Atlas Cookbook (locale)

L. Sbordone & P. Bonifacio

last modified July 05 2005

Index

Introduction

This is intended as a small, non exaustive tutorial to learn the basics of ATLAS 9, Width and Synthe as distributed in our Linux porting. The idea is that a new user may initially use it to get the general picture of how things work, and then go deeper in what he personally needs for his work. At the moment only Atlas is extensively documented in Kurucz, L., SAOSR 309, 1970; although referring to a previous version of the code, it's up to date in the majority of the topics, and describes in depth the employed algorithms and both input and output file structure.
As a consequence, we will not explain the meaning of everything in the various inputs and outputs, concentrating on the purpose of:

1. Produce an atmosphere model for a star
2. Derive its iron abundance from the EW of some observed lines
3. Produce a synthetic spectrum of the same star (for example, to compare it with the observed one)

Of course this choice comes largely from the fact that this is what we use the ATLAS suite for...

Please, remember this is not gospel, many things described here can be accomplished differently, and the merits and limits of any solution to the problems are, of course, debatable.

NOTE: the scripts, input files, models and such cited here have been tested and should work. Nevertheless, most browser attempt to execute files with given extensions (such as .com files). Since we want them to be opened as text files for inspection for the purpose of this tutorial, they have been renamed when necessary by adding a .txt extension. if you want to run these scripts or use these input files, remove this extension.

ATLAS 9: The model calculation

As example, we will consider a giant star with Teff = 4904 K, log g 2.30, [Fe/H]=0. It's not significant for our purpose how we determined such atmospheric parameters.
We will derive this model from a previously existing one, which may be one taken from published grids (see for example the ones available on Bob Kurucz's site) as well one we derived in some previous work. We will use this starting model, which differs only by having log g = 2.4.
First of all let's give a look at the model file structure.

Model structure

CAVEAT: Atlas, Width and Synthe have been written with explicit FORMAT statements, so that ALL the formats ARE important. If the codes expect something like GRAVITY 2.30000, GRAVITY 2.30, or GRAVITY 2.30000 (more than 1 space between Y and 2) or whatever else different from the "original" WILL produce an error.

The first lines are something like:

```
TEFF   4904.  GRAVITY 2.40000 LTE
```

- Self explanatory, we have here Teff, the gravity and the indication that the model has been computed in LTE.

```
TITLE Sgr 628 (141) model 4
```
- Here is reproduced the title we assigned to the model during the calculation. Only for reference. It's treated as a string, so is free format. It can also be changed safely, of course.

```
OPACITY IFOP 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0
```
- Control cards defining which opacity sources have been taken into account in the model calculation.

```
CONVECTION ON   1.25 TURBULENCE OFF  0.00  0.00  0.00  0.00
```
- This indicates that convective transport had been used, 1.25 being the mixing length parameter. The successive lines start to define the abundances used...

```
ABUNDANCE SCALE   1.00000 ABUNDANCE CHANGE 1 0.92070 2 0.07836
```
- The abundance scale is a scaling factor that is then applied to all the subsequently defined abundances.. The SCALE is a multiplicative factor: assuming we are using a solar abundance pattern, a SCALE 1.00000 means that we have here a model of a [Fe/H]=0. star, while SCALE 0.31623 corresponds to a [Fe/H]=-0.5 (10^-0.5=0.31623), SCALE 0.1000 to a [Fe/H]=-1.0 and so on.
The first two ABUNDANCE CHANGE define H and He abundances. They are given in number fraction of the total atoms (ONLY for H and He)

```
ABUNDANCE CHANGE  3 -10.94  4 -10.64  5  -9.49  6  -3.52  7  -4.12  8  -3.21
```
- Here follows the definition of the abundances for the other elements. Each of these lines contains up to 6 elements. For each one there's first the atomic number, and then the abundance. For all non-H/He elements ("metals") the abundance is described as log(N(element))-log(N(H+He)). This is a different scale from the "traditional" one generally used in literature, which is given by log(N(element))-log(N(H))+12. With the PRESENT H and He solar abundances, we have:
A(traditional)=A(Kurucz)+12.04

For example, for iron Kurucz gives -4.54 +12.04= 7.5. The abundances listed in the example model are the ones of a NEW ODF model. See below for more details on this.

Then follows the model structure. Atlas computes up to 72 layers in the atmosphere.

```
READ DECK6 72 RHOX,T,P,XNE,ABROSS,ACCRAD,VTURB, FLXCNV,VCOMV,VELSND
 4.58588550E-03   2785.2 1.152E+00 5.708E+07 2.908E-05 9.104E-03 1.000E+05 0.000E+00 0.000E+00 7.836E+05
 5.99938449E-03   2835.2 1.507E+00 8.233E+07 3.385E-05 8.950E-03 1.000E+05 0.000E+00 0.000E+00 7.423E+05
 7.64013804E-03   2878.5 1.919E+00 1.132E+08 3.844E-05 8.731E-03 1.000E+05 0.000E+00 0.000E+00 7.118E+05
```

- The first line lists what is in the underlying columns. Atlas uses RHOX (integral density along the atmosphere, see Kurucz (1970)) as running variable. Then you have Temperature, Pressure and so on. The 7th column is the microturbulence for which the model has been calculated. As you can see Atlas do not list in the model all the physically interesting parameters, but only the ones that are needed to define the model structure. In fact we will see that all the other interesting parameters that are derived during the model calculation (like, say, the Mg II / Mg I ratio or the rosseland optical depth along the atmosphere...) are available in a complementary output file produced during model calculation (the .dat file, see below), but are not included in the model file to keep it small. Input script

See the input script here . The first thing you have to pay attention to is in:

```
#Opacity and ODF file calls
ln -s  /usr/local/kurucz/ODF/NEW/kapp00.ros fort.1
ln -s  /usr/local/kurucz/ODF/NEW/p00big1.bdf fort.9
ln -s  /usr/local/kurucz/lines/molecules.dat fort.2
```

- tcsh shell command that link some needed files to the proper names expected by atlas. The KAPxxx.ros files are rosseland opacity tabulations, and should be chosen accordingly to the metallicity of the chosen model. P00 -> [Fe/H]=0; M05 -> [Fe/H]=-0.5 and so on. the xxxbig1.bdf files are the NEW ODF, big type, used for model calculation, where the first three character are to be chosen like in the .ros file case. Here you have to pay attention to a (potentially) important detail: each ODF is calculated with a given microturbulence value, the "big1" in the name indicates a 1 km/s microturbulence. In principle, the Vturb for which the model is calculated should be the same for which the ODF has been calculated. See below for more details. The other link does not need to be changed.

```
#Starting model assignation
ln -s t4904g240p00a0vt10_l.mod fort.3
```

