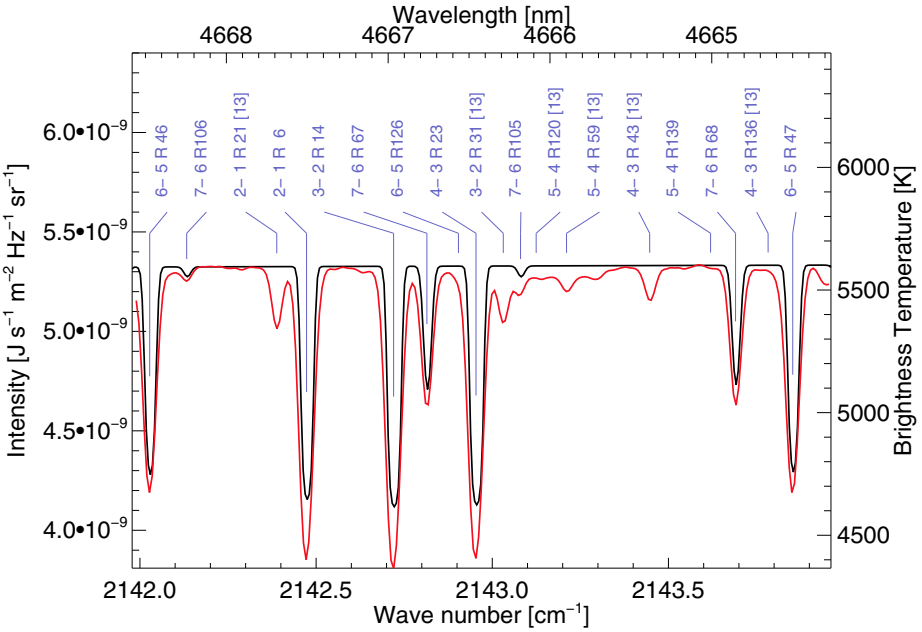
User's manual for RH radiative transfer code



Han Uitenbroek
huitenbroek@nso.edu

June 15, 2010

Contents

1 Introduction

This manual describes version 2 of the numerical radiative transfer code based on the MALI (Multi-level Approximate Lambda Iteration) formalism of Rybicki & Hummer (1991, 1992). The code solves the combined equations of statistical equilibrium and radiative transfer for multi-level atoms and molecules in a given plasma under general Non-LTE conditions. There are currently four different versions available for different geometries (see Table 1). The Rybicki & Hummer formalism allows radiative (bound-free as well as bound-bound) transitions to overlap in wavelength. This is implemented for all four versions of the code. In addition, each version includes the effects of partial frequency redistribution (PRD) in bound-bound transitions when necessary. Both angle-averaged redistribution in plasmas without macroscopic flows is implemented, as well as the more general Angle-dependent redistribution formalism. Convergence of the Approximate Lambda Iteration (ALI) scheme can be accelerated with Ng's (1974) optimization method. The formal solution of the transfer equation with a given source function is implemented with the Feautrier difference method or the short characteristics method (which is an integral method), where appropriate. Non-LTE is solved in molecules for transitions between vibrational levels as described by Uitenbroek (2000). Multiple atoms and molecules can be solved in Non-LTE at the same time.

name		**geometry**
rhf1d	—	1-D plane-parallel geometry
rhsc2d	—	2-D Cartesian geometry
rhsc3d	—	3-D Cartesian geometry
rhsphere	—	1-D spherically symmetric geometry

Table 1: Versions for different geometries.

All three Cartesian versions (rhf1d, rhsc2d, rhsc3d) can calculate the full Stokes vector resulting from the Zeeman effect in atoms and molecules, as well as the linear polarization resulting from scattering by electrons (Thomson) and atoms (Rayleigh).

In addition to opacities and emissivities from the transitions in the "active" atoms and molecules, the code accounts for "background" radiation sinks and sources due to other atoms, molecules, and all relevant continuum processes, including (but not exclusive) H^- bound-bound and bound-free processes, scattering off free electrons (Thomson) and Rayleigh scattering off neutral hydrogen helium, and H_2, hydrogen free-free processes, and bound-free processes in OH and CH molecules. Chemical equilibrium is calculated for an arbitrary mixture of molecules.

The code is written in the C language and currently compiles and runs, without modification, on the architecture – operating system combinations listed in Table 2. For the most part the different geometrical versions use the same basic routines, e.g., to calculate opacities, solve the statistical and chemical equilibrium equations, read in the atomic data. This is possible because most operations, although acting on spatially varying quantities, are geometry independent. The most notable exceptions are atmosphere initialization, the formal solution of the transfer equation, and writng of output data. This sharing of code by the

architecture		operating system
SUN sparc	—	SunOS 5.x
x86_64 compatible	—	SunOS 5.x
x86 compatible	—	Linux 2.x
x86_64 compatible	—	Linux 2.x
i386 compatible	—	Darwin

Table 2: Currently supported computer systems.

different versions obviously simplifies testing procedures and maintenance of the code.

A number of widget driven IDL routines is available to read output files and analyze the results. Most of these routines provide graphical output that requires only a few mouse-clicks to reach. All binary output files are written in the platform-independent external data representation (XDR) format to simplify transport of output data to other computer platforms without having to worry about floating point representations and the order of bytes.

Han Uitenbroek
National Solar Observatory/Sacramento Peak
P.O. Box 62, Sunspot, NM 88349, USA
Tel.: 1 (575) 434-7032
E-mail: huitenbroek@nso.edu

2 Installation

2.1 Extracting the files

The transfer code and supplementary files are distributed in 5 gzipped tar files:

archive	content
`rhv2src.tgz`	Source files, include files and make files
`rhv2inputs.tgz`	Sample input files
`rhv2idl.tgz`	IDL analysis routines
`rhatoms.tgz`	Sample atomic and molecular data input files
`rhatmos.tgz`	Sample atmospheric data input files

Table 3: Distribution files

To install the program and accessory files first make a directory `src/rh` and copy the distribution files there. Each of these distribution files can then be uncompressed and extracted with:

```
% tar xzvf rhv2src.tgz
```

etc. for each of the other files. Note that on Solaris the tar utility does not have the z option for uncompression. In this case use:

```
% gzip -d -c rhv2src.tgz | tar xvf -
```

