MULTI exercises

February 2015

Contents

1 Introduction 3

2 Installing MULTI 3

3 Test runs - getting to know variables 4

4 Formation of Na D 7

5 Comparing with LTE 8

6 Optimising the depth-scale, Mg-b lines 9

7 Hinode SOT BFI intensities 11

8 Response functions 12

9 Terms vs detailed levels in atomic model, Si 10827 Å line 14

10 non-LTE effects on abundance determinations 15

11 Building and manipulating model atoms 16

12 Running 3D cases column by column 18

1 Introduction

The following exercises use the non-LTE code MULTI. Version 1.0 of the code is documented in the Uppsala Astronomical Observatory report 33 (including a complete source code listing). Changes from version 1.0 to version 2.3 (the one you will be using) are documented in `mul23.pdf` including a description of all switches in the input file `INPUT`. A quick start guide is in `multi_manual.pdf`.

IDL procedures will be used to study the output. Documentation of some of them is in `idldoc.pdf`. This is work in progress and some routines are only documented in the source code itself.

Documentation on variable names in the IDL output files and how they are indexed can be found in `multi_manual.pdf`.

All the documentation files and the MULTI version 2.3 distribution is available at
http://folk.uio.no/matsc/mul23

2 Installing MULTI

MULTI is a Fortran-77 code to solve non-LTE line formation problems when the atmospheric temperature structure is given (there is thus no energy equation). The code will be used in the following to study non-LTE line formation.

The first step is to install MULTI, see `multi_manual.pdf` for instructions:
How to get distribution
Making the executables
You should make the executables `mul23gus.x` and `mul23lus.x` and make links to the `run` subdirectory. Also make sure you have downloaded the file `ftsatlas.idlsave` and put it in the `idl` subdirectory.

3 Test runs - getting to know variables

Now you can start making a few runs. Try first with some of the setups included in the official distribution: 6-level models of hydrogen and singly ionized calcium and the VAL3C model atmosphere (takes about a second CPU time). Go to the `run` subdirectory and issue the command (if your current directory is not in your search path you may need to write `./run.csh`):

```
./run.csh ca6 val3c 23gus
```

You should always check the run by inspecting the `JOBLOG` file. In this case it should look something like:

```
FOLLOWING BACKGROUND ABSORBERS EXCLUDED:

   KR  LOWER LEVEL           UPPER LEVEL
    6  CA II 3P6 4S 2SE      CA III GROUND TERM
    7  CA II 3P6 3D 2DE 3/2  CA III GROUND TERM
    8  CA II 3P6 3D 2DE 5/2  CA III GROUND TERM
    9  CA II 3P6 4P 2PO 1/2  CA III GROUND TERM
   10  CA II 3P6 4P 2PO 3/2  CA III GROUND TERM
 NORMAL END
```

This means that for each bound-free continuum in the model atom, the corresponding background opacity from the `ABSDAT` file has been excluded. This is as it should be. The `NORMAL END` signifies convergence.

Now we can try the hydrogen run:

```
./run.csh h6  val3c 23gus
```

From the `EMAX` printouts on the screen it is clear that the run does not converge and MULTI gives up when `ITMAX` iterations have been performed. In the `JOBLOG` file there are messages about negative opacities and at the end the text `NO CONVERGENCE`.

We can then try the local operator version. The global operator version gives faster convergence for small model atoms but has a smaller convergence radius (quite evident in this example). Try the local operator version by replacing `23gus` above with `23lus`. There is now convergence. The `JOBLOG` file reports negative opacities but that is during the iteration sequence and not for the final solution.

The results are most easily inspected using the accompanying IDL programs. Add `mul23/idl` to your idl path by including the following line in the file `.cshrc` in your home directory (or similar to the `.bashrc` file):

```
setenv IDL_PATH "+~/mul23/idl:+$IDL_PATH"
```

Start IDL, make MULTI variables available, read in variables and try a few plots

```
IDL> @common_multi
IDL> default,'h6_val3c'
IDL> mulrd
IDL> plot,height,temp
IDL> plot,height,n[nk-1,*]/totn
IDL> print,alamb
IDL> double,4,outint,xx,yy
IDL> plot,xx,yy
```

These commands will plot the temperature as function of height in the atmospheric model, the ionization fraction as function of height, print the wavelengths (in vacuum) of the transitions (center of line for bound-bound transitions and photoionization edge for bound-free), and plot the intensity as function of wavelength from line-center for the H-α line. It is recommended to go through the list of variables in `multi_manual.pdf` and print/plot combinations to get familiar with the variables available in the `IDL1` file.