- this, rather obviously, is the model chosen as a starting guess. All its parameters (temperature, gravity, composition) may be different from the final one, a closer match will make the convergence faster. A caveat: every elemental abundance not explicitly declared (see below) will be taken from the starting model.

```
#Atlas is called and fed with his input control cards
/usr/local/kurucz/bin/atlas9mem_newodf.exe <<EOF
READ KAPPA
READ PUNCH
MOLECULES ON
READ MOLECULES
FREQUENCIES 337 1 337 BIG
```

- Atlas input starts here. It's composed as a series of "commands" or control cards that atlas expect coming from standard input. The first of this lines feeds everything that follows (until the "EOF" card is reached) to Atlas through stdin. Notice that this is the _newodf version of the code. The first cards instruct Atlas to use the opacity and molecular files provided above, and to calculate (MOLECULES ON) the molecular equilibrium. The last line relates to the kind of ODFs Atlas is being used.
For any change inside here remember the caveat given at the beginning about the explicit formats. Also, since here we are not dealing with shell script, but with explicit ATLAS commands, it is not possible, e.g., to comment a line, which would be considered as a command starting with a "#", and considered an error.

```
VTURB 1.0E+5
CONVECTION OVER 1.25 0
TITLE Sgr 628 (141) model 4
```

- The first line sets the microturbulence that will be used. in CGS, so, 1.0e+5 = 100,000 cm/s = 1 km/s. This is a good value for a model of a standard giant star, we will also see that models to be feed to Synthe are better calculated with microturbulence 1 km/s (see below for the reason). As stated above, this Vturb assignation should be coherent with the ODF employed. It is worth noticing that ATLAS 9 is way more sensitive to the used ODF: if you calculate a model with 1 km/s ODF but impose a 2 km/s VTURB, you will actually obtain the same model you would have had by correctly putting VTURB = 1 km/s.
The second line activates the treatment of convection. This is a sort of "hack": overshooting convection is implied, with mixing length parameter 1.25, but with overshooting parameter set to 0, which is equivalent to shutting off overshooting.
The third line assigns the title.

```
ABUNDANCE SCALE   1.00000 ABUNDANCE CHANGE 1  0.91930 2 0.07824
 ABUNDANCE CHANGE  3 -10.94  4 -10.64  5  -9.49  6  -3.52  7  -4.12  8  -3.21
 ABUNDANCE CHANGE  9  -7.48 10  -3.96 11  -5.71 12  -4.46 13  -5.57 14  -4.49
 ABUNDANCE CHANGE 15  -6.59 16  -4.71 17  -6.54 18  -5.64 19  -6.92 20  -5.68
 ABUNDANCE CHANGE 21  -8.87 22  -7.02 23  -8.04 24  -6.37 25  -6.65 26  -4.54
 ABUNDANCE CHANGE 27  -7.12 28  -5.79 29  -7.83 30  -7.44 31  -9.16 32  -8.63
 ABUNDANCE CHANGE 33  -9.67 34  -8.63 35  -9.41 36  -8.73 37  -9.44 38  -9.07
```

- Here we find again the same abundance cards we saw above in the model. This shows also a characteristic of Atlas inputs and outputs: Atlas, in fact, always reads commands that then it interprets. The process is exactly the same when it reads an input model or the calculation control cards. ABUNDANCE CHANGE commands instruct the code to assume that the rest of the line contains up to 6 abundance for specified elements. There's no need for them to be sequential, sorted or whatever. There's also no need that any line to contain elements different from the ones in any other. Any new AB. CH. line updates the elements that contains. So, as we will see below for Synthe, if you alter, say, the Si abundance, the best practice is to duplicate the related AB. CH. line and change the Si abundance in the SECOND one, so keeping track of what the original values was. since all the values not specified here remain the ones assigned in the starting model, we consider a good practice to keep here a complete, ordered set of ABUNDANCE CHANGE for all the elements. See (1) for the relationship between ODF and ABUNDANCE CHANGE cards

```
SCALE 72 -6.875 0.125 4904. 2.30
ITERATIONS 15 PRINT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
PUNCH 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
BEGIN ITERATION 10 COMPLETED
SCALE 72 -6.875 0.125 4904. 2.30
ITERATIONS 15 PRINT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
PUNCH 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
BEGIN ITERATION 10 COMPLETED
SCALE 72 -6.875 0.125 4904. 2.30
ITERATIONS 15 PRINT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
PUNCH 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
BEGIN ITERATION 10 COMPLETED
END
EOF
```

- Here the program is instructed to start the calculation. As you see, we have here three blocks repeated. Every block iterates the model 15 times, so that, if you assume to need 45 iterations, all you have to do is to repeat three modules. Consider that Atlas does not have ANY internal convergence test, it's up to you to verify at the end that your model has reached the convergence (see below how you do this verification). The first line of any module contains:

- The number of layers (72 is the maximum, does not make so much sense to put less)
- The position of the outermost atmospheric layer. The value is expressed in logarithm (base ten) of the rosseland optical depth, this to say, the outermost layer in this case will have rosseland optical depth of 10^(-6.875). The deepest value is always set to tauross = 100
- ?
- Teff
- Log g.

The second and third line host, first of all, the number of iteration, then two different groups of control card, PRINT and PUNCH. This is related to the fact that Atlas produces 2 kinds of outputs, the models (comes out on unit 7 inside the Fortran code, is linked to the .mod file in the script, see below) controlled by the PUNCH control card, and the standard output (unit 6 inside the code) controlled by the PRINT control card. You will notice that each card is followed by 15 numbers, 0 or 1. The meaning is: for each one of the 15 iteration, the output (model or stdout) is produced if the corresponding value is an 1, otherwise is suppressed. in the case of the example, they are produced, for both outputs, only for the last model of the block. In fact, does not make so much sense under Linux to output more than the final model, since it is overwritten every time is produced (under VMS different file version are produced). For the PRINT output some explanation is needed. Atlas sends out on stdout all the physical quantities calculated, this to say all the numeric density for all the ions of all the elements and molecules, temperature, density, rosseland opacity and optical depth, convective flux fraction, sound velocity, metric depth, radiative acceleration and much more, for every layer in the calculated atmosphere (The graph showed in astro-ph/0406268 are in fact derived from this output). For every model iteration that is PRINTed, is about 1600 lines of annotated, human readable tables. It's easy to see how this informations may be extremely interesting for you, to analyze the physics of the atmosphere you are computing. In fact, unless you are hacking the code, you will be interested only to the tabulations for the final atmosphere, so it may make sense to put 1 only in the last print card of the last execution block. By default, atlas prints this out to screen, which is both useless and slowing down the calculation: it is thus customary to redirect it to a file (see below).

```
    #The exit model is renamed
mv fort.7 t4904g230p00a0vt10_l.mod

#Some garbage is removed
rm -f fort.*
date
```

- The exit model is produced as fort.7, here is renamed at its final name. after this, temporary files are removed.