After extraction the source and input files needed to build and run the transfer code are contained in the following directory structure:

```
rh/
rh/Atmos
rh/Atoms
rh/Molecules
rh/makefiles
rh/rhf1d
rh/rhf1d/run
rh/rhsc2d
rh/rhsc2d/run
rh/rhsc3d
rh/rhsc3d/run
rh/rhsphere
rh/rhsphere/run
rh/tools
```

2.2 Building the executables

The main directory `rh` contains basic geometry independent routines that are shared by the different geometric versions. The `Makefile` in this directory builds two libraries, `librh.a` which holds general routines and routines for the background opacity package, and, if you have a FORTRAN 90 compiler available, `librh_f90.a` which holds a few routines written in F90 for efficiency (in particular for complex number arithmetic). The central makefile `Makefile` includes different makefile additions depending on the used architecture and operating system. These additional files are named `makefile.$CPU.$OS`, where the environment variables CPU and OS should contain the name of the employed architecture and operating system. Usually, under the UNIX operating system these names can be found with the `uname` command: `uname -m -s` (under Linux and Darwin), or `uname -s -p` under Solaris. An exaple of the first case is:

```
% setenv CPU `uname -m`
% setenv OS  `uname -s`
```

In the case of a SUN sparc work station running Solaris the name of the included file resulting from these values for CPU and OS would be `makefile.sparc.SunOS5`. It is convenient to set these two environment variables in your .cshrc file (when using csh or tcsh) in your home directory, or the equivalent thereof .bashrc (using bash). In the latter case you would use:

```
% export CPU=`uname -m`
% export OS=`uname -s`
```

Remember to source the appropriate resource file after adding the above commands:

```
% source ~/.cshrc
```

or

```
% source ~/.bashrc
```

so that the definitions are active in the terminal wndow you are using. The next time you open a window the commands are executed automatically.

The geometry specific routines for each of the four versions are located in the four sub-directories rhf1d, rhsc2d rhsc3d, and rhsphere. The `Makefile` in each of these sub-directories will compile the geometry specific routines and link them with the two general libraries in the parent directory to produce the geometry specific executables. As in the main directory the Makefile includes platform specific macro definitions. The `makefile.$CPU.$OS` files in this case are located in the `rh/makefiles` directory and are named in the same fashion as the ones in the main directory.

In addition, the makefiles in each of the geometry specific subdirectories will build two routines called solveray and backgrcontr. The first program solves the radiation field in a specified direction using the population numbers n and mean intensities J from a converged solution. The second program calculates the separate contributions of different background processes to the opacity for a specified number of wavelengths.

To build the code for each of the four versions, first build the rh main libraries:

```
% make
```

Then go to each of the rhf1d, rhsc2d, rhsc3d, and rhsphere directories and again type make. This should create all the executables.

3 Acessory files

3.1 The Atmos sub-directory

The Atmos sub-directory contains atmospheric input files for all four geometry specific versions the code. The formats of these input files are necessarily different for each case, although they are very similar for the 1-D plane-parallel and spherically symmetric versions. The following naming convention is used (but

not enforced) to distinguish between input files for the different geometric versions. It is encouraged to use the generic template name_N_0[xN_1[xN_2]][_s].atmos, where N_0, N_1, and N_2 refer to the size of the first, and optionally second and third dimensions of the atmosphere, respectively, and s refers to a spherical input atmosphere. Some examples of atmosphere input file names are FALA_80.atmos for a plane-parallel atmosphere with 80 points in depth, FALC_82_s.atmos for a spherical atmosphere with 82 radial points, and ns_63x63x64.atmos for a three-dimensional atmosphere with a 63×63 horizontal grid and 64 depth points.

3.2 The Atoms sub-directory

The Atoms directory contains a number of atomic model input files (.atom) that can be used with all versions of the code, irrespective of geometry. The format of these input files is described in Section **4.2.1**.

In addition, the Atoms sub-directory contains input files with tabulated partition function and ionization potentials for the first 100 elements of the periodic table provided by R. L. Kurucz [1]. The latter files, Atoms/ionpot.input and Atoms/pf_Kurucz.input, respectively, are used when a Kurucz line list is used to calculate background-line opacities. Example files for Kurucz line lists can be found in the Atoms/Kurucz sub-directory.

The Atoms sub-directory contains the file abundance.input, which is used by the background package to read in the abundances values of all elements. By default it contains the solar abundances for the first 100 elements of the periodic table.

Finally, the Atoms/wave_files sub-directory contains files with wavelength lists that can be used to add wavelength points that are not generated from the atomic and molecular input files. Their format is explained in Section **4.4**.

3.3 The Molecules sub-directory

The Molecules directory contains a number of molecular model input files (.molecule) that can be used with all versions of the code, irrespective of geometry. The format of these input files is described in Section **4.3.1**. Typically only di-atomic molecules are considered in the solar atmosphere, but the code accepts multi-atomic molecules of any size, and sets up the required chemical equilibrium network from the composition specified for each molecule in the respective .molecule file.

Example files with molecular line lists are given in sub-directories named after the molecule, e.g., Molecules/CO, and Molecules/CH for the CO and CH molecules, respectively.

3.4 The rh*d/run sub-directories

The run sub-directories in each of the geometric source code directories are the preferred directories to run the respective codes. Each contains the primary input files keyword.input, atoms.input, and molecules.input which specify the keyword settings, and which atomic and molecular models to include.

[1] See: http://cfa-www.harvard.edu/~kurucz

3.5 Utilities in the tools sub-directory

The tools sub-directory contains several small useful programs, for instance to facilitate the production of atomic input files. The programs are built by the central `Makefile` in the tools directory which also uses the same mechanism to import platform dependent makefile macros through `makefile.$CPU.$OS` files as described above. Some programs are also linked to the `librh.a` library in the parent directory. Two particularly useful routines are convertatom to convert atomic models in MULTI format to the format for the code described here, and impact to calculate collisional rate coefficients for a given model atom using the impact approximation (Seaton, 1962) for neutral bound-bound and Van Regemorter's (Van Regemorter, 1962) approximation for ionic bound-bound collisions. See Table 4 for a more complete list of tool programs.

program		purpose
addgrad	—	Calculates radiative damping from level lifetimes
avgmolweight	—	Calculates average molecular weight for atmospheric model
convertatom	—	Converts atomic model from MULTI format
impact	—	Calculates collisional rate coefficients
make_h	—	Creates hydrogen atomic model with arbitrary number of levels
printneff	—	Prints effective quantum numbers of given model atom
waveinfo	—	Lists wavelengths in wavelength table input file

Table 4: List of tool programs.

