur plots

1. *gridformal:* The `formal`-output files:
   - Create a directory, e.g. `grid.dir` that contains subdirectories (or symbolic links to directories) with the file `MODEL`, so that all `MODEL` files can be found by
     `ls .../grid.dir/*/MODEL`. The names of the subdirectories (or symbolic links) should refer to our usual grid spacing of the parameters, e.g. "`14-17`".
   - Edit the job `gridformal` in your `~/work/wrjobs`, e.g. `gridformal1`: set the path to the grid directory (see above) and the path to the output directory, i.e. the directory where the files `formal.out` and `formal.plot` for every model of the grid are written to,
     e.g. `formal_HeI_5876.dir`. Note, that the names of the `plot` and `.out` files are the names of the original grid subdirectories, e.g. `17-14.plot`.
   - The necessary files for a the normal `formal` job i.e. the correct `FEDAT_FORMAL`, `DATOM`, and `FORMAL_CARDS` with the respective line(s) or multiplett only, must exist in the `wrdata` directory associated with the `gridformal` job, for instance `wrdata1` for `gridformal1`.
     For larger grids (i.e. with many models), it may be time-saving to set the option `NOREDIS`[7] *in* `FORMAL_CARDS`. *Furthermore, the equivalent width of a line, measured in the observed spectrum is usually measured without the scattering wings.*
   - The command
     `sub gridformal1`
     starts a sequence of `formal` jobs, one for each model of the grid. The status of the `gridformal` job can be monitored by dint of the command
     `stat tlog gridformal1.`
     The `MODEL` files are not copied to the `wrdata(1)` directorz, but instead, a symbolic link is created. When the `gridformal` job has finished, this link targets to the nonexistent file "`finished`".

   After a successful run of `gridformal`, the output directory contains the `formal.plot` and `formal.out` files. Only the latter are needed for the next step.
2. *my_isoex* Collecting the data

[7] no photon redistribution due to electron scattering

- The file iso.part1 is needed in the output directory. For instance, in the case of He II 4686 it may have the following content:

```
PEAK
DIVISORDIRECTORY = ~/models.dir/formal_HeI_5876
** Box for isoplots:
RRBOX
*      scale   min   max  tick  schrift  wert
xkas   0.      4.45  5.3  0.10   0.1      0.0
ykas   0.    -0.3   1.9   0.10   0.5      0.0
xlet=\CENTER\log (&IT&N\* / kK)
ylet=\CENTER\log (&IR&N&Tt&M / &IR&N\S )
logx
logy
*PPCM 20.0
PPCM 40.0
grid 8,0.2
end
---------------------------------------------------
*NAME:
line
TYPE : COMM
Kasten=RRBOX
*
HEADER nodiel-wngrid: He II 4686
WRPLOT 5  0.1
MONO
SLOPE -0.6
STEP= 30.00
*STEP= 1.00
* first number = number of the following parameters
* but is not longer evaluated
*--------------------------------------------
*level 0  -1 -2 -4 -10 -20 -40 -80
format=f5.0
```

- Edit the macro for the plotfile header and the contour increment in iso.part1 if necessary.
- Change to the output directory and execute my_isoex. This program is available as a Fortran binary for DEC Alpha (~wrh/libraries.dir/my_isoex.exe) and Linux 64Bit (~htodt/bin/my_isoex.exe). However, more flexible and transparent (easy to edit) is the alternative bash-script my_isoex.bash (e.g. from ~htodt/proc.dir/my_isoex.bash), with following abilities:
  - by setting the keyword PEAK (at the beginning of any line in iso.part1, not case sensitive) or as a command line argument (at arbitrary position and not case sensitive), the peak values instead of the equivalent widths are use;
  - by providing an additional output directory from another spectral line, either within iso.part1 with leading keyword divisordirectory:
divisordirectory=~/models.dir/formal_HeI_5876
or simply as another command line argument, the ratio of the equivalent widths of the current directory divided by the *divisordirectory* is used for the contour plot. Analogously, the peak values are used, if keyword PEAK is set.

  my_isoex or my_isoex.bash creates the file isoex.iso.
- In the last step, the Fortran program isopro.exe processes the file isoex.iso and produces the WRplot file isopro.plot with the intended contour plot is created. The program isopro.exe works as follows:
  - firstly it asks for the input file, if left blank, the file isoexe.iso in the current directory is taken as input file - press ENTER. You have to press ENTER once more and than terminate the program by “CTRL + D”.

- the following keywords are interpreted by `isopro.exe`:

STEP step widths, determines number of iso lines between minimum and maximum (default is 10)

level alternatively, prescribe the values for the isolines to plot

FORMAT format of the isoline labels, that are the values for EW or peak height (FORTRAN syntax), e.g. `FORMAT=f5.1`
default: `f5.2` (set in subroutine `INCOM` in variable `ISOFOR`)

xkas description of the x-axis scaling (WRplot syntax)

ykas description of the x-axis scaling (WRplot syntax)

xlet label for the x-axis (WRplot syntax)

Part V. References

References

Allen, C. W. 1973, Astrophysical quantities, ed. Allen, C. W.