As briefly noted before, the input script is launched from shell, redirecting the STDOUT output to a file...

$./tutorial_atlas.com > logfile.log

logfile.log contains all the before indicated detailed calculations. Among other things, for every layer, error on the flux conservation and on the derivative of the flux are tabulated. The rightmost columns of the end of the file will look something like this:

```
 ERROR   DERIV
-0.001   -3.153
-0.001   -1.109
-0.001   -1.135
-0.001   -1.067
-0.001   -0.995
-0.001   -0.922
-0.001   -0.838
-0.001   -0.752
-0.001   -0.680
```

This tabulation is used to evaluate model convergence. Typically, values below 1% flux error and 10% flux derivative error along all the atmosphere are what we require to consider the model converging. You may want to to set this to different values, taking into account your needs and the code limitation: Atlas produces LTE one-dimensional models, so that, for example, at optical depths of about 10^-6 is bound to be wrong. Low gravity, low temperature models may not converge in the outermost layers, which are , by the way, insignificant for the synthesis of all but the strongest saturated line (which would not be correctly reproduced anyway, since there no LTE hypothesis can hold). So you may want to neglect this kind of convergence problems.
To quickly check the model convergence, you may use the provided PERL script ifconv.pl. You may copy it in a directory in your PATH, or copy (link) it in the working directory and launch it by typing

./ifconv.pl logfile.log

Which will produce an output like:

```
Checking convergence from file logfile.log

 wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww

The model is CONVERGING

Max flux error 0.099 at layer 72
Where T =   9172.2  and Rosseland Depth is  1.000E+02
Max flux derivative error 3.153 at layer 1
Where T =   2785.2  and Rosseland Depth is  1.334E-07
```

WIDTH: Abundance analysis

WIDTH typical user has measured, in some way, equivalent widths for a given number of (ideally unblended) lines of a given ion in a stellar spectrum, and wants to derive a mean abundance for this ion by averaging the abundance derived from every single line. To do this, WIDTH:

1. Takes as input each line: the assumed atomic data and the measured EW
2. Takes as input the chosen atmosphere model for the star.
3. Given the abundances in the model, recomputes the
4. Takes the abundance for the ion in the given atmosphere model, computes the line transfer in the given atmosphere, and derives the line EW.
5. Compares it with the measured one, and alters iteratively the assumed abundance until the two EW match.
6. Repeats the process for each line.
7. If the user asked for it, averages the abundances derived from a given set of lines, and produces three useful graphs for the averaged sample of lines: abundance vs. excitation potential, abundance vs. EW, abundance vs height in the atmosphere. For each one of them a linear fit is performed on the sample, and the value of the slope of the fitting line is provided.

This can be done for one or more ions in at the same time, by putting all them together in the WIDTH input file.

The input file

For the star for which we calculated the model above, we want to measure Fe I and Fe II abundances. This is the input file we are going to use. From above:

```
VTUR
  1 1.95 1.85 2.05
```

- The VTUR control card instructs width to look in the subsequent line for the adopted values of the microturbulence. Since microturbulence is typically one of the parameters that are set by looking at the abundance from various lines, typically Fe I lines (by imposing near-zero slope in the abundance vs. EW fit), WIDTH allows you to specify up to three microturbulence values, and all the calculations are repeated for each one of them. The first number indicates how much microturbulence values to consider, if, like here, is 1, the calculation will be performed for 1.95 only, otherwise you may put 2 or 3.

```
LINE     9.92  615.1617    SGR 141
 615.1617 -3.299  3.0   17550.175  2.0   33801.567    26.00(4P)4s a5P(4F)4p y5D
 615.1617   0  8.19 -6.20 -7.82FMW  0 0  0  0.000  0  0.000
```

- A group of three lines starting with the control card LINE identifies the record of a measured line. The first line contains the measured EW in picometers, then the measured wavelength for the line, in nanometers, then a label (SGR 141 here, but can be anything). Note that the measured wavelength is not not used by width, only taken along to the output. In the second line the first value is the laboratory wavelength, the second is the line log gf, then followed by the levels and energies of the transition. then it follows the ion identification (26.00 = Fe I) and the levels description. In the third line the laboratory wavelength is repeated, followed by other transition data, between which we find the code of the log gf bibliographic source (FMW for this line) These codes are totally free for you to set (except for the maximum length of 4 characters). See below for more informations on these codes. To prepare such line records, we have included in the suite a program called crea_width which purpose is to create a complete WIDTH input file from a set of measured lines and a chosen model. The program takes the line data from the Kurucz' stile linelists included in this suite, and described below when speaking about SYNTHE : of course, you may create an alternate linelists to have crea_width to produce records with the atomic data you favor, or change it to read in a different linelist format.

```
AVER
LINE ....
AVER
```

- Lines which derived abundances should be averaged are contained between two AVER control cards, which should be alone in a line. If you decide to exclude a given measured line from the averaged abundance of the ion, you may simply cut and paste it outside the AVER ... AVER block. Its abundance will still be calculated and included in the output file, but not used in the average. You may put as much AVER blocks you want in an input WIDTH file. WIDTH will understand no "nesting" of AVER blocks, so that the first AVER starts the block, the second one ends it, the third starts a new block and so on. If AVER control cards are not in even number, WIDTH will exit with error. It is safe to include a single line in an AVER block (as for example crea_width does): WIDTH will output it as an average abundance with 0.0 error, and produce no statistical graphs, that would make no sense. It may also be advisable to do so, since it allows you to grep the output file for ABUNDANCE statements and quickly retrieve the abundances of all ions, including the ones for which a single line was measured (see below for details about the output file).

```
LINE     7.10  526.4812    SGR 141
 526.4812 -3.190  2.5   26055.423  1.5   45044.168    26.01(3G)4s a4G(5D)4p z4D
 526.4812   0  8.61 -6.67 -7.95FMW  0 0  0  0.000  0  0.000
AVER
END
TEFF   4904.  GRAVITY 2.30000 LTE
TITLE Sgr 628 (141) model 4
 OPACITY IFOP 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0
```