4 Input files

All four versions of the codes use the primary input files keyword.input and atoms.input, and molecules.input, which are ordinarily placed in the respective `run` directories. Other input files are specified in either of these three primary input files. These include files for the atomic data (for the "active" atom *and* the background atom files), molecular data, model atmosphere, abundance values, and Kurucz line lists for background line opacities. In addition, it is also possible to specify a wavelength input table with additional wavelengths, not collected from the atomic and molecular transitions that are treated in detail, in the keyword.input file. The command line option `-help` shows all available command line options. Note that these command line options may be abbreviated to their unambiguous length.

4.1 The primary input file keyword.input

By default the code looks for the primary input file keyword.input in the current working directory. An alternative file may be specified with the `-input` command line option (which may be abbreviated to `-i`), e.g.,

```
% ../rhf1d -i my_keywords.input
```

which would run the 1-D version from the run sub-directory with the file `my_keywords.input` as the primary keyword input.

The RH code uses three classes of keywords: KEYWORD_REQUIRED, KEYWORD_DEFAULT, and KEYWORD_OPTIONAL. Keywords of the first type are required to be set explicitly in the input file. The code will exit with an error if one of the KEYWORD_REQUIRED is not set. The other two keyword types have built-in default values which are set in the readinput.c file in each of the geometry specific source code directories and do not have to be set explicitly. The difference between the two is that when KEYWORD_DEFAULT keywords are set explicitly in the input file a warning is issued that the default value of that keyword is overridden, while values for KEYWORD_OPTIONAL keywords are accepted silently. Running the code with the `-showkeywords` (optionally abbreviated to `-s`) shows all the keyword settings of the pertinent input file. Table 5 lists all keywords with their type and default values.

The structure of the `keywords.input` file is a list of keyword - value pairs separated by an = sign:

KEYWORD = value.

The comment line character for the `keywords.input` file is as in most other ASCII input files the pound sign (#). Empty lines are allowed and are discarded when the input is read. A warning is issued for unrecognized keywords in the input file (i.e., ones that are not included in the readinput.c), and the code will abort if an illegal value of a KEYWORD_REQUIRED keyword is used.

Table 5: List of general keywords, valid in all versions

keyword	type	default value
ATMOS_FILE	KEYWORD_REQUIRED	
ABUND_FILE	KEYWORD_REQUIRED	
ATMOS_ITOP	KEYWORD_OPTIONAL	
ATOMS_FILE	KEYWORD_REQUIRED	
MOLECULES_FILE	KEYWORD_REQUIRED	
NON_ICE	KEYWORD_OPTIONAL	
WAVETABLE	KEYWORD_OPTIONAL	
N_MAX_SCATTER	KEYWORD_OPTIONAL	
I_SUM	KEYWORD_REQUIRED	
N_MAX_ITER	KEYWORD_REQUIRED	
ITER_LIMIT	KEYWORD_REQUIRED	
NG_DELAY	KEYWORD_OPTIONAL	0
NG_ORDER	KEYWORD_OPTIONAL	0
NG_PERIOD	KEYWORD_OPTIONAL	1

PRD_TRESHOLD	KEYWORD_REQUIRED	
PRD_N_MAX_ITER	KEYWORD_OPTIONAL	8
PRD_NG_DELAY	KEYWORD_DEFAULT	0
PRD_NG_ORDER	KEYWORD_DEFAULT	0
PRD_NG_PERIOD	KEYWORD_DEFAULT	0
XRD	KEYWORD_OPTIONAL	
J_FILE	KEYWORD_REQUIRED	
STARTING_J	KEYWORD_REQUIRED	
BACKGROUND_FILE	KEYWORD_REQUIRED	
HYDROGEN_LTE	KEYWORD_DEFAULT	0
SOLVE_NE	KEYWORD_OPTIONAL	
METALLICITY	KEYWORD_DEFAULT	
KURUCZ_DATA	KEYWORD_OPTIONAL	none
KURUCZ_PF_DATA	KEYWORD_OPTIONAL	none
OPACITY_FUDGE	KEYWORD_OPTIONAL	none
ATOM_OUTPUT	KEYWORD_DEFAULT	atom.out
ATMOS_OUTPUT	KEYWORD_DEFAULT	atmos.out
GEOMETRY_OUTPUT	KEYWORD_OPTIONAL	geometry.out
SPECTRUM_OUTPUT	KEYWORD_DEFAULT	spectrum.out
OPACITY_OUTPUT	KEYWORD_OPTIONAL	none
RADRATE_OUTPUT	KEYWORD_OPTIONAL	none
DAMPING_OUTPUT	KEYWORD_OPTIONAL	
COOLING_OUTPUT	KEYWORD_OPTIONAL	
VMICRO_CHAR	KEYWORD_REQUIRED	
VMACRO_TRESH	KEYWORD_OPTIONAL	0.1
LAMBDA_REF	KEYWORD_DEFAULT	500.0
STOKES_INPUT	KEYWORD_OPTIONAL	
STOKES_MODE	KEYWORD_OPTIONAL	
MAGNETO_OPTICAL	KEYWORD_DEFAULT	
BACKGROUND_POLARIZATION	KEYWORD_DEFAULT	
B_STRENGTH_CHAR	KEYWORD_DEFAULT	
N_THREADS	KEYWORD_OPTIONAL	
LIMIT_MEMORY	KEYWORD_DEFAULT	
ALLOW_PASSIVE_BB	KEYWORD_DEFAULT	
VACUUM_TO_AIR	KEYWORD_OPTIONAL	0

PRINT_CPU	KEYWORD_OPTIONAL	0

The following keywords specify the input parameters applicable to all versions:

ATMOS_FILE – The model atmosphere input file.

ABUND_FILE – The input file with abundance values for the background elements.

ATOMS_FILE – The input file for the model atoms.