You may also use the xmovie widget procedure. This procedure is really meant for animation of time-dependent variables (two dimensional y and possibly x-arrays) but can also be used to get interactive control of axis ranges. The only sliders that make sense then are the xmin, xmax, ymin and ymax sliders.

```
IDL> xmovie,height,temp
```

You may restrict the range to begin with and add symbols

```
IDL> xmovie,height,temp,yrange=[0,20000],/psym
```

There are several output files and you may have to switch on writing of such a file in order to have access to the variables mentioned (IWOPAC, IDL1, IDLNY and IDLCNT in the `mul23/input/input.h6` and `mul23/input/input.ca6` files). The distribution `input.ca6` has all switched on. You can check background opacities in the idlopc file with

```
IDL> default,'ca6_val3c'
IDL> mulrd
IDL> opcrd
% Compiled module: OPCRD.
IDL> opcplot,taulg,0
```

This will give the opacity contributions at 5000 Å. All wavelengths for which there are background opacity data are in the array `xla`. Just change the index in the `opcplot` command to get the opacity contributions at another wavelength.

The file `IDLNY` (renamed `idlny.ca6_val3c` by the `run.csh` script) contains variables that are functions of wavelengths that are not stored in memory. Example usage:

```
IDL> kr=0
IDL> print,convl(alamb[kr])
IDL> dum=min(abs(q[0:nq[kr]-1,kr]),nu0)
IDL> nyrd,kr,nu0,mdep=150,/jnydp
IDL> plot,height,alog10(tauq)
IDL> plot,alog10(tauq),s,/ylog
IDL> oplot,alog10(tauq),iplus,linestyle=2
IDL> oplot,alog10(tauq),iminus,linestyle=1
IDL> oplot,alog10(tauq),jny,linestyle=3
```

The first command stores transition index 0 in variable `kr`, the second prints the air wavelength of the transition (without `convl` you would get the vacuum wavelength) which shows that transition 0 is the Ca II H-line, the next command finds the frequency index (`nu0`) for the line-center. This is recommended because for atmospheres without velocity fields all profiles will be symmetric and MULTI will only calculate half the line and have line-center in frequency index 0 but if there are velocity fields a two-sided quadrature is used and the frequency index for the line center will be the middle index. With the method above you are sure to get the right frequency index for the line-center. The next command reads the data for this transition and frequency (`mdep=150` gives the value of MDEP in the PARAM file when making the executable) and we can then plot the line center monochromatic optical depth (`tauq`) as function of height and the total source function (`s`) and outgoing and incoming intensity. These are also angle dependent but it is only the most vertical ray that is stored (and the μ value for that ray is in `xmu[nmu-1]`). Finally we plot the mean intensity.

The file IDLCNT (renamed `idlcnt.ca6_val3c` by the `run.csh` script) contains contribution functions on a lg(tau) scale. Read the file and plot contribution functions to intensity for all frequency points of the Ca II K-line with

```
IDL> cntrb
IDL> plotcntrb,1,taulg,cntrbi
```

Plot contribution functions to relative absorption for all frequency points with

```
IDL> cntrb
IDL> plotcntrb,1,taulg,cntrbr
```

Plot the contribution function to relative absorption at line center for the Ca II K-line together with the temperature structure with

```
IDL> plot2,taulg,cntrbr[*,0,0],temp*1.e-3,y1title='C!dR!n(lg(!4s!3!d500!n))',$
IDL>   y2title='T [kK]',xtitle='lg(!4s!3!d500!n)',xmargin=[8,8]
```

4 Formation of Na D

Use data files for Na I. The atomic data file is called `atom.na12`. Before you can make a run with this model atom you may need to modify the parameter statements in the program to accommodate the larger model atom needed here (see `multi_manual.pdf` for a more complete description of dimensioning variables). The parameters are set in files `PARAM` and `PARAMW`. Check that `MK` in `PARAM` and `MK1` in `PARAMW` are at least as large as `NK` in `atom.na12` (`NK=12` in `atom.na12` explaining the naming convention used for files the atomic data files), that `MLINE` in `PARAM` is at least as large as `NLINE` in `atom.na12`, `MWIDE` in `PARAM` at least as large as `NCONT` in `atom.na12`, `MRFIX` at least as large as `NRFIX` and `MQ` at least as large as the largest `NQ` in `atom.na12`. (You may hope that MULTI is smart enough to tell you which dimension has to be changed and try running the version you have already compiled to get a message into file `JOBLOG` of what dimension has to be changed but that could potentially be dangerous). If needed, edit the file `PARAM` and `PARAMW` and recompile the code:

```
cd ~/mul23/source_dp
make mul23gus.x
make mul23lus.x
```

You may then run the na12 case:

```
cd ~/mul23/run
./run.csh na12 val3c 23gus
```

Inspecting the `JOBLOG` file shows there are negative opacities also in the converged solution. In IDL we can see that this is a very long wavelength line and this may then happen and actually be physical (maser action). In any case this is in the very top of the atmosphere and we get convergence. To make sure this line does not affect the D-lines one can make a new atom with this line excluded and make sure the results do not change. We skip this step here.