- After the last LINE block, and the closing AVER, an END control card tells WIDTH that all the lines have been fed and it's time to look for the model. This is simply the ATLAS model, but notice that the very end is edited, since now it looks like:

```
PRADK 7.7203E-01
READ MOLECULES
MOLECULES ON
BEGIN                    ITERATION  15 COMPLETED
END
```

```
STOP
```

The launch script

The WIDTH launch script is very simple:

```
#!/bin/csh -vf
rm -f fort.*
date
ln -s /usr/local/kurucz/lines/molecules.dat fort.2
/usr/local/kurucz/bin/width9.exe < tutorial_atlas.wid
```

Essentially, it simply links some needed files into the appropriate local names, inputs (from STDIN) the width input file (the .wid) and launches the program. WIDTH outputs on STDOUT, so you must redirect it to file, for example:
./tutorial_atlas_width.com > tutorial_atlas.out

The output file

WIDTH produces a massive output file . ATLAS model files, as above seen, contain only the minimum set of informations needed to constrain the atmosphere structure, but all the rest (e.g. the number densities for each molecule along the atmosphere) need to be recalculated in order for WIDTH to compute the line transfer. To do so, WIDTH (and SYNTHE) calls ATLAS as a subroutine (as you may have noted in the makefile a version of ATLAS is compiled within WIDTH and SYNTHE), and the relative results constitute the first part of the output. This part of the output is typically useless, so that anything included between something like:

```
Fri Feb  4 10:39:12 UTC 2005
1MOLECULES INPUT
    1          1.00  0.000 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
    2          1.01  0.000 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
    3          2.00  0.000 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
```

and something else like:

```
0XNFPOH  5.9273E+02  6.8275E+02  7.9394E+02  9.0672E+02  1.0229E+03  1.1505E+03  1.2964E+03  1.4647E+03  1.6598E+03  1.8848E+03
         2.1446E+03  2.4491E+03  2.8138E+03  3.2369E+03  3.7431E+03  4.3324E+03  5.0498E+03  5.8988E+03  6.8840E+03  8.0746E+03
         9.3951E+03  1.1764E+04  1.3862E+04  1.9528E+04  2.1035E+04  2.9429E+04  3.7962E+04  4.9438E+04  6.3580E+04  8.0912E+04
         1.0175E+05  1.2645E+05  1.5448E+05  1.8452E+05  2.0924E+05  2.4024E+05  2.7957E+05  3.2855E+05  3.8852E+05  4.5956E+05
         5.4596E+05  6.4964E+05  7.7093E+05  9.1222E+05  1.0703E+06  1.2464E+06  1.4321E+06  1.6178E+06  1.7829E+06  1.9007E+06
         1.9586E+06  1.9719E+06  1.9168E+06  1.7714E+06  1.5412E+06  1.2499E+06  9.3548E+05  6.3971E+05  4.0258E+05  2.3555E+05
         1.2965E+05  6.9865E+04  4.2107E+04  2.9555E+04  2.2883E+04  1.8530E+04  1.5436E+04  1.3093E+04  1.1182E+04  9.6754E+03
         8.3277E+03  7.2960E+03



 615.1617 -3.299 3.0   17550.175  2.0   33801.567    26.00(4P)4s a5P(4F)4p y5D
 615.1617   0  8.19 -6.20 -7.82FMW  0 0  0  0.000  0  0.000
0    SGR 141                                          615.1617  8.19 -6.20 -7.82  1.00     9.92    -5.004
```

(about 1560 lines on a 72 layers model) can be ignored, unless you are interested, e.g., in the ionization structure of the atmosphere. At the end of the above reproduced part we see part of the first atomic line output record, that will in general look like:

```
 615.1617 -3.299 3.0   17550.175  2.0   33801.567    26.00(4P)4s a5P(4F)4p y5D
 615.1617   0  8.19 -6.20 -7.82FMW  0 0  0  0.000  0  0.000
0    SGR 141                                          615.1617  8.19 -6.20 -7.82  1.00     9.92    -5.004
        VTURB  ABUND    -5.53    -5.03    -5.00    -4.53    -5.00
         1.95    EW    0.786    0.989    0.998    1.113    0.997
                       6.10     9.74     9.96    12.98     9.92
              DEPTH     1.32     1.02     1.00     0.72     1.00
```

In the first two lines here wee see reproduced the atomic data, laboratory wavelength log gf and so on. The third line (the one starting with a 0 in the very first column) holds the line label, the observed wavelength, three other values we will not describe here, the rosseland optical depth of formation of the best fitting line, the observed EW in picometers, and the derived ion abundance for that line. This last one is expressed in the same way above found in the ATLAS models , but, this time, the true, non scaled abundance is presented. In other words, you do not have to take into account the ABUNDANCE SCALE value in the model: -5.004 +12.04 => A(Fe) = 7.036 (in the Grevesse & Sauval scale) according to this line.
In the following four lines we have an output of the fitting process followed by WIDTH in search of the final abundance value. At the left, in vertical we have the indication of the employed microturbulence (VTURB 1.95), at the right, the successive attempts made by WIDTH are listed, first row the abundance for which the calculation has been performed, below the EW, and finally the formation depth for the line (Rosseland). The rightmost column is the one at which WIDTH decided that convergence had been reached. In a sense, this provides a small curve of growth for this transition, since EW is computed at different abundances, but this is not the best way to obtain a curve of growth. WIDTH has a dedicated command to produce a curve of growth, should the used need it.