MOLECULES_FILE – The input file for the model molecules.

WAVETABLE – Input file for table of additional wavelengths, not gathered from atomic or molecular input files. In the rhf1d version 500.0 nm is automatically added to the wavelengths.

N_MAX_SCATTER – The maximum number of lambda-iterations allowed for the background scattering problem before the main iterations start. Iterations stop when relative changes in mean intensity J fall below the value of ITER_LIMIT (see below).

I_SUM – The level for which the rate equation is replaced with the particle conservation equation to close the set of statistical equilibrium equations. If set to -1 (usually the preferred option) then at each depth the row with the biggest population eliminated.

N_MAX_ITER – The maximum number of main iterations.

ITER_LIMIT – The minimum change in relative populations that should be reached. Iterations stop when this accuracy is reached, or when the maximum number specified in N_MAX_ITER is reached, whichever comes first.

NG_DELAY – The number of main iteration after which Ng convergence acceleration is turned on.

NG_ORDER – The order of Ng acceleration. Usually, second order gives good acceleration, and minimizes storage of the number of previous solutions.

NG_PERIOD – The interval of main iterations at which Ng acceleration is performed. Usually, an interval of 3 with second order acceleration works well.

PRD_TRESHOLD – The change in relative population numbers after which PRD sub-iteration is turned on, in which the PRD scattering integral is iterated by repetitive formal solutions with fixed population numbers, until mean intensity J reaches the same accuracy as the population numbers.

PRD_N_MAX_ITER – Maximum number of PRD sub-iterations.

PRD_NG_DELAY – The number of PRD sub-iteration after which Ng convergence acceleration for the scattering integral is turned on.

PRD_NG_ORDER – The order of Ng acceleration for PRD sub-iterations.

PRD_NG_PERIOD – The interval of iterations at which acceleration of the PRD sub-iterations is performed.

XRD – Boolean flag enabling cross-redistribution between PRD levels that share a common upper level. The RH code determines automatically which levels to include if set to TRUE.

J_FILE – The filename of the mean intensity J input and output. Mean intensities are written to this file at convergence. They are read from this file when starting solution option OLD_POPS_AND_J is used.

STARTING_J – Determines the starting solution for the angle-averaged radiation field J_λ. Allowed options are NEW_J to start with the radiation field calculated from the initial solution of the population numbers, and OLD_J to start from the radiation field obtained in a previous run. In the latter case the file specified in J_FILE should be present and contain a valid solution for the radiation field consistent with the atmosphere, aand tomic and molecular data inputs.

BACKGROUND_FILE – Name of the file to which background absorption, scattering and emission coefficients are written.

HYDROGEN_LTE – If set to TRUE background hydrogen populations are forced to be in LTE. If set to FALSE Non-LTE hydrogen populations should be provided in the atmosphere input file (see ATMOS_FILE).

SOLVE_NE – Boolean flag. Set to TRUE to enable the (LTE) number density of electrons from the atmospheric input data.

METALLICITY – Logarithm of the metallicity scaling factor. If set to for instance -1.0 all metal abundances are scaled by a factor of 0.1.

KURUCZ_DATA – Specifies name of file that contains list of files with line lists in Kurucz format to be used for background line opacity calculations.

KURUCZ_PF_DATA – Input file name for the Kurucz partition function tables for the first 100 periodic table elements. Used when a line list is specified with KURUCZ_PF_DATA.

OPACITY_FUDGE – File that contains wavelength-dependent multiplicative factors that can be applied to the background absorption and scattering coefficients to mimic the UV background line haze.

ATOM_OUTPUT – Output file for atomic data. XDR format.

ATMOS_OUTPUT – Output file for model atmosphere data. XDR format.

GEOMETRY_OUTPUT – Output file for geometry data. XDR format.

SPECTRUM_OUTPUT – Output file with wavelengths and emergent intensities. XDR format.

OPACITY_OUTPUT – Output file for opacities of "active" transitions. XDR format.

RADRATE_OUTPUT – Output file for radiative rates in "active" transitions. XDR format.

DAMPING_OUTPUT – Output file for damping parameters of bound-bound transitions with line shape equal to VOIGT, or PRD.

COOLING_FILE – Output file for net cooling rates.

VMICRO_CHAR – Characteristic value of microscopic velocity broadening in km s^{-1}. Determines the translation of Doppler width into wavelength when setting up the wavelength grid for line transitions. Use a smaller value for closer wavelength spacing.

VMACRO_TRESH – The treshold in km s^{-1} of macroscopic plasma motions above which which velocity shifts are accounted for in the line profile calculation.

LAMBDA_REF – Reference wavelength. In the 1-D plane-parallel version rhf1d opacities at this wavelength are used to convert geometric height scale and column mass scale, and a $\tau_{\lambda_{\rm ref}}$ scale.

STOKES_INPUT – Specifies input file with magnetic field data. When this keyword is specified the polarization in all four Stokes parameters in lines resulting from the Zeeman effect is calculated.

STOKES_MODE – Specifies the iterative behavior in case STOKES_MODE is set. Allowed options are: NO_FIELD, to ignore Zeeman effects, FIELD_FREE, iterate Non-LTE solution with un-polarized profiles and perform a formal solution with polarization only after convergence, POLARIZATION_FREE, iterate with Zeeman-split profiles, but only calculating Stokes I, and perform a formal solution with full polarization only after convergence, and FULL_STOKES, perform full Stokes Non-LTE iterations (this option is often much slower, and not usually necessary with field strengths found in the solar atmosphere).

BACKGROUND_POLARIZATION – Boolean keyword to turn on calculation of linear polarization resulting from scattering of anisotropic radion (in the vertical direction) off electrons (Thomson) and atoms (Rayleigh). Keyword STOKES_MODE should be set to FULL_STOKES when BACKGROUND_POLARIZATION is set to TRUE.

B_STRENGTH_CHAR – Define a characteristic field strength to aid in the distribution of wavelength points in polarized line profiles. Additional equidistant grid points will be centered within Q_WING around the centers of the Zeeman $\sigma_\pm$ and π components.

N_THREADS – Enables paralellization via multi-threading of the formal solution when set to a number larger than 1. Typically the value should be set to the number of available processors on a shared-memory multi-core machine.

