MULTI user's manual

Version 2.3 - March 2015

Contents

1 Introduction

This manual describes the version 2.3 of the radiative transfer code MULTI. A detailed documentation of MULTI version 1.0 is found in *Carlsson, M.: 1986, Uppsala Astronomical Observatory Report 33: A Computer Program for Solving Multi-level Non-LTE Problems in Moving or Static Atmospheres* (http://folk.uio.no/matsc/mul23/report33.pdf) with documentation of the changes from version 1.0 to 2.3 in
http://folk.uio.no/matsc/mul23/mul23.pdf
This manual is meant as a practical guide on how to get started.

2 Installation

2.1 How to get the distribution

The MULTI 2.3 distribution is available at http://folk.uio.no/matsc/mul23. Download the tar file and unpack the distribution.
`tar xvf mul23.tar`
The directory `mul23` will be created together with subdirectories
`doc idl input run source source_dp`
If you want to plot FTS atlas spectra (used in the exercises) you also need to download the file `ftsatlas.idlsave` and place it in the `idl` subdirectory.
If you want to try the 3D wrapper routine you need to download the file `atmos3d.t246.6` and place it in the `run` subdirectory.
MULTI exists in a single precision version and a double precision version. It is recommended to *always* run the double precision version. You create the double precision version from the single precision one by:
`cd mul23/source`
`./make_mul23_dbl.csh`
answer `y` to the question whether it is OK to delete all files in the `source_dp` directory. When you run the script you may get warning messages like
`rm: No match.`
because there are no files to delete.

2.2 Making the executables

You need to specify the name of the Fortran compiler, link command and compile/link options. This is most easily done in your startup script file (`.cshrc` for cshell and derivates, `.bashrc` for bash-shell) in your home directory. Example lines for the ifort compiler in the `.cshrc` file:

```
setenv FC ifort
setenv LINK ifort
setenv FOPT "-O2 -fpe0"
```

In the `.bashrc` file you would instead enter
```
export FC=ifort
export LINK=ifort
export FOPT="-O2 -fpe0"
```

After entering the proper lines you need to get the changes activated. This is done either by logging out and in again or by sourcing the script-file.

Go to the double precision source directory and make the executables:
```
cd ../source_dp
make mul23lus.x
make mul23gus.x
```

The first make command will make the version with a local operator, the second make command will make the global operator version (the meaning of the other letters is `u` for the Uppsala background opacity package and `s` for a version using scratch-files for some variables to save usage of core memory). You may get warning messages about arithmetic if-statements being obsolete feature.

3 Running the code

It is recommended to run MULTI from a directory separate from the source-code directory. You first establish a link between the executable in the `source_dp` directory and the `run` directory. This needs to be done only once:
```
cd ../run
ln -s ../source_dp/*.x .
```

You can now use the script `run.csh`. It accepts the name of the atom as first argument, the name of the atmosphere as the second and the version of `mul` as third argument:
```
./run.csh ca6 val3c 23gus
```

You should get something like the following printout on your screen:

```
Atom designation is: ca6
Atmos designation is: val3c
Version is: 23gus

 LOG(EMAX (   5,  87))=    2.6476  EMAX (   5,  87)= 4.4427E+02
 LOG(EMAXJ(   2,  87))=   -0.0010  EMAXJ(   2,  87)= 9.9777E-01

 LOG(EMAX (   4,  10))=    1.3479  EMAX (   4,  10)= 2.2281E+01
 LOG(EMAXJ(   5,   1))=   -0.4995  EMAXJ(   5,   1)= 3.1663E-01

 LOG(EMAX (   3,  16))=   -0.8869  EMAX (   3,  16)=-1.2975E-01
```

```
 LOG(EMAXJ(   2,  31))=   -0.8032  EMAXJ(   2,  31)=-1.5733E-01

 LOG(EMAX (   5,  10))=   -1.2332  EMAX (   5,  10)=-5.8454E-02
 LOG(EMAXJ(   2,  24))=   -1.6717  EMAXJ(   2,  24)= 2.1295E-02

 LOG(EMAX (   4,   7))=   -2.0205  EMAX (   4,   7)=-9.5384E-03
 LOG(EMAXJ(   2,  24))=   -2.5823  EMAXJ(   2,  24)=-2.6167E-03

 LOG(EMAX (   4,   6))=   -3.0136  EMAX (   4,   6)=-9.6913E-04
 LOG(EMAXJ(   1,   1))=   -3.2960  EMAXJ(   1,   1)=-5.0584E-04
0.736u 0.121s 0:00.87 97.7% 0+0k 5+5io 0pf+0w
```

The printout is the maximum relative change in population density (`EMAX`) and radiation field (`EMAXJ`) for each iteration with the numbers in parenthesis being the level number (or transition number) and depth where the change was the largest. The above example means that after 6 iterations the maximum relative change was smaller than 10^{-3} which was the convergence criterion set in the input file.