After the output for all the lines in an AVER block, the mean result is given:

```
***** THE ABUNDANCE FROM   9       26.00 LINES IS  -4.77+/- 0.11
```

followed by the three statistical plots we have cited above. Their text-based graphics look pretty old fashioned, but is nevertheless effective. Here we show the first one, excitation potential vs. abundance. The other two show EW and height in the atmosphere, both against abundance:

```
   LOG ABUND VS. CHI
   SLOPE= 0.971E-01 PER EV USING    9 LINES
    4.80 .----.----.----.----.----.----X----.----.----.----.----.----.
    4.70 .    .    .    .    .    .    I    .    .    .    .    .    .
    4.60 .    .    .    .    .    .    I    .    .    .    .    .    .
    4.51 .    .    .    .    .    .    I    .    .    .    .    .    .
    4.41 .    .    .    .    .    .    I    .    .    .    .    .    .
    4.32 .    .    .    .    .    .  X I    .    .    .    .    .    .
    4.22 .    .    .    .    .    .  X X XI   X    .    .    .    .    .
    4.12 .    .    .    .    .    .    I    .    .    .    .    .    .
    4.03 .    .    .    .    .    .    I    .    .    .    .    .    .
    3.93 .    .    .    .    .    .  X  I    .    .    .    .    .    .
    3.84 .----.----.----.----.----.----I----.----.----.----.----.----.
    3.74 .    .    .    .    .    .    I    .    .    .    .    .    .
    3.64 .    .    .    .    .    .    I    .    .    .    .    .    .
    3.55 .    .    .    .    .    .    I    .    .    .    .    .    .
    3.45 .    .    .    .    .    .    I    .    .    .    .    .    .
    3.36 .    .    .    .    .    .    I    .    .    .    .    .    .
    3.26 .    .    .    .    .    .    I    .    .    .    .    .    .
    3.16 .    .    .    .    .    .    I    .    .    .    .    .    .
    3.07 .    .    .    .    .    .    I    .    .    .    .    .    .
    2.97 .    .    .    .    .    .    I    .    .    .    .    .    .
    2.88 .----.----.----.----.----.----I----.----.----.----.----.----.
    2.78 .    .    .    .    .    .  I X    .    .    .    .    .    .
    2.69 .    .    .    .    .    .    I    .    .    .    .    .    .
    2.59 .    .    .    .    .    .    I    .    .    .    .    .    .
    2.49 .    .    .    .    .    .    I    .    .    .    .    .    .
    2.40 .    .    .    .    .    .    I    .    .    .    .    .    .
    2.30 .    .    .    .    .    .    I    .    .    .    .    .    .
    2.21 .    .    .    .    .    .    I    . X    .    .    .    .    .
    2.11 .    .    .    .    .    .    I    .    .    .    .    .    .
    2.01 .    .    .    .    .    .    I    .    .    .    .    .    .
    1.92 .----.----.----.----.----.----I----.----.----.----.----.----.
    1.82 .    .    .    .    .    .    I    .    .    .    .    .    .
    1.73 .    .    .    .    .    .    I    .    .    .    .    .    .
    1.63 .    .    .    .    .    .    I    .    .    .    .    .    .
    1.53 .    .    .    .    .    .    I    .    .    .    .    .    .
    1.44 .    .    .    .    .    .    I    .    .    .    .    .    .
    1.34 .    .    .    .    .    .    I    .    .    .    .    .    .
    1.25 .    .    .    .    .    .    I    .    .    .    .    .    .
    1.15 .    .    .    .    .    .    I    .    .    .    .    .    .
    1.05 .    .    .    .    .    .    I    .    .    .    .    .    .
    0.96 .----.----.----.----.----.----I----.----.----.----.----.----.
    0.86 .    .    .    .    .    .    I    .    .    .    .    .    .
    0.77 .    .    .    .    .    .    I    .    .    .    .    .    .
    0.67 .    .    .    .    .    .    I    .    .    .    .    .    .
    0.58 .    .    .    .    .    .    I    .    .    .    .    .    .
    0.48 .    .    .    .    .    .    I    .    .    .    .    .    .
    0.38 .    .    .    .    .    .    I    .    .    .    .    .    .
    0.29 .    .    .    .    .    .    I    .    .    .    .    .    .
    0.19 .    .    .    .    .    .    I    .    .    .    .    .    .
    0.10 .    .    .    .    .    .    I    .    .    .    .    .    .
    0.00 .----.----.----.----.----.----I----.----.----.----.----.----.
      -3.77    -3.97    -4.17    -4.37    -4.57    -4.77    -4.97    -5.17    -5.37    -5.57    -5.77
1VTURB 1.950      SGR 141
```

Notice that the plot is drawn in such way that the fitting line would be vertical when its slope is zero. A near-to-zero slope in the abundance vs. EW graph, for example, is the typical criterium employed to infer that a correct microturbulence has been chosen for the star. By asking WIDTH to compute the abundances for three microturbulences around a value you consider likely, you can easily find the value for which the slope sign changes. As can be seen above, each line is identified by a X in the graph. Should two lines fall above the same point at the graph resolution, the X will be substituted by a "2".

Other WIDTH features

To be done

SYNTHE: spectral synthesis

SYNTHE is not actually a single program, but a suite of different programs , called one after the other by an input script with the purpose of producing a synthetic spectrum. To do it, SYNTHE takes as input the chosen atmosphere model (where the chosen chemical abundances are also listed), the lists of the atomic and molecular transitions we want to include in the calculation (the lines we want to "see" in the synthetic spectrum), and of course the parameters of the calculation such as the spectral range we want to synthesize.
SYNTHE takes the previously calculated atmosphere structure from the model, and similarly to what WIDTH does, recalculates the excitation and ionization populations for the ions, which is not contained in the model. Since often you may want to use observed abundances in the synthesis, this may lead, in extreme cases, to inconsistencies where, in the model calculation, solar scaled abundances have been used. See an example in
Sbordone et al. 2005
. In these cases, it could be necessary to resort to an opacity sampling model, such as the ones produced by ATLAS 12 or MARCS.

The atmosphere model

SYNTHE obviously needs as input the model of the atmosphere for which the synthetic spectrum should be calculated. Nevertheless, the model produced by ATLAS cannot be feed "as it is" into SYNTHE. The SYNTHE-ready model differs in one thing: some control cards should be added on top of the model. For the purpose of comparing a synthetic spectrum against the observed spectrum of a (non rotating) star we need to produce a SURFACE FLUX spectrum, taking into account the molecular contribution. The consequent series of control cards is:

```
SURFACE FLUX
ITERATIONS 1 PRINT 2 PUNCH 2
CORRECTION OFF
PRESSURE OFF
READ MOLECULES
MOLECULES ON
```

We will show later how to synthesize the spectrum of a rotating star .