Compare the source functions of the two D-lines. Are they equal? Why/why not? Compare with the Planck function. Where is the monochromatic optical depth unity? From the Eddington-Barbier relation, do you expect an emission core or not? Is there an emission core? Compare with the LTE intensity profile. Compare with the observed profile. Is the two level approximation good?

Hint: Some useful commands are

```
IDL> default,'na12_val3c'
IDL> mulrd
IDL> print,convl(alamb)
IDL> kr=0
IDL> dum=min(abs(q[0:nq[kr]-1,kr]),nu0)
IDL> nyrd,kr,nu0,mdep=150,/jnydp
IDL> plot,taulg,tradb(sl[*,kr],alamb[kr])
IDL> kr=1
IDL> oplot,taulg,tradb(sl[*,kr],alamb[kr]),line=2
IDL> oplot,taulg,temp,line=3
IDL> scale2,taulg,alog10(tauq),0.2
IDL> double,kr,outint,xx,yy
IDL> plot,xx,yy,xran=[-2,2]
```

5 Comparing with LTE

When we have a completed run we often want to compare with LTE. Some LTE values are included in the output files (like the LTE population densities in `nstar` and the Planck function in `bp`) but we don't have the line-profile. The easiest way of doing that is to use MULTI again. To get the output saved in a different set of files we copy the atom file to one with _LTE appended to the name, we copy the `DSCALE` and `INPUT` files. For the files we are not to change we may make a link instead of copying (but the `INPUT` file we will modify):

```
cd ../input
ln -s atom.na12 atom.na12_LTE
ln -s dscale.na12_val3c dscale.na12_LTE_val3c
cp input.na12 input.na12_LTE
```

In the `input.na12_LTE` file we set `ICONV=1, ISTART=1, ILAMBD=0, ITMAX=0`. This will set the starting approximation to LTE, do no lambda iterations or iterations and write out results even though we have not met the convergence criterion.

```
cd ../run
./run.csh na12_LTE val3c 23gus
```

will now give us output files with the LTE case. Compare the D-profiles. Are there central reversals now?

6 Optimising the depth-scale, Mg-b lines

The depth-scale used for the calculations is given in the file `DSCALE`. It is important that the depth-quadrature resolves gradients in the atmosphere and that there are enough points per decade in optical depth for the important transitions in the model atom under study. The optimal depth-scale is therefore dependent on both the model atmosphere and the model atom. For several atoms and atmospheres there exist `DSCALE` files in the distribution but not for all. In this exercise we will use MULTI to construct a good `DSCALE` file.

We start by running MULTI with the `atom.mgb13` atomic file and the FALC model atmosphere. There is no `dscale.mgb13_falc` file in the `input` subdirectory but there is a generic `dscale.falc` that will be chosen instead. This depth-scale includes the whole range of the FALC atmosphere with points equidistant in lg(cmass) (indicated with a negative number of depth-points and the top and bottom point given):

```
./run.csh mgb13 falc 23gus
```

There is some new printout on the screen. This comes from the fact that we have switched on opacities from line-blanketing (`IOSMET=1` in the `INPUT` file). To save time we have set `ELIM2=0.01`. In the final run with the optimized depth-scale we can set that to 0.001 which is quite a standard value. In the `JOBLOG` file there are some warning messages about `DLGTMX` being large.

In the file `out.mgb13_falc` there is information written on the properties of the depth-scale (through `IWLGMX=1` in `input.mgb13`). This information comes after the `EMAX` printout and has the beginning and end:

```
1 D LG TAUNY IS MAXIMUM LG(TAUNY(K))-LG(TAUNY(K-1))
   K   LG DEPTH  D LG TAUNY  MAX TAUNY  MIN TAUNY
   1  -5.170330   0.000000  3.163E-08  1.049E-08
   2  -5.130025   5.397582  2.621E-03  1.552E-07
   3  -5.089719   0.617897  9.112E-03  2.908E-07
   4  -5.049414   0.359678  1.750E-02  4.142E-07
   5  -5.009109   0.539064  2.703E-02  5.257E-07
..
 147   0.714250   0.383054  1.247E+05  2.218E+00
 148   0.754555   0.419984  1.375E+05  4.121E+00
 149   0.794861   0.457600  1.503E+05  7.839E+00
 150   0.835166   0.452531  1.615E+05  1.513E+01
```