4 Input files

MULTI reads a number of input files with the name in all upper case. The run script links the appropriate input files from the `input` directory into the `run` directory. In all input files lines starting with * in the first column are treated as comment lines. The input files used are:

`ABSDAT`	Background opacity data
`ABUND`	Abundances used for the background opacity calculation
`ATMOS`	Atmospheric model
`ATOM`	Atomic data
`DSCALE`	Depth discretization to be used
`INPUT`	Switches regulating how the run should be done

These files must be present when running. In addition there are a number of optional input files needed for special cases:

`ABSLIN`	list of lines to include as background opacity
`ABSMET`	opacity sampling file for line blanketing
`ABVAR`	abundance as function of depth
`ATOM2`	detailed atomic data for a given transition
`JFIX`	radiation field to be used to calculate line blanketing
`RSTRT`	populations to start from if restart

4.1 ABSDAT

This file contains data for the calculation of background opacities with the Uppsala package for calculating continuous opacities. The input specifies partition function data, photoionization cross-sections etc. This file is read in *fixed format* and when modified this

has to be taken into account — do not change the format of the data!! For a detailed description of the data, see documentation in the source code. For a condensed explanation see `mul23.pdf`.

4.2 ABUND

Contains abundances used for the background opacity calculation. The format is
(ID abundance) for all elements
the number of elements is specified in `ABSDAT`. abundance can be given either relative to hydrogen (H=1.0) or on a logarithmic scale where the abundance of hydrogen is 12. If molecules are calculated and the atom is a possible constituent in a molecule (H, C, N, O) the amount of the species bound in molecules is calculated first and the remainder is used for the non-LTE calculation. The total abundance of the element calculated in detail is then taken from the `ABUND` file rather than from the `ATOM` file which is the default.

4.3 ATMOS

Gives the atmospheric model. The format is:
ID (text string)
Depth scale type (only first character matters: M, T or H)
lg(g) (cgs)
Number of depth-points in ATMOS
(lg(depth), Temperature, Ne[cgs], Vz[km/s], Vturb[km/s]) as function of depth
(NH(n=1,5), NP) as function of depth

The depth-scale can be either lg(column mass) (M), lg(tau 500nm) (T) or height in km (H). Note that the depth-scale given in the `ATMOS` file is not the one used in the non-LTE calculation — it is only used to specify the atmospheric model as function of depth. The actual deph-discretization to be used for the non-LTE calculations is given in the file `DSCALE`. Vz is positive for upward velocity. Points are given from the top of the atmosphere (index runs from the top of the atmosphere downwards).

The hydrogen population numbers for the first five bound levels and the number of protons are used to calculate the total density and the background opacity due to bound-free absorption of hydrogen. If NH is not given, LTE is assumed and the mass density and the hydrogen population numbers are calculated from the electron density and the temperature. If one is not sure the equation-of-state is the same for the code giving the electron density this may be dangerous and it may then be better to calculate the hydrogen population densities from a given density to make sure the mass density is correct.

4.4 ATOM

Gives the atomic data. Format:
ID

abundance atomic-weight
NK NLINE NCONT NRFIX
(ecm, g, label, ion) for all NK levels
(J, I, F, NQ, QMAX, Q0, IW, GA, GVW, GS) for all NLINE bound-bound transitions
(J, I, A0, NQ, LMAX, LMIN) for all NCONT bound-free transitions
(J, I, P, A0, TRAD, ITRAD) for all NRFIX fixed transitions
collisional routine name
collisional data

where ecm is energy in cm^{-1}, g is the statistical weight, label is a 20 character string, ion is the ionization stage (1 for neutral, 2 for singly ionized, etc).

J is upper level index, I lower level index, F oscillator strength, NQ number of frequency points, QMAX last frequency point, Q0 is transition frequency from Doppler core to Lorentzian wings (both in units of typical doppler width set in `INPUT`), IW is 0 for normal bound-bound transitions, is 1 for transitions with wavelength dependent background opacity and is > 1 for transitions with an absorption profile consisting of a weighted average of IW Voigt profiles, see `mul23.doc` for an explanation. GA is the inverse lifetime of the upper level, GVW the van der Waals broadening parameter (< 20 is interpreted as a multiplicative factor to the classical value, > 20 is a value according to Antree, Barklem & O'Mara. GS is the Starck broadening parameter. If QMAX is negative, the detailed frequency points are given.