Baum, E., Hamann, W.-R., Koesterke, L., & Wessolowski, U. 1992, A&A, 266, 402

Benson, R. S. & Kulander, J. L. 1972, Sol. Phys., 27, 305

Berger, J. M. 1956, ApJ, 124, 550

Berrington, K. A., Fon, W. C., & Kingston, A. E. 1982, MNRAS, 200, 347

Chambe, G. & Lantos, P. 1971, Sol. Phys., 17, 97

Cowan, R. D. 1981, The theory of atomic structure and spectra, ed. R. D. Cowan

Cowley, C. R. 1971, The Observatory, 91, 139

Daltabuit, E. & Cox, D. P. 1972, ApJ, 177, 855

Gräfener, G. & Hamann, W. R. 2005, A&A, 432, 633

Gräfener, G., Koesterke, L., & Hamann, W. 2002, A&A, 387, 244

Gräfener, G. & Vink, J. S. 2013, A&A, 560, A6

Gräfener, G., Vink, J. S., de Koter, A., & Langer, N. 2011, A&A, 535, A56

Griem, H. R. 1960, ApJ, 132, 883

Griem, H. R. 1967, ApJ, 147, 1092

Griem, H. R., Baranger, M., Kolb, A. C., & Oertel, G. 1962, Physical Review, 125, 177

Hamann, W. R. 1981, A&A, 93, 353

Hamann, W.-R. 1987, Line Formation in Expanding Atmospheres: Multi-Level Calculations using Approximate Lambda Operators, ed. W. Kalkofen, 35–+

Hamann, W.-R. & Gräfener, G. 2003, A&A, 410, 993

Hillier, D. J. 1987, ApJS, 63, 965

Hillier, D. J. & Miller, D. L. 1998, ApJ, 496, 407

Hubeny, I., Hummer, D. G., & Lanz, T. 1994, A&A, 282, 151

Hummer, D. G. 1963, MNRAS, 125, 461

Hummer, D. G. & Mihalas, D. 1967, ApJ, 150, L57

Hummer, D. G. & Seaton, M. J. 1963, MNRAS, 125, 437

Jefferies, J. T. 1968, Spectral line formation, ed. J. T. Jefferies

Karzas, W. J. & Latter, R. 1961, ApJS, 6, 167

Koester, D., Vauclair, G., Dolez, N., et al. 1985, A&A, 149, 423

Kramers, H. 1923

Kubát, J. 2001, A&A, 366, 210

Kubát, J., Puls, J., & Pauldrach, A. W. A. 1999, A&A, 341, 587

Langer, N. 1989, A&A, 210, 93

Lemke, M. 1997, A&AS, 122, 285

Leuenhagen, U. & Hamann, W.-R. 1994, A&A, 283, 567

Lucy, L. B. 1964, SAO Special Report, 167, 93

Martins, F., Hillier, D. J., Bouret, J. C., et al. 2009, A&A, 495, 257

Mendoza, C. 1992, Atomic Data from the Opacity Project, ed. P. L. Smith & W. L. Wiese, Vol. 407, 85

Mihalas, D. 1967, ApJ, 149, 169

Mihalas, D. 1978, Stellar atmospheres

Mihalas, D. & Stone, M. E. 1968, ApJ, 151, 293

Morton, D. C. 1991, ApJS, 77, 119

Najarro, F., Figer, D. F., Hillier, D. J., Geballe, T. R., & Kudritzki, R. P. 2009, ApJ, 691, 1816

Oskinova, L. M., Hamann, W., & Feldmeier, A. 2007, A&A, 476, 1331

Schoening, T. & Butler, K. 1989, A&AS, 78, 51

Schoning, T. & Butler, K. 1989, A&AS, 79, 153

Seaton, M. J. 1958, Reviews of Modern Physics, 30, 979

Seaton, M. J. 1960, Reports on Progress in Physics, 23, 313

Shenar, T., Hamann, W. R., & Todt, H. 2014, A&A, 562, A118

Unsoeld, A. 1968, Physik der Sternatmosphaeren mit besonderer Beruecksichtigung der Sonne

Unsöld, A. 1951, Naturwissenschaften, 38, 525

Unsöld, A. 1955, Physik der Sternatmosphären, mit besonderer Berücksichtigung der Sonne (Berlin, Springer, 1955. 2. Aufl.)

Verner, D. A. & Yakovlev, D. G. 1995, A&AS, 109, 125

Verner, D. A., Yakovlev, D. G., Band, I. M., & Trzhaskovskaya, M. B. 1993, Atomic Data and Nuclear Data Tables, 55, 233

Vidal, C. R., Cooper, J., & Smith, E. W. 1973, ApJS, 25, 37

Wessolowski, U. 1991, Dissertation: Der WN6-Prototyp HD 192163

Wiese, W. L., Fuhr, J. R., & Deters, T. M. 1996, Atomic transition probabilities of Carbon, Nitrogen, and Oxygen. A critical data compilation (NSRDS-NBS 4, Washington, D.C.: US Department of Commerce, National Buereau of Standards, 1966)

Wiese, W. L., Smith, M. W., & Glennon, B. M. 1966, Atomic transition probabilities. Vol.: Hydrogen through Neon. A critical data compilation (NSRDS-NBS 4, Washington, D.C.: US Department of Commerce, National Buereau of Standards, 1966)