The input script

The input script links the input files and calls the programs needed to read them and produce the synthetic spectrum. For example, we will derive a spectral synthesis of the spectral range around the Na I D doublet ~589 nm. The input script we present here is a very basic one, an example of a more powerful one can be found below .
First of all, we create some variables needed to build the product filenames:

```
#!/bin/tcsh
#example SYNTHE input script, for the ATLAS cookbook.
rm -f for00*
rm -f fort.*
set model = syn_nr_t4904g230p00a0vt10_l.mod
set teff = t4904
set glog = g230
set met = p00
```

Then we proceed to link some needed input files to their "internal" names. These lines do not need to be tipically changed

```
ln -s  /usr/local/kurucz/lines/he1tables.dat fort.18
ln -s /usr/local/kurucz/lines/molecules.dat    fort.2
ln -s /usr/local/kurucz/lines/continua.dat  fort.17
```

Then we feed the model to the program used to read it in, XNFPELSYN. It executes the first calculations:

```
/usr/local/kurucz/bin/xnfpelsyn.exe < $model
```

And finally we pass the first block of control cards to the program SYNBEG, in a fashion similar to the one we used before for ATLAS input:

```
/usr/local/kurucz/bin/synbeg.exe <<EOF
AIR         588.0    590.0    600000.    1.67    0    30    .0001    1    0
AIRorVAC  WLBEG     WLEND     RESOLU    TURBV  IFNLTE LINOUT CUTOFF        NREAD
EOF
```

Here we have some very important control cards. In the first line of control cards we have the values, the second acts as a reminder:

- AIR indicates that the wavelengths are in AIR. VAC would provide vacuum wavelengths
- WLBEG and WLEND are the starting and ending points of the synthesis, in nanometers
- RESOLU is the resolution at which the calculation is performed. Practically, SYNTHE calculates the transfer through the atmosphere at wavelength intervals with such spacing. Of course, reducing the resolution will lead to a faster calculation, but also to a poorer sampling of the radiative transfer through the atmosphere. We thus suggest not to go below a resolution of 100000. This value is adequate for comparison with high resolution observed spectra.
- TURBV is the microturbulence we want SYNTHE to add to the one in the atmosphere model. Since microturbulence is added by summing the squares, and we have a VTURB=1 model, we need to add 1.67 to obtain the final 1.95 km/s.
- IFNLTE is set to 0 because we want a LTE calculation
- CUTOFF is used to keep the weakest transitions out of the output files. With this setting, any absorption subtracting at its center less than 1/10000 of the intensity will be cut off.

Subsequently, the script starts to build the linelist. SYNTHE uses two formats for the linelists, the first one for atomic lines, the second for molecular transitions. As a consequence, two different programs (RLINE2 and RMOLECASC respectively) are used to read them in and add them to the global linelist.
To read in an atomic linelist we have something like:

```
ln -s /usr/local/kurucz/lines/gf0600.100 fort.11
/usr/local/kurucz/bin/rline2.exe
rm -f fort.11
```

The gf****.100 files are distributed with this Linux port, and also available on Bob Kurucz's site . In these files, lines are grouped in 100 nm chunks ending with the number in the file name. For example, the gf0600.100 file will contain all the atomic transitions between 500 and 600 nm. This general rule is not fully respected by the files gf0800.100 and gf 1200.100 which contain 200 nm chunks (600-800 nm and 800 1200 nm respectively) due to the lower number of atomic transitions in these ranges. Given the range of our example calculation, we only need the gf0600; should you need to synthesize spectra that go beyond the range covered by a single file, you will simply need to add all the files you need in sequence, such as:

```
ln -s /usr/local/kurucz/lines/gf0500.100 fort.11
/usr/local/kurucz/bin/rline2.exe
rm -f fort.11
ln -s /usr/local/kurucz/lines/gf0600.100 fort.11
/usr/local/kurucz/bin/rline2.exe
rm -f fort.11
```

In general, it is easier to always read in all the atomic linelists instead of always changing it to the appropriate one. It may nevertheless have some significant impact on the computation time on slow or low memory systems. The atomic linelists are in plain ASCII format, making them easy to read and modify, should you prefer, for example, different log gf values for some lines. Programs are also available to convert in "Kurucz format" the linelists of popular databases like VALD. The format of the gf****.100 files is clearly described in the related section of Bob Kurucz's site . Here we will only mention that the bibliographic reference codes that appear in these lines (and in many other places like in the WIDTH input and output), and refer to the bibliographic source of the line log gf, are listed in the gfall.ref file shipped with the Linux port of the ATLAS suite. At the same site, you may also find the versions of the linelists taking into account (where appropriate) HFS splitting of the lines. They are read in the same way.

NOTE: Since all and only the lines passed to SYNTHE here will be considered in the synthesis, should you need to synthesize, say, a single line, you may accomplish it by feeding rline (or rmolecasc) with an input file containing only the transition you are interested in.

The molecular linelists are read by blocks like these:

```
ln -s /usr/local/kurucz/molecules/h2bx.dat fort.11
/usr/local/kurucz/bin/rmolecasc.exe
rm -f fort.11
ln -s /usr/local/kurucz/molecules/nhax.dat fort.11
/usr/local/kurucz/bin/rmolecasc.exe
rm -f fort.11
ln -s /usr/local/kurucz/molecules/sihax.dat fort.11
/usr/local/kurucz/bin/rmolecasc.exe
rm -f fort.11
```

Here the format is different: the molecular lines are packed "species-wise", and independently of wavelength, so that e.g. all the CN transitions will be in the same file. Again, refer to the appropriate section at Kurucz's site for detailed informations. As a consequence, you are tipically bound to read them all in regardless the spectral range of interest.

After this, the "true" SYNTHE is called:

```
/usr/local/kurucz/bin/synthe.exe
```

Then the input for SPECTRV is prepared, first the model, then a series of control card are put inside fort.25 for SPECTRV to read them. These control cards do not need to be changed for our purposes.

```
ln -s $model fort.5
set  outspec = ${teff}${glog}${met}.flx
cat <<EOF >fort.25
0.0      0.       1.       0.       0.       0.       0.       0.
0.
RHOXJ     R1       R101     PH1      PC1      PSI1     PRDDOP    PRDPOW
EOF
/usr/local/kurucz/bin/spectrv.exe
ln -s $outspec fort.21
mv fort.7 ${outspec}
rm fort.5
```