A0 is the cross-section at the edge (cgs), LMAX is the wavelength at the photoionization edge in Ångström, LMIN is the minimum wavelength to be considered. The photoionization cross-section wavelength variation is assumed to be hydrogenic. If LMAX is negative the detailed photoionization cross-sections are read as (XL, ALPHA) with XL the wavelength in Ångström and ALPHA the cross-section (cgs) for all frequency points starting with the edge and given in decreasing XL.

For fixed transitions, see the ATOM routine and `mul23.doc` for documentation.

It is recommended to use the collisional routine GENCOL instead of writing individual routines for each atom. See the source code for documentation.

4.5 `DSCALE`

Gives the depth-discretization to be used. It is often necessary with some experimentation to get a depth-discretization that resolves the gradients in optical depth. The format of the file is:
ID
Depth scale type (only first character matters: M, T or H) (must be the same as in `ATMOS`)
NDEP lg(tau_500nm)(top_point)
(lg(depth) for all NDEPTH points

If NDEP is negative you will get an equidistant depth-scale and instead of giving all depth-points you give the top and bottom ones.

4.6 INPUT

The file INPUT contains variables that regulate starting approximation, convergence criterion etc.

The full list of INPUT variables with explanations is given in Section ??.

5 Output files

After a run you should always check the output file JOBLOG. Error messages are written to this file (e.g. problems with dimensions being too small), warning messages (e.g. negative opacity) and information messages (e.g. which background absorbers have been excluded). If the run was successful there is the final line NORMAL END in the JOBLOG file.

Other output files are OUT, RSTRT2, DSCAL2, IDL1, IDLNY, IDLCNT, IDLOPC, JNY. The run-script run.csh renames these (except DSCAL2 to xxxxxx.atom_atmos, e.g. out.ca6_val3c. All output files except OUT are in binary format and are read with IDL routines, see Section ??.

6 Changing dimension variables

MULTI was written in strict Fortran-77 standard. That means no lower-case characters in the source-code, variable names restricted to eight characters, no dynamic allocation of memory etc. This means that it is necessary to set statically the dimensions of the variables. This is done in three files: PARAM PARAMW PARAMO. If dimensions have been set too low this is normally detected when running MULTI and a message will be written to the file JOBLOG with information on which variable in the above files need adjustment. All variables set the maximum value that can be used (maximum number of depth points in the atmosphere etc). The meaning of the variables that you need to worry about are:

PARAM	
variable	maximum number of:
MDEP	depth points in the atmosphere
MK	energy levels in the model atom
MLINE	bound-bound transitions in the model atom
MWIDE	bound-free transitions plus bound-bound transitions with frequency dependent background opacity
MRFIX	transitions treated with fixed rates
MQ	frequency points in one transition
MMU	angle points

PARAMW	
MDEP1	depth points in the atmosphere
MK1	energy levels in the model atom

PARAMO	
MT	depth points in the atmosphere

The reason you have three variables that set the maximum number of depth points in the atmosphere (`MDEP, MDEP1, MT`) and two for the maximum number of energy levels in the model atom (`MK, MK1`) is that for the global operator, most memory is consumed by variables with dimension `MDEP1, MK1` and one can save memory by setting those to the actual values needed for a particular run and save compilation time by setting `MDEP` and `MK` to larger values. `MT` is separate in order to make the opacity package as self contained as possible. The other dimensioning variables in `PARAMO` have to do with the background opacity package and if you are brave enough to change anything in the background opacity treatment you probably know the meaning of those variables.

7 3D wrapper

There exist a 3D wrapper that runs MULTI column-by-column in a 3D atmosphere. Each column is treated as an independent atmosphere so there are no rays going through the different columns. It is possible to get a 1.5D run by first solving the non-LTE problem column-by-column and then interpolate the population densities to whatever direction is of interest and then do just the formal solution with given population densities.

The 3D wrapper can be compiled in four versions: with local or global operator and with serial code or MPI-parallelized code:

```
make mul23luc_3d.sx
make mul23guc_3d.sx
make mul23luc_3d.x
make mul23guc_3d.x
```

In addition to the normal MULTI input files you need an atmosphere file in 3D. This is in binary format with the format evident from the IDL procedure `watmos_multi3d`. You

also need a file `INPUT_3D` that gives the switches etc for 3D runs. An example follows:

```
0   3           icont_opt (=-1 means fill in out3d, otherwise 0), iredo_max (=3 is
'dtrho'         dscopt (orig, dtrho or dscal2)
1 504 1         ix0,ix1,ixstep
1 504 1         iy0,iy1,iystep
-1 51           krsave, nusave if(krsave.lt.0) you give nrad_write and kr for trans
 3              nrad_write
 2              kr=0
 4              kr=1
 5              kr=2
  3             nruns
'cb24bi.t=69.00'
'cb24bi.t=69.10'
'cb24bi.t=69.20'
```

`krsave` and `nusave` gives the line-number for the line you want monochromatic opacities written to file `x3d`. If `krsave` is negative you give the number of lines where you want intensities written to file `Iv3d` in variable `nrad_write` and that number of lines with the transition indices `kr`.