The `MAX TAUNY` column shows the maximum monochromatic optical depth over all transitions, frequencies and angles. This should be small (order `1.E-04` at the top. From this example we see that our depth-scale extends higher than is really necessary for our model atom. We will only use a few points too much so that is OK. At the bottom we inspect the last column which shows the minimum monochromatic optical depth over all transitions, frequencies and angles. This should be more than one, otherwise we see through the atmosphere and the bottom boundary condition (which assumes a semi-infinite atmosphere) would need to be changed. The third item of interest is the second column. It shows the maximum step in lg(tau) over all transitions, frequencies and angles. This should be as even as possible and preferably on the order of 0.2 and certainly not bigger than one. There is a mechanism in MULTI that tries to achieve this: MULTI writes out a more optimal depth-scale to the file `DSCAL2`. We can use this file in a next step:

```
\cp DSCAL2 ../input/dscale.mgb13_falc
./run.csh mgb13 falc 23gus
```

Inspection of the `out.mgb13_falc` file shows that we have achieved a depth-scale that has a rather constant `D LG TAUNY` of 0.16-0.17 but still with a big jump close to the top. Repeating this procedure a few times will give a smooth depth-scale.

You may look at the three Mg-b lines calculated here using the procedure `plot_spectrum_id`, see also the following exercises:

```
IDL> @common_multi
IDL> default,'mgb13_falc'
IDL> plot_spectrum_id,kr=1,/fts
```

There is a good match between the calculated spectrum and the FTS atlas. You may switch off line-blanketing (setting `IOSMET=0` in the `input.mgb13` file to see the effect of line-blanketing. As with the LTE case above the best way to do that is to link/copy the input files to files with a new extension so the output is stored with different names:

```
cd ../input
sed -e 's/IOSMET=1/IOSMET=0/' input.mgb13 > input.mgb13_noblank
ln -s dscale.mgb13 dscale.mgb13_noblank
ln -s atom.mgb13 atom.mgb13_noblank
cd ../run
./run.csh mgb13_noblank falc 23gus
```

The `sed` command is an editor command that will make the required change in the input file and store the result with the new name. You may of course instead copy the input file to the new name and use a regular editor to make the change.

You can now compare the two runs and see that without line-blanketing there is too little neutral magnesium (non-LTE overionization) and the lines get formed too low where the source function is higher leading to too weak lines. It is also instructive to make an LTE run similar to the Na case and compare. The value of `IOSMET` will not matter but it is best to set it to zero to avoid the timeconsuming calculation of line opacities that will only be used to calculate continuum intensities that we are not interested in. Note especially the line-cores in LTE.

7 Hinode SOT BFI intensities

In order to calculate SOT BFI intensities we need to solve the NLTE equations for Ca II and include overlapping lines within the SOT BFI filters. Included in the distribution there are files to accomplish this with the exception of the CN filter where the appropriate linelist has not been produced yet. The calculation is made in two steps: first the Ca II problem is solved and then these populations are used to perform the formal solution with background opacity from lines added (using `IOPACL=1` in `INPUT`). You may need to change the dimensioning in `PARAM` and `PARAMW`.

```
cd ~/mul23/run
./run.csh sotbfi_init falc 23gus
\cp rstrt.sotbfi_init_falc RSTRT
./run.csh sotbfi falc 23gus
```

Note that the last run takes some time because of the large number of frequency points. Since the old populations are used there is only a formal solution performed and there will be only one line with EMAX printout.

We can look at the calculated intensities and compare with the FTS atlas:

```
IDL> default,'sotbfi_falc'
IDL> mulrd
IDL> kr=0
IDL> plot_spectrum_id,kr=kr,/fts
```

The continuum level is not given in the FTS atlas and `plot_spectrum_id` just takes as default the highest intensity in the wavelength band and in the case of a strong line where you don't go far enough into the wing to get continuum intensity you get the wrong normalization. You may specify the value of the continuum with keyword `icont`. Note that `icont` should be given in cgs units and the plot is in SI units.

```
IDL> plot_spectrum_id,kr=kr,/fts,icont=2.9e-5
```

There are many keywords that regulate the behaviour of `plot_spectrum_id`. You may change the number of lines identified with the `limit` keyword. A larger value will give only the stronger lines. The length of the ID lines in the plot is proportional to the LTE population of the lower level of the transition.

```
IDL> plot_spectrum_id,kr=kr,/fts,icont=2.9e-5,limit=1.e-3
```

We may plot the FTS atlas with an overplot of the Hinode BFI filter transmission function:

```
IDL> restore,'../idl/bfi.idlsave'
IDL> text=''
IDL> for kr=0,5 do begin xl=wl[0:nl[kr]-1,kr] &$
IDL>  plot_spectrum_id,/fts,xl=xl,int=tl[0:nl[kr]-1,kr],$
IDL>  atmos='Hinode BFI filter',limit=5e-4,/air & read,text
```

You may of course combine the two: plot MULTI and FTS intensities and overplot the Hinode BFI filter transmission as well.