LIMIT_MEMORY – Boolean keyword that enables saving of memory space by storing the angle-averaged mean intensity J and line profiles φ to temporary files on disk, rather than keeping them in memory. In particular the line profiles (variables of dimension $N_{\rm space} \times N_{\rm ray} \times N_{\rm wavelength}$) require huge amounts of space in multi-dimensional geometry. Typically disk access is slower than memory access, so this option usually slows down execution.

ALLOW_PASSIVE_BB – Boolean keyword to allow for background lines to be included in the calculation.

VACUUM_TO_AIR – If set to TRUE convert vacuum wavelengths to air values for wavelengths above VACUUM_TO_AIR_LIMIT which is set in `spectrum.h` in the main directory. If set to FALSE only vacuum wavelengths are written to output.

PRINT_CPU – If set to TRUE CPU usage is printed to standard output.

Some keyword options are specific to one or more geometric versions of the code. These are discussed in the subsections below.

4.1.1 Specific keywords for rhf1d

The following keyword(s) are specific to the 1-D plane-parallel version of the code.

keyword	type	default value
NRAYS	KEYWORD_OPTIONAL	
HYDROSTATIC	KEYWORD_OPTIONAL	

Table 6: List of specific keywords for rhf1d

NRAYS – Number of rays used in the angle discretization. The angles are spaced on the zeros of the Gauss-Legendre polynomials as a function of $\cos(\theta)$, where θ is the angle of the ray with the normal direction of the atmosphere.

HYDROSTATIC – Boolean keyword to turn on hydrostatic equilibrium iterations.

4.1.2 Specific keywords for rhsc2d and rhsc3d

keyword	type	default value
ANGLE_SET	KEYWORD_OPTIONAL	
INTERPOLATE_3D	KEYWORD_DEFAULT	

Table 7: List of specific keywords for rhsc2d and rhsc3d

The following keyword(s) are specific to the 2-D and 3-D Cartesian versions of the code.

ANGLE_SET – Defines the set of rays for angle discretization. Currently, discretization from Carlson (1963) is used. Options are: SET_A2, SET_A4, SET_A6, SET_A6, with respectively 2, 6, 12, and 20 angles per octant, and SET_B4, SET_B6, SET_B8, with 6, 12, and 20 angles per octant. The SET_VERTICAL option which provides a single ray pointing in the positive z-direction. In addition, the keyword can be set to the value SET_GL_NiXNa, where N_i and N_a are the number of angles in inclination and azimuth, respectively, per octant.

INTERPOLATE_3D – Option to set interpolation scheme (in 3-D only). Valid options are LINEAR_3D and BICUBIC_3D, for linear and bicubic interpolation in the horizontal planes, respectively. Parabolic interpolation is used in the vertical direction in both versions.

4.2 Atomic input data

4.2.1 Atomic data list `atoms.input`

The `atoms.input` file specifies the filenames of the atomic models that are to be used in ACTIVE and PASSIVE treatment. Populations of ACTIVE atoms are updated according to Non-LTE radiative transfer, while those of PASSIVE atoms are kept fixed, either at LTE values, or at values obtained by reading a previously calculated population number file (see below). The `atoms.input` file is read with routines in `readatoms.c` in the `rh` main directory. The format of the atomic list input file is described described below.

```
# Natom
   4
#
```

```
# Atoms
#  model file        ACTIVE/PASSIVE  INITIAL_SOLUTION   population file

  ../../Atoms/H_6.atom    ACTIVE    ZERO_RADIATION    pops.H.out
  ../../Atoms/C.atom      PASSIVE   LTE_POPULATIONS
  ../../Atoms/O.atom      ACTIVE    OLD_POPULATIONS   pops.O.out
  ../../Atoms/Si.atom     PASSIVE   ZERO_RADIATION
#
```

Empty lines or lines with starting with the comment character # are discarded on read. The first real entry specifies that 4 atoms are included. These atoms, and the manner in which they will be treated, are specified with one line each. Each line has 3 or 4 entries. The first line should **always** specify the hydrogen atom to be used, otherwise the program aborts with an error. Similarly, no more than one entry for the same species is allowed. In each atomic input line the first entry specifies a atomic model input file, the second specifies whether the atom is to be treated in LTE (PASSIVE) or Non-LTE (ACTIVE), the third specifies the starting solution for that atom, and the last specifies the name of the corresponding population number file.

Valid starting solution options are

ZERO_RADIATION – Initial solution of population numbers is computed by setting the the radiation field to zero in the statistical equilibrium equations for the level populations. This option can only be used in ACTIVE atoms. It is ignored for PASSIVE atoms and replaced internally by LTE_POPULATIONS.

LTE_POPULATIONS – Initial solution is set to LTE values.

OLD_POPULATIONS– Initial solution is read from file in forth entry. this requires that the specified file exists and contains valid numbers.

The resulting populations of ACTIVE, after Non-LTE iterations, are written to files with the name `pops.ID.out`, with ID the one- or two-letter atom ID, in upper case. If the population file with that name already exists, it will be overwritten. At startup only the population files of atoms that specify OLD_POPULATIONS as starting solution are read, otherwise the files are ignored, even when they are specified.

In the above example two atoms are treated in Non-LTE, namely hydrogen and oxygen. The starting solution for Oxygen is read from the file `pops.O.out`, which will be overwritten on output. Carbon and silicon are treated in LTE, and the populations of both are set to LTE values.

4.2.2 The format of the `*.atom` atomic input data files

Both active and background model atoms are read by the readAtom procedure in the file `readatom.c` in the main source directory. The format of the atomic input file is described Table 8.

An example of an atomic data input file (for ionized calcium)is given in Appendix A.

input	format
ID	**(A2)**
	Two-character atom ID.
Nlevel Nline Ncont Nfixed	**(4I)**
	Number of levels, lines, continua, and fixed radiation temperature transitions.
level_entries	Nlevel × **(2F, A20, I)**
line_entries	Nline × **(2I, F, A, I, A, 2F, A, 6F)**
continuum_entries	Ncont × **(I, I, F, I, A, F)**
fixed_entries	Ncont × **(2I, 2F, A)**

Table 8: Format of the atomic data input file.