Spectrv completes the synthesis calculation. As you may see, here an "outspec" file name is created by using the above defined variables so that its name reflects the calculation parameters. This filename is given to the output file of spectrv. The "flx" format chosen to be binary, to reduce the size of the output, but this makes it unreadable by most plotting packages, so we will convert it to ASCII by means of the SYNTOASCANGA program. Another thing we need in order to compare the synthesis with an observed spectrum is to broaden it to the instrumental resolution of interest. This may well be done by using other kind of tasks, but in the suite is included an ad hoc program able to read the binary .flx files, BROADEN. All this is accomplished in the final lines:

```
set brspec = br_${outspec}
ln -s $brspec fort.22
/usr/local/kurucz/bin/broaden.exe << EOF
GAUSSIAN        7.00KM
EOF
```

Here we create a name for the broadened version of outspec, which will still be in .flx format. Then we launch BROADEN and pass with the usual trick two control cards, the kind of broadening (GAUSSIAN) and the desired FWHM, in km/s (7 km/s ~ 42000 resolution). Now that we have both the unbroadened and broadened version of the spectrum, we produce an ASCII version of both. A notice is in order here: the .flx file contains both the spectrum and the informations on the lines appearing in it (wavelength, log gf, ion or molecule, residual intensity etc.), useful to identify them when plotting the spectrum. SYNTOASCANGA will separate them in two different files,one for the spectrum (which we usually call .asc) and one for the linelist (.dat), so we need two output filenames.

```
ln -s  $outspec  fort.1
set asc = tutorial_synthe_br70.asc
set lines = tutorial_synthe_br70.dat
set lines_nob = tutorial_synthe_nb.dat
set asc_nob = tutorial_synthe_nb.asc
rm -f fort.2
ln -s $lines_nob fort.3
ln -s $asc_nob fort.2
ln -s  dmp.dmp fort.4
/usr/local/kurucz/bin/syntoascanga.exe
```

Here we link the outspec as input for sytoascanga, set the filenames (for both unbroadened and broadened spectrum), link the outputs of syntoascanga to the names for the unbroadened, and launch the program

```
rm -f fort.2
rm -f fort.1
rm -f fort.3
ln -s $brspec  fort.1
ln -s $asc fort.2
ln -s $lines fort.3
/usr/local/kurucz/bin/syntoascanga.exe
rm -f fort.*
```

Here, finally, we do the same for the broadened spectrum, then clean up some garbage. This ends the script.

The output files

SYNTHE, or better the script we described above, produces various outputs. If you launch the script "as it is" you get a massive screen output, which is way better to redirect to a file (screen output slows down computation a lot):

./tutorial_atlas_synthe.com > log.synthe

This output is mostly useful for debug purposes, and in most cases you will be fine overwriting it at the next calculation (or "harvesting" it for informations you may want to keep and then deleting it). Then you have the .flx output files, possibly broadened and unbroadened, and the .asc and .dat files. It is up to you to decide whether to keep them all or not. Here we will briefly describe the formats of the .asc and .dat files
THE .ASC FILE (unbroadened and broadened) contains the actual spectrum, on four columns:

```
 5892.3967    0.33850954E-05    0.38414344E-05    0.881206
 5892.4066    0.33574777E-05    0.38414424E-05    0.874015
 5892.4164    0.33321944E-05    0.38414504E-05    0.867431
 5892.4262    0.33105134E-05    0.38414583E-05    0.861786
 5892.4360    0.32936416E-05    0.38414663E-05    0.857392
 5892.4458    0.32826049E-05    0.38414742E-05    0.854517
```

wavelength (Å), real flux, "continuum" flux, and residual intensity, practically (column 3)/(column 2)

THE .DAT FILE contains the linelist for the synthesis you made:

```
 589.2466 -2.173 2.0  54375.673  1.0  37409.552   26.00  K94   0.7820
 589.2662 -2.289 3.0  31008.995  3.0  47974.554   24.00  K88   0.9908
 589.2693 -2.288 3.0  51294.217  3.0  34328.750   26.00  K94   0.6413
 589.2723 -1.781 1.5  86710.837  0.5  103676.220   26.01  K88   1.0000
 589.2745 -1.494 4.0  50466.172  3.0  33500.854   28.00  K88   0.7005
 589.2800 -4.030 2.0  17726.987  1.0  34692.146   26.00  FMW   0.3207
 589.2868 -2.350 0.0  16017.317  1.0  32982.280   28.00  FMW   0.1790
 589.2897 -2.589 2.5  62905.810  2.5  45940.930   25.00  K88   0.9999
 589.3036 -4.103 3.5  21646.420  3.5  38610.900   23.00  K88   1.0000
 589.3037  0.280 0.5  62402.150  1.5  79366.627   32.01  MIG   0.9993
 589.3133 -4.210 2.5  17054.960  1.5  34019.160   23.00  K88   1.0000
```

Which, in a fashion similar to the one in the gf****.100 files, contains the wavelength (nm) the log gf, the transition levels, the ion, the bibliographic reference of the log gf, and the unbroadened residual intensity at line center. This last value is obviously useful to sort out which ones are the important transitions, or when labeling them in a plot. We want to repeat the the r.i. provided is the UNBROADENED one, BROADEN does not recalculate these values. As a consequence you may guess that the .dat files from the broadened and unbroadened spectra are, in fact exactly the same.

Rotating stars

To simulate a rotating star, we need to compute separately the surface intensities at many different inclinations through the atmosphere. The "vertical" intensity will then be added (with proper surface rescaling) to the "inclined" components, which are properly Doppler shifted to simulate the rotational velocity, faster at the star limbs. In order to do this, we need to:

- instruct the program, though the cards added on top of the model , to calculate a given number of intensities, at given angles, instead of the SURFACE FLUX. This is accomplished by using this set of "topcards":

```
SURFACE INTENSI 17 1.,.9,.8,.7,.6,.5,.4,.3,.25,.2,.15,.125,.1,.075,.05,.025,.01
      ITERATIONS 1 PRINT 2 PUNCH 2
      CORRECTION OFF
      PRESSURE OFF
      READ MOLECULES
      MOLECULES ON
```

Where we instruct the program to compute 17 surface intensities, of which we give the sinuses of the latitude. Such a division should be adequate for any rotating star calculation.