8 Response functions

We are often interested in deducing the bulk velocity from the Doppler-shift of a line or the temperature change from the change of intensity. In these cases we should use the *response function* instead of a contribution function. The response function gives how a given observed quantity changes (*e.g.* Hinode BFI intensity) when we change a physical parameter in the atmosphere (*e.g.* temperature) as a function of depth. There are thus three defining quantities in a response function. From this definition we get:

$$\frac{\Delta I}{I} = \int_{-\infty}^{\infty} R_{\Delta I,T}(z)\frac{\Delta T}{T}(z)dz \tag{1}$$

where $\Delta I/I$ is the relative change in Hinode BFI intensity and $\Delta T/T$ is the relative change in temperature.

Numerically we may determine the response function by introducing a perturbation into the atmosphere and calculate the response.

Use the procedure `watmos_dt` to generate a series of atmospheres that have a temperature perturbation of 1% from the bottom of the atmosphere to a given height (you need to go through the previous exercise first).

```
IDL> default,'sotbfi_falc'
IDL> mulrd
IDL> watmos_dt,'atmos.falc'
```

This creates 151 atmospheres with a temperature perturbation of 1% from the bottom and up to a given depth-point. Since we are going to integrate the intensity over the Hinode SOT BFI transmission function we don't need as many frequency points as in the previous exercise and we will use `atom_sotbfi_nq384` with 384 frequency points instead of the 1536 points in `atom_sotbfi`. We use a special script to run all 151 runs:

```
./run_sotbfi_all.csh
```

This script first makes one run similar to the one in the previous exercise to set up the proper `DSCALE, ABUND, ABSDAT` etc input files. It uses the population densities from this run as starting solution for a non-LTE run with a temperature perturbation in the last point, solves the formal solution for $\mu = 1$ with background line opacities and loops through all 151 atmospheres. This will take some time (on a Macbook Pro 7s per atmosphere for a total of 18 minutes). We now have 151 `idl1` files with the information we need to calculate response functions. We collect the important information with the routine `intt_dt` and calculate the response functions with `make_resp_sotbfi` and plot the result with `plot_response_sotbfi`

```
IDL> intt_dt,'sotbfi_nq384_falc'
IDL> make_resp_sotbfi,'sotbfi_nq384_falc'
IDL> plot_response_sotbfi,'sotbfi_nq384_falc',atmosid='FALC'
```

You should get something like Fig. 1.