4.2.3 Format of the `kurucz.input` line list input file

In addition to line transitions that are specified as part of PASSIVE atoms in the `atoms.input` file, a list of line transitions can be specified in the `kurucz.input` file set by the value of the KURUCZ_DATA keyword in the `keyword.input` file. The format of the `kurucz.input` is simply a list of files with Kurucz-style line lists (see: `http://cfa-www.harvard.edu/~kurucz`). The format of these files is also listed in the `kurucz.c` source file in the `rh` main directory. The lines that are specified throughKurucz line lists are treated in LTE, with opacities computed with the Saha-Boltzmann relations with values of partition functions provided in the pf_Kurucz.input input file.

To make sure that the wavelengths of the lines that are requested trough the Kurucz line input files are covered it is necessary to add the proper list of wavelengths through the wavetable file specified with the file specified with the value of the WAVETABLE keyword in `keyword.input` (see Section **4.4**).

4.3 Molecular input data

The format of the molecule list data input file is very similar to that of the `atoms.input` (see Section **4.2.1**).

```
# Nmolecule
#
 5

# molecules
#
  ../../Molecules/H2.molecule    PASSIVE    LTE_POPULATIONS
  ../../Molecules/CH.molecule    PASSIVE    LTE_POPULATIONS
  ../../Molecules/CO.molecule    ACTIVE     LTE_POPULATIONS
```

```
../../Molecules/CN.molecule    PASSIVE    LTE_POPULATIONS
../../Molecules/H2O.molecule   PASSIVE    LTE_POPULATIONS
```

The first molecule in the list should **always** be H_2. An fatal error is created if this is not the case, or if a duplicate species is specified in the files.

In contrast to the atomic case, the only option for initial solution of the molecular pipulation numbers is LTE_POPULATIONS. In the above sample the CO molecule is treated in Non-LTE, the others are treated in LTE.

4.3.1 Molecular data list `molecules.input`

A sample molecular input file (for CO) is shown in Appendix B. Molecules are read by the readMolecule procedure in the `readmolecule.c` file.

4.4 The wavetable input file

A list of additional (to the ones that are automatically generated from the ACTIVE bound-bound and bound-free transitions in the atoms and molecules specified in the `atoms.input` and `molecules.input` files) wavelength grid points can be specified with the file named in the WAVETABLE keyword of the `keyword.input` file. This file is a binary file with the following format, and can be generated in a manner as specified in the following example:

```
IDL> Nlambda = 3000L
IDL> lambda  = air_to_vacuum(429.0D0 + 3.0*dindgen(Nlambda)/(Nlambda - ))
IDL> openw, 1, /XDR, 'Gband.wave'
IDL> writeu, 1, Nlambda, lambda
IDL> close, 1
```

This example creates a wavelength file, `Gband.wave` with 3000 wavelength points stretching from 429.0 nm to 432.0 nm centered around the G band at 430.5 nm. The number of wavelengths (Nlambda) is specified in a 4-byte integer, and the wavelengths (lambda) are written as an array of double precision floats. Internally the code computes in vacuum wavelengths, so before storing wavelengths have to be converted from air to vacuum wavelengths (the required routine `airtovacuum.pro` is supplied with the IDL source files in the distribution). Upon output these wavelengths are converted to air again (if the keyword VACUUM_TO_AIR is set to TRUE, which is the default).

4.5 Atmospheric input files

Input files for atmospheric data in the one-dimensional version rhf1d have to be given in the format of the MULTI transfer code. This is an ASCII format. Several examples are provided with the distribution. The atmospheric input file for the spherical version rhsphere is almost the same, apart from the radius of the atmosphere, which is specified in km as the second parameter on the third line (not counting comment lines)

of the `.atmos` file, after the log of gravity. For the other two versions the format is binary. In each case it is specified in detail in the `readatmos.c` file in the source directory for each of the versions.

4.5.1 Input files for Magnetic fields

Magnetic field strength B, and field inclination γ and azimuth χ in case of Zeeman effect calculations, are specified in a separate file (usually with the extension `.B`). The format for these files is binary in all cases, with double float arrays of dimension of the atmospheric quantities for each of the three quantities. For instance the following IDL commands would write a constant magnetic field of 1000 Gauss with an inclination of 45 deg and azimuth of zero to the file `two-d.B` for a two-dimensional atmosphere of 100×64 points:

```
IDL> GAUSS_TO_TESLA = 1.0D-4
IDL> openw, 1, /XDR, 'two-D.B'
IDL> writeu, 1, dblarr(100, 64) + 1000.0*GAUSS_TO_TESLA, $
  dblarr(100, 64) + 45.0/!RADEG, dblarr(100, 64}
IDL> close, 1
```

Note that field strengths are given in Tesla, while both arrays of angles are given in radians.

4.6 The `ray.input` files

The angular grid for used in Non-LTE computations is typically dictated by the requirement of adequate sampling of all solid angles, and therefore, does not in general include a given direction for which the emergent intensity is requested (i.e., in case of comparison with a certain observation). In particular, the emergent intensity in the vertical direction is ususually not part of the iterative solution. For this purpose a seperate program is provided that performs a formal solution, using the population numbers and angle-averaged mean intensity from the converged Non-LTE iteration. This code in the three versions rhf1d, rhsc2d, and rhsc3dis called `solveray`. The exception is rhsphere, which always includes the vertical direction, and does not have this additional solver available. The input file for `solveray` is `ray.input` its format is different for all three geometry versions and is described in the following subsections.

4.6.1 rhf1d

```
## ray.input for rhf1d
mu
Nsource
```

The format for `ray.input` in the one-dimensional version is given above. *mu* is the $\cos(\theta)$ for which the emergent intensity is to solved. The intensity in the vertical direction is obtained by setting $mu = 1.0$. The solution is written to the file `spectrum_`μ, for example, `spectrum_1.00` for the vertical. N_{source} is the

number of source function and opacity wavelengths that are stored with the solution. Typically, this is set to 0.

4.6.2 rhsc2d

```
## ray.input for rhsc2d
mu_x  mu_z
Nsource
```

In the two-dimensional version μ_x and mu_z are the angles with the x and z axes, respectively. The intensities in the vertical direction are obtaimed by setting $\mu_x = 0.0$ and $mu_z = 1.0$. Results are written to the file `spectrum_`μ_x`_`μ_z. $N_{\rm source}$ as above.