The output comes in a number of files described in the last part of the IDL variables section.

8 INPUT variables

DIFF Diffusion parameter. When $\Delta\tau_{\nu\mu}$ is larger than DIFF the diffusion approximation is used in routine BMAT. A value of 5 has normally been used without causing problems.

ELIM1 When the maximum relative correction in the population numbers, EMAX, decreases below this value, the matrix W is not recalculated and only back-substitutions are performed in EQSYST. This saves a lot of computing time but the value has to be sufficiently small so that the matrix is not too far from the correct one. A value of 0.1 has been proven safe.

ELIM2 When EMAX decreases below this value the iteration procedure is stopped and the populations are then used in the formal solution. The difference between the populations and the infinitely converged populations is normally a factor 5–10 smaller than EMAX. The value chosen depends on the internal accuracy wanted. A value smaller than 0.01 is seldom neither needed nor meaningful since other errors (basic assumptions, depth discretization etc) are larger.

QNORM All relative frequencies in the program are in a unit determined by a scaling constant, the value of which is given by QNORM. The unit of QNORM is km per second at line center. The value is arbitrary and a change of QNORM results in a scaling of all the frequencies. It is recommended to set QNORM to a typical Doppler width in km per second at line center. It is then easier to specify the frequency quadrature, to locate the transition from the Doppler core to the wings in strong lines etc.

THIN When $\tau_{\nu\mu}$ is less than THIN the radiation field is in effect lambda iterated. This improves the stability properties and reduces the CPU time required for each iteration, see Section 3.2 in the Uppsala report. A value of 0.1 has been optimal in almost all cases studied.

IATOM2 It is possible to go through the formal solution with a new set of atomic levels or transitions read from the file ATOM2. The converged populations are then redistributed to the new levels that have the same label as the old ones. This is done proportional to the statistical weights. The file ATOM2 is only read if IATOM2=1. This option can be used such that the file ATOM contains the terms and conglomerate lines while ATOM2 contains the explicit sublevels and lines and/or other frequency points.

ICONV The populations and the ionization fractions are always written to the file OUT (if $IWN \neq 0$). The rest of the printout will depend on the value of ICONV. If ICONV=0 only a converged solution will give detailed printout, If ICONV=1 a detailed printout will always be attempted. Normally ICONV is set to 0 to avoid time consuming nonsense printouts and possible arithmetic overflows from a non-converged run. In some instances this printout is of interest even

if EMAX is greater than ELIM2; notably when ITMAX=0. ICONV=1 will then force these detailed printouts.

IHSE Used only if the model atom is hydrogen. The value then determines if hydrostatic equilibrium (HSE) integrations should be performed.

ILAMBD Determines the number of lambda iterations to be performed on the starting approximation determined by ISTART. A value larger than 1 is hardly needed.

IOPAC When a restart is done with no changes influencing the background opacities it is not necessary to recompute these. IOPAC=0 will read the file INIT and use the old opacity file OPC. The normal value is IOPAC=1. For background opacities in non-LTE values of 2, 3 and 4 are used. IOPAC=4 writes the file BMET2, IOPAC=2 reads the file BMET, IOPAC=3 reads BMET and writes BMET2, see section **??**.

ISTART Determines the starting approximation together with ILAMBD. ISTART=0 sets the intensities equal to zero in the rates and then solves the statistical equilibrium equations. ISTART=1 sets the populations equal to the LTE values. ISTART=-1 reads the starting approximation from the file RSTRT; for the format see routine WRSTRT. Such a file is written by the routine WRSTRT at the end of each run and this file can then be used for a restart or to get additional printouts without running the whole problem anew. For the latter application set ITMAX=0 and ICONV=1. ISTART ¿ 1 uses an escape probability starting solution with ISTART-1 iterations from $n = n^{LTE}$

ISUM The statistical equilibrium equation for level ISUM is replaced by the particle conservation equation. If the statistical equilibrium equation for one level has much smaller coefficients than the others it should NOT be replaced. This is often the case for the continuum level. If ISUM=0 the code will choose the level with the highest population density to be replaced - normally a safe option. If numerical instabilities seem to be present a good first try is to change the value of ISUM.