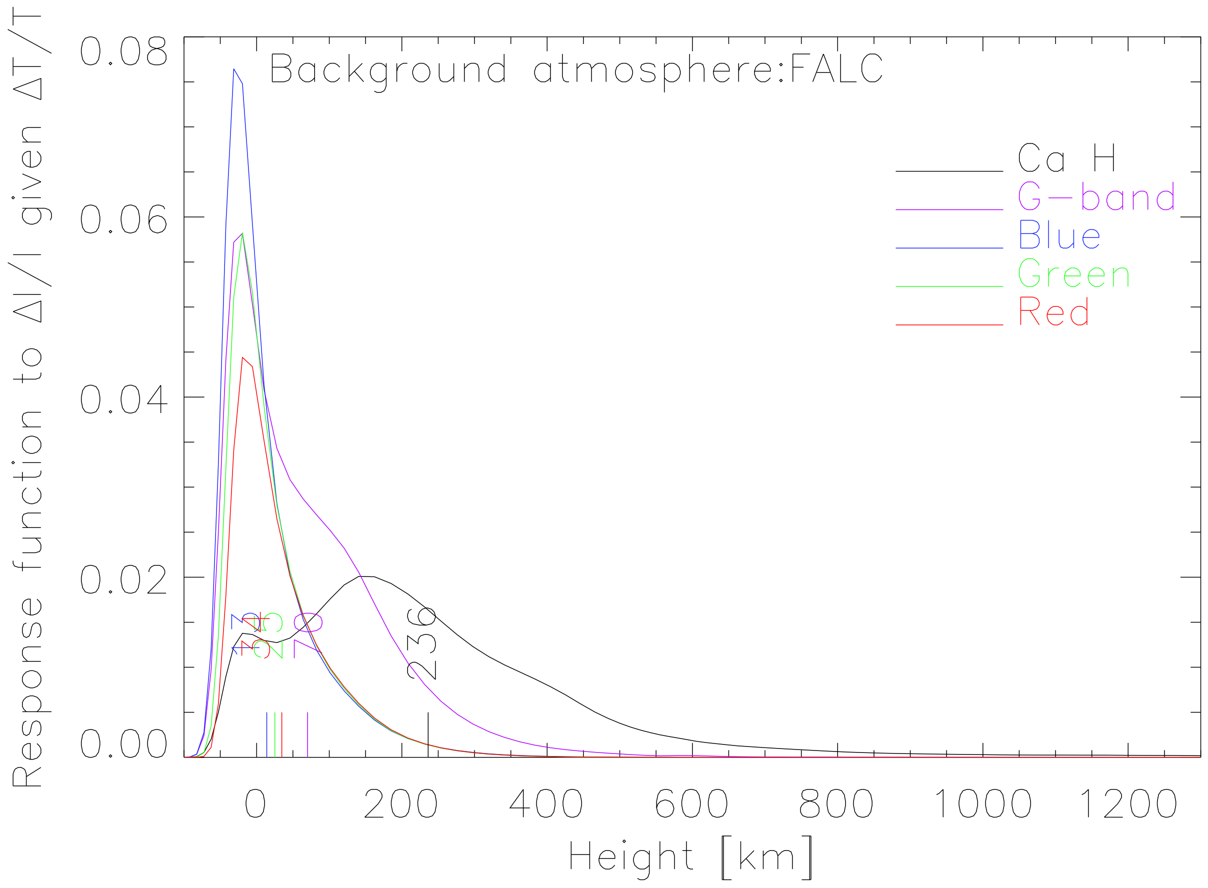

Figure 1: Response functions for Hinode SOT BFI filters and the FALC background atmosphere

9 Terms vs detailed levels in atomic model, Si 10827 Å line

We can often use terms as energy levels in atomic models instead of splitting them up in the individual levels of doublets, triplets etc. The representative transitions will then not have the right opacity or wavelength and can not be compared with any observed spectral lines. There is a mechanism in MULTI that takes care of both aspects and we will illustrate that with calculating the Si I 10827 Å line (that is of great interest because it can be observed together with the He I 10830 Å line). We first use a model atom that only has terms and even some levels being merged terms according to the procedure in Bard & Carlsson, ApJ 682,1376 (2008). We need to get the rates right so we run with `NMU=5` to get a good angle quadrature for the calculation of J_ν. We need to get the ionization balance right so we run with opacity sampling for background opacity (`IOSMET=1`).

```
./run.csh si23_init falc 23gus
```

We use these populations in a restart with no iterations (`ISTART=-1, ILAMBD=0, ITMAX=0, ICONV=1`), with a linelist for background lines (`IOPACL=1`) and only calculating for vertical rays (`NMU=1`). We use an additional atomic data file, `atom2.si23` where the individual lines and levels are given. This file does not have to be complete but you need to have at least the upper an lower levels of the transitions of interest and a continuum level (in order to calculate the broadening that depends on the ionization potential). The populations in the original atom (`atom.si23`) are redistributed over levels in `atom2.si23` that have the same label (NB! you can not specify the J value of the individual levels since there has to be an exact match between the label in `ATOM` and `ATOM2` for this to work). Reading of the file `ATOM2` (linked to `atom2.si23` by the script) is switched on by `IATOM2=1` in the `INPUT` file (in this case `input.si23`).

```
mv rstrt.si23_init_falc RSTRT
./run.csh si23 falc 23gus
```

We can inspect the results:

```
IDL> @common_multi
IDL> default,'si23_falc'
IDL> mulrd
IDL> plot_spectrum_id,kr=0,/fts
```

You may improve the fit by adding macroturbulent broadening to the line:

```
IDL> plot_spectrum_id,kr=0,/fts,vmac=2
```

Note that for macroturbulent broadening to work you need to have equidistant wavelength grid in the atomic data file. This is achieved by setting `Q0` and `QMAX` equal. Note also that the linelist in `abslin.si23` is not very good (not unusual at near-IR wavelengths).

10 non-LTE effects on abundance determinations

The abundance of elements is a crucial parameter for the understanding of the origin of the elements. The tool used is the spectroscopic analysis of spectral lines. Normally one assumes local thermodynamic equilibrium (LTE) in such analysis. This may be dangerous as deviations from LTE may be substantial.

1. Choose one element. Possible elements with input files in the distribution are Li, Mg, Si, Na and O (for these we have good model atoms - construction of a model atom from scratch is too laborious for this exercise). Choose lines that are of interest. The lines should be in the optical part of the spectrum and reasonably strong but pure absorption lines with no emission peaks.

2. Modify the existing model atom to make sure the lines of interest are well treated (modify `NQ, Q0` and `QMAX` in the atom file to get the line profile well covered, use `atmos.falc` for this exercise and copy the `dscale.ca6_falc` file into `dscale.ext_falc`).