4.6.3 rhsc3d

```
## ray.input for rhsc2d
mu_x  mu_y
Nsource
```

In the two-dimensional version μ_x and mu_y are the angles with the x and y axes, respectively. The intensities in the vertical direction are obtaimed by setting $\mu_x = 0.0$ and $mu_y = 0.0$. Results are written to the file `spectrum_`μ_x`_`μ_y. $N_{\rm source}$ as above.

5 Running the codes

Go to one of the four geometry-specific subdirectories, type make to build the particular version you wanted (say the 1-D plane version rhf1d). Go to the run subdirectory and customize the keyword.input, and atoms.input molecules.input files. Then type

```
% ../rhf1d
```

to start the code, and similar commands for the other versions in their respective run directories. To obtain emergent intensities in additional directions run the `solveray` program (see Section **4.6**):

```
% ../solveray
```

6 Output files

Most output files are written as unformatted files using the XDR (eXternal Data Representation) format. See `man xdr` on most Unix machines, or refer to the chapter on input/output in the IDL manual. In this

way output generated on a computer with one architecture can be read transparently with IDL on one with a different architecture.

In addition to the environment variables OS and CPU a third variable should be defined, namely RH_IDL_PATH, which should point to the directory in which the IDL analysis files are placed, for instance like:

```
% setenv RH_IDL_PATH $HOME/src/rh/idl
```

, or

```
% export RH_IDL_PATH=$HOME/src/rh/idl
```

, in case you are using bash.

7 IDL output analysis

To read the output data and look at the results a widget driven IDL program analyze is available. The program is run from the directory from which the transfer code was run. It has a point-and-click interface to get a first look at all kinds of output data such as line profiles, source functions, contribution functions, fluxes, level populations etc. It is convenient to precompile all the subroutines for analyze. There is a procedure initrh.pro that will do this:

```
IDL> @initrh
IDL> analyze
```

The graphical user interface will now allow you to check many results by point-and-click.

In addition there is a procedure `readall.pro` that will read most of the output data and, through the use of common blocks, will make the data available at the main programming level, mostly in the form of structures.

```
IDL> .r readall
IDL> help, NAMES='*'
IDL> help, /STRUCTURES
```

The emergent intensities calculated for additional directions can be read by applying the `readray.pro` function to the resulting `spectrum_*` files, for instance in the two-dimensional case for the vertical direction:

```
IDL> ray = readray('spectrum_0.00_1.00')
```

8 Units

All physical quantities in the code are specified in the International System of Units Taylor (SI, see e.g., 1995), which is available on the Web [1]). Some input routines accept variables with cgs units, mostly to be compatible with MULTI input files (see Sect **8.2**). It is good practice in case of these exceptions, however, to convert the variables in question to SI units immediately in the input routine. Table 9 lists a number of

Name	Value	purpose
NM_TO_M	$1.0\,10^{-9}$	nm to m
CM_TO_M	$1.0\,10^{-2}$	cm to m
KM_TO_M	$1.0\,10^{+3}$	km to m
ERG_TO_JOULE	$1.0\,10^{-7}$	erg to J
G_TO_KG	$1.0\,10^{-3}$	g to kg
MICRON_TO_NM	$1.0\,10^{+3}$	μm to nm
MEGABARN_TO_M2	$1.0\,10^{-22}$	Mb to m^2

Table 9: Unit conversions

conversion factors that can be used in the code to explicitly show where such a conversion is done.

Table 10 lists the mathematical constants that are defined in constant.h.

name	value
PI	3.14159265358979
SQRTPI	1.77245385090551

Table 10: Mathematical constants

8.1 Physical constants

Table 11 lists the physical constants that are defined in include file constant.h together with their values and units.

8.2 Exceptions for specific input files

References

Carlson, B. G. 1963, in B. Alder, S. Fernbach, M. Rotenberg (eds.), Methods in Computational Physics, Vol. 1, 1–43, Academic Press, New York

[1] http://physics.nist.gov/PhysRefData/contents.html

name	value	units	represents
CLIGHT	$2.99792458\,10^{+8}$	m s^{-1}	speed of light
HPLANCK	$6.6260755\,10^{-34}$	J s	Planck's constant
KBOLTZMANN	$1.380658\,10^{-23}$	J K^{-1}	Boltzmann's constant
AMU	$1.6605402\,10^{-27}$	kg	atomic mass unit
M_ELECTRON	$9.1093897\,10^{-31}$	kg	electronic mass
Q_ELECTRON	$1.60217733\,10^{-19}$	C	electronic charge
EPSILON_0	$8.854187817\,10^{-12}$	F m^{-1}	permittivity of vacuum
RBOHR	$5.29177349\,10^{-11}$	m	Bohr radius
E_RYDBERG	$2.1798741\,10^{-18}$	J	Rydberg's constant (infinite mass)
EV	$1.60217733\,10^{-19}$	J	electron Volt

Table 11: Physical constants

Ng, K. C. 1974, J. Chem. Phys., 61, 2680

Rybicki, G. B., Hummer, D. G. 1991, A&A, 245, 171

Rybicki, G. B., Hummer, D. G. 1992, A&A, 262, 209

Seaton, M. J. 1962, Proc. Phys. Soc. London, 79, 1105

Taylor, B. N. 1995, NIST Special Publication 811. Guide for the use of the International System of Units, Technical Report 811, National Institute of Standards and Technology, United States department of Commerce