ITMAX The maximum number of iterations allowed. Most cases studied converge in 3–8 iterations, hydrogen with HSE integrations requires more, 10–20. Using a local operator also gives slower convergence. With collisional-radiative switching a very large value is needed, 50–100. If the populations seem to converge but ITMAX iterations were insufficient to meet the ELIM2 criterion it is possible to continue from the partially converged solution, see variable ISTART. A value of 0 makes it possible to do only the formal solution for known populations, see variable ICONV.

ITRAN Determines the method of solving the equations of radiative transfer with a known source function. ITRAN=0 gives the ordinary Feautrier method, ITRAN=1 gives the Feautrier method with cubic spline accuracy, ITRAN=2 gives the Feautrier method with Hermite accuracy. On a depth scale smooth

in max $\Delta \lg \tau_{mu\mu}$ ITRAN=2 will give a higher accuracy but if the depth scale is very irregular instabilities may occur, see Section 3.3 in the Uppsala report. ITRAN=3 (not recommended) and ITRAN=4 gives an integral method based on fitting the source function with a cubic spline, see routine TRANI.

NMU Gives the number of angle points to be used. Normally it is sufficient with three points in static atmospheres and five in atmospheres with velocity fields.

IWxxxx Printout option. IWxxxx=0 inhibits printout from the specific routine. IWxxxx¿0 normally gives a printout for every IWxxxx depth point. If IDL is available, most printouts should be switched off and the IDL routines used instead (see variables IDL1, IDLNY, IDLCNT and IDLOPC).

- IWABND Abundances printed to file OUT.
- IWATMS Atmospheric parameters printed to file OUT.
- IWATOM Atomic parameters printed to file OUT.
- IWCHAN Graphic convergence pattern printed to file OUT. Seldom of interest and quite large.
- IWDAMP Damping parameters printed to file OUT.
- IWEMAX Maximum corrections to population densities and to mean intensities printed to file OUT and to the standard output (terminal). Very useful.
- IWEQW Equivalent widths printed to file OUT.
- IWEVEC Error vector printed to file OUT. Only of debugging interest and very voluminous.
- IWHEAD Heading lines printed to file OUT. Useful.
- IWHSE Hydrostatic equilibrium printout to file OUT.
- IWLGMX Maximum steps in max $\Delta \lg \tau_{mu\mu}$ printed to file OUT. Very useful in checking depth-scale.
- IWLINE Transition data written to file OUT.
- IWLTE LTE populations written to file OUT.
- IWN Populations and ionization fractions written to file OUT.
- IWNIIT Populations as function of depth and iteration written to file NIIT.
- IWOPAC Opacities written to file OUT.
- IWRAD Radiation quantities written to file OUT.
- IWRATE Rates written to file OUT. Difficult to interpret and lots of data.
- IWSTRT Starting approximation written to file OUT.
- IWTAUQ Monochromatic optical depths written to file OUT. Lots of data.

IWTEST Lots of debug-character data written to file OUT.

IWWMAT Matrix written to file WMAT.

IWJFIX File JFIX2 written. Switched on by IWRAD ¡ 0 in earlier versions. Default value is IWJFIX=0.

IWARN Warning messages can be partly switched off with IWARN. A value of zero switches off all but the most important warning messages, IWARN=1 includes more warning messages but excludes warnings about negative opacities (these warnings are seldom of any consequence and can fill up the JOBLOG file). Default value is IWARN=2.

IOPACL Regulates the inclusion of background opacities from lines. IOPACL=1 sets the source function to the Planck-function for background lines; IOPACL=2 uses a two level formula with the collisional rate calculated with the van Regemorter formula. See section **??**. Default value is IOPACL=0.

ISCAT ISCAT=1 switches on a scattering version of Feautrier solving consistently for the scattering part of the background source function. If there is a lot of scattering in the continuum opacity the standard lambda iteration of the background source function may lead to slow convergence (or even stabilization). ISCAT=1 solves this problem at the expense of more computing time. Default value is ISCAT=0.

INCRAD INCRAD=1 flags the inclusion of an incident radiation field. This radiation field is read from the file IMINUS. See section **??**. Default value is INCRAD=0.

INGACC INGACC=1 flags the application of Ng acceleration to the iteration procedure. Almost necessary in connection with the local operator and often useful also for the global operator. Will sometimes lead to divergence. Default value is INGACC=1.