3. Choose atmospheric models for the exercise. An overview of existing model atmospheres:

```
ls ../input/atmos.t*
```

one example is `atmos.t6000g4.00m0.00x1.0`

This is a model atmosphere with an effective temperature of 6000K, lg(g)=4.00, [Fe/H]=0.00, Vmic=1.0 km/s

There are corresponding dscale files.

4. Check the formation of the chosen lines. Compare non-LTE with LTE. Find explanations of the non-LTE effects if they are important and find out how they depend on the atmospheric parameters (Teff, lgg, [Fe/H])

11 Building and manipulating model atoms

The realism of the non-LTE calculation is critically dependent on the model atom. We need to include all energy levels that have an influence on the formation of the spectral line we are interested in. There are no fixed rules on how many levels you have to include — to find this out is part of the process. As a rule of thumb you need to include all levels up to the upper level of the transition of interest and you need to include the important ionization and recombination channels. If the line is in the majority ionization stage it is less important to have an accurate ionization balance (since practically all of the atoms are then in the ionization stage of interest). If the line is in a minority ionization stage it is crucial to get the ionization balance right which also means treating line-blanketing. Since computers are fast it is possible to start with a large model atom with several hundred energy levels and use non-LTE calculations to find out which levels can be merged or removed; see Bard & Carlsson, ApJ 682,1376 (2008) for a description of methods to construct a tractable atomic model from a large one.

To get the large, comprehensive model atom to start from you need to access atomic data from several sources. A good starting point is the opacity project data-base (topbase) where you find good quality theoretical photoionization cross-sections but also oscillator strengths for allowed transitions and energy levels. Another source for oscillator strengths is the NIST database. Collisional data are more scarce and one is often forced to use rather crude approximations - van Regermorter's approximation for allowed transitions in ions and Seaton's impact approximation for neutrals. For more highly ionized species the Chianti database is a good source. All these sources have been combined by Phil Judge, HAO, into one database with an IDL interface and possibility to write out in almost MULTI format. This package is called HAOS-DIPER and is available for free download from:

`http://www.hao.ucar.edu/modeling/haos-diper/`

Note that there are some IDL routines with the same names as in the MULTI package so when running HAOS-DIPER you should make sure that HAOS-DIPER routines come first in the search path.

You can manipulate model atoms with HAOS-DIPER, see the manual. When you have the model atom you want you can write the result to a MULTI format atomic file but there are a few steps you need to do first. You need to convert the collisional rate format to one that is supported by MULTI and you need to fill in quadrature information for the lines. This is done with something like the following commands (makes an atomic model for Si 1:

```
IDL> diper,regime=1
IDL> @cdiper
IDL> diprd,'si',1
IDL> colcnv,/ohm
IDL> iw=where(trn.nq eq 0)
IDL> trn[iw].nq=51
IDL> trn[iw].qmax=50
IDL> trn[iw].q0=2
IDL> trn[iw].gw=1.0
IDL> atomwr,'atom.si1'
```

You then need to add the radiative damping using the MULTI IDL routines `ratom, gacalc, watom`.

There are also a few routines in the MULTI distribution that can come in handy.

`remap_photo` will remap photoionization cross-section data to fewer frequency points using criteria of accuracy in the process.

`atom_renum` will renumber energy levels and all transitions involved. This can come in handy when merging models for several ionization stages from separate atomic data files. You may also remove energy levels.

`atom_merge` will merge two energy levels and all transitions involved. This routine makes assumptions on the collisional data present and does not work for all MULTI model atoms.

To get ABSLIN files with line lists for background opacity the best source is the Vienna Atomic Line Database (VALD) available at

```
http://vald.astro.univie.ac.at/~vald/php/vald.php
```

You request all lines in a spectral region, typically 20 Ångström wide, with a request of the type (see the webpage for an explanation on how to send requests):

```
begin request
extract all
long format
3940, 3960
end request
```

This request will give you a linelist for the region 3940-3960 Å in VALD format as an email. This is saved to a file (e.g. vald_3950.data) and converted to MULTI format file (e.g. abslin.3940-3960) with:

```
IDL> read_vald,'vald_3950.data','abslin.3940-3960'
```

You can manipulate the ABSLIN files with the IDL procedures rabslin (to read an ABSLIN file into a structure) and wabslin (to write an ABSLIN file from a structure). To concatenate two ABSLIN files abslin.3940-3960 and abslin.3960-3980 you would issue the IDL commands:

```
IDL> rabslin,lines1,'abslin.3940-3960'
IDL> rabslin,lines2,'abslin.3960-3980'
IDL> lines=[lines1,lines2]
IDL> wabslin,lines,'abslin.3940-3980'
```