Uitenbroek, H. 2000, ApJ, 536, 481

Van Regemorter, H. 1962, ApJ, 136, 906

A An example model atom input file: Ca II

The following is an example of an atomic data input file for singly ionized calcium.

```
# Calcium II: 5 levels + continuum (bound-free from 4P treated with fixed rates)
  CA

# Nlevel  Nline   Ncont   Nfixed
    6       5       3        2

#  E[cm^-1]    g           label[20]         stage    levelNo
#                |----|----|----|----
     0.000   2.00    'CA II 3P6 4S 2SE    '    1         0
 13650.212   4.00    'CA II 3P6 3D 2DE 3  '    1         1
 13710.901   6.00    'CA II 3P6 3D 2DE 5  '    1         2
 25191.541   2.00    'CA II 3P6 4P 2PO 1  '    1         3
 25414.427   4.00    'CA II 3P6 4P 2PO 3  '    1         4
 95785.470   1.00    'CA III 3P6 1SE      '    2         5

# j   i       f     type  Nlambda symmetr  qcore  qwing  vdWapprx          vdWaals            radiative  Starck
#                                                                    H              He
  3   0  3.300E-01     PRD  50     ASYMM   10.0   450.0   UNSOLD   1.50   0.00   1.00   0.00   1.48E08  3.0E-12
  4   0  6.600E-01     PRD  50     ASYMM   10.0   450.0   UNSOLD   1.50   0.00   1.00   0.00   1.50E08  3.0E-12
  3   1  4.420E-02   VOIGT  25      SYMM    4.0   100.0   UNSOLD   1.50   0.00   1.00   0.00   1.48E08  3.0E-12
  4   1  8.830E-03   VOIGT  25      SYMM    4.0    60.0   UNSOLD   1.50   0.00   1.00   0.00   1.50E08  3.0E-12
  4   2  5.300E-02   VOIGT  25      SYMM    4.0   100.0   UNSOLD   1.50   0.00   1.00   0.00   1.50E08  3.0E-12

# Photoionization rates
# j   i  alpha [m^-2]   Nlambda     Wavel. Dep.   lamb_min [nm]

#   CA II 3P6 4S 2SE
  5   0   2.0363E-23      15        EXPLICIT        35.0
  104.4   2.0363E-23
  100.0   2.0974E-23
   95.0   2.1455E-23
   90.0   2.1704E-23
   85.0   2.1715E-23
   80.0   2.1489E-23
   75.0   2.1025E-23
   70.0   2.0332E-23
   65.0   1.9419E-23
   60.0   1.8302E-23
   55.0   1.7001E-23
   50.0   1.5539E-23
   45.0   1.3944E-23
   40.0   1.2248E-23
   35.0   1.0486E-23

#   CA II 3P6 3D 2DE 3
  5   1   6.1484E-22      20      HYDROGENIC        30.0

#   CA II 3P6 3D 2DE 5
  5   2   6.1484E-22      20      HYDROGENIC        30.0

# Fixed transitions
# j   i    Strength        Trad      Option

#   CA II 3P6 4P 2PO 1
  5   3   2.3823E-22      4925.0     TRAD_PHOTOSPHERIC

#   CA II 3P6 4P 2PO 3
  5   4   2.3823E-22      4925.0     TRAD_PHOTOSPHERIC

# Collisional bound-bound strengths (integrated over Maxwellian)
# OMEGA is dimensionless
 TEMP    6          3000.0     5000.0     7000.0    15000.0    50000.0   100000.0
 OMEGA   1  0    2.378E+00  2.284E+00  2.203E+00  1.920E+00  1.961E+00  1.846E+00
 OMEGA   2  0    3.568E+00  3.426E+00  3.304E+00  2.879E+00  2.942E+00  2.770E+00
 OMEGA   2  1    1.778E+00  2.869E+00  4.693E+00  1.266E+01  3.189E+00  7.926E+00
 OMEGA   3  0    4.842E+00  5.288E+00  5.548E+00  6.148E+00  8.482E+00  1.200E+01
 OMEGA   4  0    9.683E+00  1.058E+01  1.110E+01  1.230E+01  1.696E+01  2.401E+01
 OMEGA   4  3    8.511E-01  1.364E+00  1.878E+00  4.002E+00  1.111E+01  1.873E+01
 OMEGA   3  1    1.360E+01  1.433E+01  1.496E+01  1.699E+01  2.280E+01  2.836E+01
 OMEGA   4  1    6.316E+00  6.488E+00  6.623E+00  7.033E+00  8.080E+00  8.899E+00
```

```
 OMEGA   3  2    4.338E+00  4.282E+00  4.203E+00  3.929E+00  3.315E+00  2.641E+00
 OMEGA   4  2    2.504E+01  2.636E+01  2.751E+01  3.122E+01  4.217E+01  5.284E+01

# Collisional bound-free strengths
# CI is given in m^3 K^-1/2
 TEMP    6          3000.0     5000.0     7000.0    15000.0    50000.0   100000.0
 CI      0  5    4.580E-18  4.580E-18  4.580E-18  4.580E-18  4.580E-18  4.580E-18
 CI      1  5    1.613E-16  1.613E-16  1.613E-16  1.613E-16  1.613E-16  1.613E-16
 CI      2  5    1.614E-16  1.614E-16  1.614E-16  1.614E-16  1.614E-16  1.614E-16
 CI      3  5    7.272E-17  7.272E-17  7.272E-17  7.272E-17  7.272E-17  7.272E-17
 CI      4  5    7.295E-17  7.295E-17  7.295E-17  7.295E-17  7.295E-17  7.295E-17
 END
```

The model has 6 levels and 5 lines (H, K and the infrared triplet). The H & K resonance lines are treated with PRD, the other lines are treated with CRD and a VOIGT profiles (another option is GAUSS, for a Gaussian profile). Bound–free transitions from ground level and the meta-stable $3D$ levels is treated in detail, while transitions between $4P$ and the continuum are treated with a fixed radiation temperature. In this case the radiation temperature is prescribed by the photospheric option TRAD_PHOTOSPHERIC in the LINEAR-B tradition. Other options are TRAD_CHROMOSPHERIC and TRAD_ATMOSPHERIC.

B An example molecular input file: CO

```
# CO model for Non-LTE calculations
  CO

# Charge
  0

# List of constituent atoms of the form nA, mB, pC for molecule AmBnCp
  C, O

# Ediss [eV]
  11.091

# Fit type for partion function and equilibrium constant
  KURUCZ_85

# Tmin  and Tmax [K]
  1.0E+3  12.0E+3

# pf_coef
  7   4.51349  18.4221  -50.0599  102.208  -128.504  87.8414  -24.8533

# eqc_coef
  6  -49.0414  14.0306  -26.6341  35.3827  -26.5424  8.32385

# Filename with line data

#-#  /data/rhsc2d/uitenbr/CO/vmax=9_Jmax=120_dv=1_26
  /data/rhsc2d/uitenbr/CO/vmax=3_Jmax=49_dv=1_26

# Nlambda and Qwing

  21    3.25

## end
```