ICRSW ICRSW=–2 will switch on collisional-radiative switching (see section **??** with values determined automatically.
ICRSW=–1 will prompt reading of values interactively.
ICRSW $>$ 0 will decrease the switching parameter equidistantly in the log with ICRSW steps per decade.
Default value is ICRSW=0.

IOSMET IOSMET=1 switches on background line opacities read from file ABSMET. A two-level epsilon is calculated using the van-Regemorter formula for the collisional rate and setting the source function to (1-epsilon)*J + epsilon*B. The van-Regermorter formula is different for neutrals and ions and the ABSMET file is an opacity sampling file where that information is not available. The approximate proportion of the opacity due to ionized species is given by EOSMET. IOSMET=2 gives line blanketing following **??**. IOSMET=3 is the same as IOSMET=1 but line opacities are set to zero below 1520 Å. See section **??**. Default value is IOSMET=0.

EOSMET gives the approximate fraction of line opacity due to ionized species when IOSMET $\neq 0$. Default value is EOSMET=0.5.

IDL1 Switches on writing of most variables to file IDL1 for later processing with IDL (Interactive Data Language). See section **??**. Default value is IDL1=1.

IDLNY Switches on writing of frequency dependent information to file IDLNY. This file will be very large for large model atoms with many transitions. Default value is IDLNY=0.

IDLCNT Switches on writing of contribution functions to file IDLCNT. Default value is IDLCNT=0.

IDLOPC Switches on writing background opacity contributions to file IDLOPC. Default value is IDLOPC=0.

9 IDL routines to analyze results

After a successful run the output is in the form of IDL files. Which files are written depend on the values of `INPUT` variables:

`IDL1`	switches on writing of the `IDL1` file. Read with IDL procedure `mulrd`
`IDLNY`	switches on writing of the `IDLNY` file. Read with `nyrd`
`IDLCNT`	switches on writing of the `IDLCNT` file. Read with `cntrb`
`IDLOPC`	switches on writing of the `IDLOPC` file. Read with `opcrd`

The run-script `run.csh` moves the output files to `xxxxx.atom_atmos` where `xxxxx` is the lower case version of the standard filename in the table above and `atom` and `atmos` are the names of the atom and atmos files, e.g. `idl1.ca6_val3c`. In IDL you set the atom and atmos names with the procedure `default`:

```
IDL> default,'ca6_val3c'
```

will thus make `mulrd` read the file `idl1.ca6_val3c`. To enable using the MULTI IDL routines you need to put the idl subdirectory in your IDL search path. You also need to define the common block variables in your main session with the command

```
IDL> @common_multi
```

A number of IDL routines are included in the distribution, help text for some is in `idldoc.pdf`.

reading routines:

`default`	set default extension for output files to be read
`mulrd`	read `IDL1` file
`nyrd`	read `IDLNY` and `JNY` files
`cntrb`	read `IDLCNT` file
`opcrd`	read `IDLOPC` file
`nyclose`	close the `IDLNY` and `JNY` files

routines to manipulate atomic data:

`ratom`	read atom file without first running MULTI
`watom`	write atom file
`c_l`	get L quantum number from atomic label
`c_n`	get n quantum number from atomic label
`c_s`	get s quantum number from atomic label
`gacalc`	calculate GA from the transitions in `ATOM`
`atom_renum`	renumber or remove atomic levels
`atom_merge`	merge atomic levels (is not quite working)
`c_colrd`	read collisional data
`c_gencolw`	write collisional data
`remap_photo`	remap photoionization cross-sections to given wavelength-set
`opctab_xl`	define wavelength set for remap_photo
`remove_xl`	called by `remap_photo`

routines to manipulate other input files:

`rabslin`	read `ABSLIN` file
`wabslin`	write `ABSLIN` file
`rabsmet`	read `ABSMET` file
`read_vald`	read VALD file and write `ABSLIN` file
`watmos`	write `ATMOS` file

various calculation and conversion routines:

`convl`	convert vacuum wavelengths to air wavelength
`dlamb`	convert MULTI frequency units to $\Delta\lambda$ in Å
`double`	make profile symetric around x=0
`intflux`	calculate integrated flux or intensity
`iondata`	data for ionization calculation
`saha`	calculate Saha ionization
`net_rates`	calculate net rates
`planck`	calculate the Planck function
`tradb`	calculate radiation temperature
`cacl_sotbfi`	calculate SOT BFI intensities
`calc_sotnfi_na`	calculate SOT NFI intensities for Na-D
`calc_sotnfi_mg`	calculate SOT NFI intensities for Mg-b

plotting routines:

`cool_sum`	add up all cooling contributions
`cool_plot`	plot the cooling function
`opcplot`	plot opacity contributions
`plotcntrb`	plot contribution function
`plot_rates`	plot rates in/out of a level
`plot_spectrum_id`	plot spectrum with `ABSLIN` labels and FTS atlas
`plot2`	plot two quantities in same plot
`scale2`	plot a secondary scale in plot
`spec_ct`	set up a spectrum colour table
`xmovie`	interactively change axis ranges, animate plot

response function routines:

`watmos_dt`	make atmospheric input files for response function calculation
`intt_dt`	make save file with all intensities from response function calculation
`make_resp_sotbfi`	Make SOT BFI response function
`plot_response_sotbfi`	Plot SOT BFI response function

numerical routines:

`intep`	Hermite spline interpolation
`ipol_int`	interpolate preserving integral
`trapez`	perform trapezoidal integration

general utilities:

`cstrip`	remove comment lines from input file
`label_bob`	label linetypes in plot
`label`	label linetypes in plot
`mtimer`	print remaining time for a routine
`reverse`	reverse indices in an array (IDL library routine)
`string3`	convert argument to three character string padded with zeros from left

routines for 3D wrapper (mul23luc_3d.x and mul23guc_3d.x):

`rout3d`	read multi_3d output files
`rn3d`	read n3d output file
`rb3d`	read b3d output file
`watmos_multi3d`	write multi_3d format atmos file
`watmos_3d21d`	write 1D MULTI atmos file from a given column in a 3D dataset