12 Running 3D cases column by column

There is a Fortran-90 wrapper that makes it possible to go through a 3D model atmosphere column-by-column and perform a 1D MULTI calculation without producing input files for each column (and they may be thousands). There is one extra input file by the name `INPUT_3D` and the atmospheric model is in a different format in binary form. We first produce the executable in the `source_dp` directory. There is a global operator version and a local operator version and you have a serial version (ending with `.sx`) and an MPI parallelized version. To use the parallel version you need to specify the compiler and linker and possibly some library. For the parallel version you also need to remove any object file that was produced for any serial version first with the `make clean` command. The four makefile commands to produce these four versions are:

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

The atmospheric data file is Fortran unformatted sequential access. The example one (`atmos3d.t246.6`) is with record count in 4 bytes. This is machine dependent and some machine/compiler combinations default to 8 byte record count. You then have to give a compiler switch to force 4-byte record count. On `sagami` with `gfortran` the switch to include in the `FOPT` environment variable is `-frecord-marker=4`. Note also that you DO NOT want the run to finish if there is a floating-point exception - you want it to continue with the other columns. You should therefore not include options that cause crash on floating point exception (like `-fpe0` for the ifort compiler).

To minimize the output you should set `IWEMAX=0` in the `INPUT` file. You should copy (or link) the `atmos3d.t246.6` file into the `run` subdirectory and then you can use the script `run3d.csh` for the serial version or `run3dmpi.csh` for the parallel version:

```
run3d.csh ca6 23guc
```

or

```
run3dmpi.csh ca6 23guc 8
```

where the last parameter is the number of cores for the parallel run.

This test-run will only calculate the calcium solution for a small part of the 3D atmospheric model. This is set in the input file `input_3d.ca6` in the input directory. This file looks like this:

```
0   3          icont_opt (=-1 means fill in out3d, otherwise 0), iredo_max (=3 is normal with crsw
'dtrho'        dscopt (orig, dtrho or dscal2)
1 235 20       ix0,ix1,ixstep
1 235 20       iy0,iy1,iystep
0 51           krsave, nusave if(krsave.lt.0) you give nrad_write and kr for transitions you want
  1            nruns
't246.6'
```

The first line consists of two variable values: `icont_opt` should be zero the first time you run with a particular atom and atmosphere. Some of the columns may not converge and then you can run with a different starting approximation specified in the `INPUT` file and only redo the non-converged columns. You then set `icont_opt=-1`. `iredo_max` is used with collisional radiative switching (`ICRSW>0` in `INPUT`). If collisional radiative switching fails, the code will retry with a larger value of `ICRSW` and do this `iredo_max` times before quitting.

`dscopt` regulates how the depth-scale is set (`DSCALE` is not used).
`dscopt='orig'` gives the original height-scale given in the `atmos3d` file

`dscopt='dtrho'` makes an optimal height-scale for each indivitual column resolving gradients in temperature and density.
`dscopt='dscal2'` will read in a previously calculated `dscal2` file

`ix0, ix1, ixstep` gives the starting column in x, the ending column in x and the step in x.
`iy0, iy1, iystep` gives the starting column in y, the ending column in y and the step in y.

`krsave, nusave` gives the line number and nu-index for which to save the monochromatic opacity into the `x3d` file.

Thereafter follows the number of atmospheric files to calculate for (in case of a time-series of 3D models it is practical to be able to run all snapshots in one multi_3d session) and the extension for their `atmos3d` files.

When running you get information messages to the screen from the reading of the atmosphere, for each column that is started and whether it converged or diverged, a summary of the number of converged and diverged columns and from the writing of output files.

As output you get a number of output files with heading information in `out3d.ext` where `ext` is the extension (in this case `t246.6`). The variables and the output file they are in are described in the manual in the section `IDL variables`. You read most files with the IDL procedure `rout3d`. You need to explicitly compile this routine to get access to the common-block variables.

```
IDL> .r rout3d
IDL> rout3d,'out3d.t246.6'
IDL> rout3d,'Iv3d.t246.6'
```

After these two commands you have access to the intensities in the variable `iv3d[il,ix,iy]` where `il` is the wavelength index and `ix,iy` are the x and y indices. The number of wavelengths in each line is in the `nq` array and the wavelengths in the `xl` array. Note that there is one extra wavelength added (with the value of zero in the `xl` array) before each set of wavelengths for a line. In the given example we have 5 lines and 5 bound-free continua:

```
IDL> print,nq
      101      101      101      101      101       25       32       32       33       33
IDL> print,total(nq)
      660.000
```

The five lines have 101 wavelength points each and the five continua between 25 and 33. In total 660 wavelengths but the `xl` array is 670 elements because of the extra points added (they contain the continuum intensity in the same way as frequency index 0 in the `outint` variable in MULTI). The first line's (Ca II H line) intensities are thus in `iv3d[1:101,*,*]`, the second line (Ca II K line) in `iv3d[103:203,*,*]` etc.