10 IDL variables

```
MOST USED VARIABLES (IDL INDEX CONVENTION)

index convention:
i       level index  [0:nk-1]
j       level index  [0:nk-1]   (j=upper level,  i=lower level)
ih      level index for nh array  [0:5]
kr      transition index  [0:nrad-1] where [0:nline-1] are lines,
                                          [nline:nrad-1] are continua
krc     continuum transition index  [0:nrad-nline-1]  =kr-nline
kfx     fixed transition index  [0:nrfix-1]
k       depth index  [0:ndep-1]
mu      angle index  [0:nmu-1]
nu      frequency index [0:nq[kr]-1] or [0:nq[kr]]
il        wavelength index for background opacities
ic       contribution index for background opacities



after running MULRD

nk              number of levels including continuum levels
nrad            number of radiative transitions treated in detail
nline           number of radiative bound-bound transitions
nrfix           number of transitions with fixed rates
nq[kr]          number of frequencies
nmu             number of angles
ndep            number of depth points

atomid          4 character identification of atom.
abnd            atomic abundance, log scale with hydrogen=12
awgt            atomic weight. input in atomic units, converted to cgs

ev[i]           energy above ground state. input in cm-1, converted to ev
g[i]            statistical weight of level
label[i]        20 character identification of level
ion[i]          ionization stage of level, 1=neutral

jrad[kr]        jrad[kr]-1 is upper level of radiative transition kr
irad[kr]        irad[kr]-1 is lower level of radiative transition kr
krad[i,j]       krad[i,j]=krad[j,i] is the number of the transition i-j
f[kr]           oscillator strength
ga[kr]          radiative damping parameter
gw[kr]          van der waals damping parameter
```

```
gq[kr]          stark damping parameter
alamb[kr]       vacuum wavelength in angstrom
ktrans[kr]      ktrans[kr] is continuum transition nr for transition kr
                 krc=ktrans[kr]

a[kr]           einstein a coefficient
b[i,j]          einstein b coefficient
c[i,j,k]        collisional transition rate

jfx[kfx]        jfx[kfx]-1 is upper level of fixed transition kfx
ifx[kfx]        ifx[kfx]-1 is lower level of fixed transition kfx
ipho[kfx]       =1 continuum, =0 line
a0[kfx]         crossection at limit
trad[kfx]       brightness temperature for continua
itrad[kfx]      radiation temperature option. =1 rad.temp=temp,
                =2 photospheric option, rad.temp=temp out to temp.lt.trad
                   then temp=trad outwards
                =3 chromospheric option, rad.temp=temp except when
                   temp .gt. trad and temp increasing outwards.
                   then rad.temp=trad

dnyd[k]         doppler width in units of a typical doppler width
adamp[k,kr]     voigt damping parameter
\bigskip
n[i,k]          population density in cm-3
nstar[i,k]      lte population density
totn[k]         total population density of atom

alfac[nu,krc]   photoionization crossection
z[k]            n[i,k] - gij*n[j,k]
gij             g[i]/g[j] for bb transitions

qnorm           unit typical doppler width in km per second at line center
hn3c2           h*ny**3/c**2

q[nu,kr]        frequency variable, in units of a typical doppler width
                 positive q for increased frequency
qmax[kr]        maximum frequency, same units as q
q0[kr]          frequency within which quadrature points are distributed
                 linearly instead of logarithmically
ind[kr]         =1 for one sided quadrature (symmetric profile)
                =2 for two sided quadrature (asymmetric profile)
frq[nu,krc]     frequency in hz for continua
                 in nu=0 the frequency for the edge is stored
                 in nu=1:nq[kr] the frequency points
```

```
xmu[mu]          cosine of angle

atmoid           72 character identification of atmosphere used
dpid             72 character identification of depth-scale
dptype           =t  depth scale is tauscale, =m depth scale is mass scale,
                     see  routine  atmos

grav             gravitation acceleration
cmass[k]         column mass
temp[k]          temperature
nne[k]           electron density
vel[k]           macroscopic velocity in doppler units
vturb[k]         microturbulence velocity in doppler units
height[k]        height above tau5000=1 in kilometers

bh[ih,k]         departure coefficient for hydrogen
nh[ih,k]         population density for hydrogen
rho[k]           density

tau[k]           standard optical depth
taulg[k]         alog10(tau)
xnorm[k]         standard opacity in cm**2 per cm**3
sl[k,kr]         line source function
bp[k,kr]         planck function

dlgtmx[k]        maximum delta lg tauq for each depth

rij[k,kr]        radiative rate from i to j per ni atom
rji[k,kr]        radiative rate from j to i per nj atom

flux[nu,kr]      monochromatic physical flux in cgs units
outint[nu,mu,kr] monochromatic surface intensity in cgs units
                  in nu=0 the continuum intensity is stored
                  in nu=1:nq[kr] outint for the nu-points of the transition

cool[k,kr]       cooling function in erg/cm**3/s
weq[kr]          equivalent width
weqlte[kr]       equivalent width in lte

physical constants:

ee               electron charge
em               electron mass
hh               planck constant
```

```
cc              velocity of light
bk              boltzmann constant
uu              universal mass constant
hce             hh*cc/ee*1.e8    lambda(angstrom)=hce/energy(ev)
hc2             2*hh*cc*1.e24    2*h*ny**3/c**2=hc2/lambda(angstrom)**3
hck             hh*cc/bk*1.e8    h*ny/kt=hck/lambda(angstrom)/t
ek              ee/bk
pi              pi

after running NYRD,kr,nu

tauq[k]         monochromatic optical depth
s[k]            monochromatic source function
sbck[k]         background source function, including scattering term
xcont[k]        background opacity relative to standard opacity
x[k]            total opacity relative to standard opacity
iplus[k]        i+
iminus[k]       i-
jny[k]          mean intensity

p[k]            feautrier mean intensity
pms[k]          p - s
rny[k]          xcont/x
scat[k]         scattering relative to background opacity
sc[k]           background source function, excluding scattering term



after running CNTRB:

cntrbi[k,nu,kr]  contribution function to intensity on a alog10(tau) scale
cntrbr[k,nu,kr]  contribution function to relative absorption
cntrbf[k,nu,kr]  contribution function to flux
xmeani[nu,kr]    average height of formation for intensity
xmeanr[nu,kr]    average height of formation for relative absorption
xmeanf[nu,kr]    average height of formation for flux


after running OPCRD:

xla[il]         wavelength for background opacity contributions
chi[k,il]       total opacity, depth k, wavelength il
prov[ic,k,il]   opacity from contributor ic, depth k, wavelength il
sumabs[k,il]    absorption
```

```
sumsca[k,il]    scattering
provid[ic]      IDL id for contributor ic
provid0[ic]     ASCII id for contributor ic
kr_xla[il]      transition number (kr) for wavelength il
ny_xla[il]      frequency number for wavelength il
```

3D variables:

variable	file	explanation
id3d	out3d	80 character ID string
nq[kr]	out3d	number of frequencies for each transition
xl[il]	out3d	wavelengths
taulg3d[ix,iy,iz]	taulg3d	lg(tau_5000)
cmasslg3d[ix,iy,iz]	cmass3d	lg(cmass)
Iv3d[il,ix,iy,iz]	Iv3d	emergent intensity for wavelength il
b3d[ix,iy,iz,i]	b3d	departure coefficient
n3d[ix,iy,iz,i]	n3d	population density
height3d[ix,iy,iz]	height3d	height-scale
x3d[ix,iy,iz]	x3d	monochromatic opacity for chosen line and frequency
xnorm3d[ix,iy,iz]	xnorm3d	opacity at 5000 A