- After the sequence of codes have produced the needed intensities, we need to perform the actual "rotation" of the spectrum. This is performed by the ROTATE program, which is called just after SPECTRV, and produces the final .flx file, to be fed to BROADEN for the instrumental boradening and/or to SYNTOASCANGA for the "translation" into ASCII. Here we see the proper form of the last part of the input script :

```
      mv fort.7 ${outspec}
      ln -s $outspec fort.1
      /data/Kurutar/bin/rotate.exe<<EOF
         1
      50.
      EOF
      mv ROT1 ${outspec}
      ln -s $outspec fort.21
      rm fort.5 fort.1
      set brspec = br_${outspec}
```

The the output of SPECTRV (fort.7) is as usual moved into $outspec. This .flx file is not a "final" one, since it contains the 17 different intensities we cjust calculated. It is then linked to fort.1, where ROTATE expects its input. Then ROTATE is called and its arguments are passed by means of the usual "EOF trick". In the first line is written the number of velocity rotations (v sin i, in fact) for which we want to calculate, since ROTATE can compute up to 20 rotated spectra for different v sin i. In the second one are written the velocities: since we want to calculate for a single velocity, we have only a 50 (km/s) here. Rotate places its output into files called ROT1, ROT2... up to ROT20. In this case we obviously have only ROT1, which we move back into $outspec. From here on everything is unchanged with respect to the "non rotating" script. The resulting synthetic spectrum can be seen here and here (links refer to the "unbroadened" ones, although the difference is small since the 7 km/s instrumental broadening is much smaller than the 50 km/s rotation).

Other utility programs: line lists creation, iterated calculations, line table compilation etc.

CREA_WIDTH

To be done

Footnotes

1. ODF NEW and OLD: The Opacity Distribution Functions (ODF) are the pretabulated opacities for a grid of gas pressures and temperatures, employed by ATLAS9. ODFs are calculated with a given microturbulence and a given chemical composition. When calculating a model, you should be sure that you are using the proper ODF: as cited in the text, a M05ABIG1 odf, for example, means [Fe/H]=-0.5, alpha enhanced, 1 km/s microturbulence. Another important distinction is the one between NEW and OLD type ODF. NEW type ODF are the ones described by Castelli & Kurucz 2003 (see the Documentation), and are the ones included with this port. With respect to the OLD type ODF, which can be downloaded from
Bob Kurucz's site
, the NEW ODFs use an improved set of molecular lines, a newer set of solar abundances, and a wider grid of precalculated temperatures. Two important consequencesw arise: the format is different between NEW and OLD ODFs, so that different versions of ATLAS are required (accordingly labelled _newodf and _oldodf). Trying to read, e.g., a NEW ODF with _oldodf atlas will lead to an I/O error. Also, you are supposed to use the proper set of solar abundances in your model calculation, i.e. the ones used during the ODF calculation. Otherwise, the numerical abundances derived for the ions during the actual model calculation at a given gas state (T and P) will differ from the ones calculated during the ODF computation. It has to be nevertheless noted that the differences between the two set of solar abundances are small, and using an "old" abundance set with a NEW odf will produce very small inconsistencies, almost negligible in most cases.

2. Shortly, 2 sets of ABUNDANCE CHANGE lines may be needed with the provided "new" ODFs:new type odf, solar scaled (the ABUNDANCE SCALE should be set propery, see text)

```
 ABUNDANCE SCALE   x.xxxxx ABUNDANCE CHANGE 1 0.91930 2 0.07824
  ABUNDANCE CHANGE  3 -10.94  4 -10.64  5  -9.49  6  -3.52  7  -4.12  8  -3.21
  ABUNDANCE CHANGE  9  -7.48 10  -3.96 11  -5.71 12  -4.46 13  -5.57 14  -4.49
  ABUNDANCE CHANGE 15  -6.59 16  -4.71 17  -6.54 18  -5.64 19  -6.92 20  -5.68
  ABUNDANCE CHANGE 21  -8.87 22  -7.02 23  -8.04 24  -6.37 25  -6.65 26  -4.54
  ABUNDANCE CHANGE 27  -7.12 28  -5.79 29  -7.83 30  -7.44 31  -9.16 32  -8.63
  ABUNDANCE CHANGE 33  -9.67 34  -8.63 35  -9.41 36  -8.73 37  -9.44 38  -9.07
  ABUNDANCE CHANGE 39  -9.80 40  -9.44 41 -10.62 42 -10.12 43 -20.00 44 -10.20
  ABUNDANCE CHANGE 45 -10.92 46 -10.35 47 -11.10 48 -10.27 49 -10.38 50 -10.04
  ABUNDANCE CHANGE 51 -11.04 52  -9.80 53 -10.53 54  -9.87 55 -10.91 56  -9.91
  ABUNDANCE CHANGE 57 -10.87 58 -10.46 59 -11.33 60 -10.54 61 -20.00 62 -11.03
  ABUNDANCE CHANGE 63 -11.53 64 -10.92 65 -11.69 66 -10.90 67 -11.78 68 -11.11
  ABUNDANCE CHANGE 69 -12.04 70 -10.96 71 -11.98 72 -11.16 73 -12.17 74 -10.93
  ABUNDANCE CHANGE 75 -11.76 76 -10.59 77 -10.69 78 -10.24 79 -11.03 80 -10.91
  ABUNDANCE CHANGE 81 -11.14 82 -10.09 83 -11.33 84 -20.00 85 -20.00 86 -20.00
  ABUNDANCE CHANGE 87 -20.00 88 -20.00 89 -20.00 90 -11.95 91 -20.00 92 -12.54
  ABUNDANCE CHANGE 93 -20.00 94 -20.00 95 -20.00 96 -20.00 97 -20.00 98 -20.00
  ABUNDANCE CHANGE 99 -20.00
 new type odf alpha enhanced
```

ABUNDANCE SCALE x.xxxxx ABUNDANCE CHANGE 1 0.92130 2 0.07842 ABUNDANCE CHANGE 3 -10.94 4 -10.64 5 -9.49 6 -3.52 7 -4.12 8 -2.81 ABUNDANCE CHANGE 9 -7.48 10 -3.56 11 -5.71 12 -4.06 13 -5.57 14 -4.09 ABUNDANCE CHANGE 15 -6.59 16 -4.31 17 -6.54 18 -5.24 19 -6.92 20 -5.28 ABUNDANCE CHANGE 21 -8.87 22 -6.62 23 -8.04 24 -6.37 25 -6.65 26 -4.54 ABUNDANCE CHANGE 27 -7.12 28 -5.79 29 -7.83 30 -7.44 31 -9.16 32 -8.63 ABUNDANCE CHANGE 33 -9.67 34 -8.63 35 -9.41 36 -8.73 37 -9.44 38 -9.07 ABUNDANCE CHANGE 39 -9.80 40 -9.44 41 -10.62 42 -10.12 43 -20.00 44 -10.20 ABUNDANCE CHANGE 45 -10.92 46 -10.35 47 -11.10 48 -10.27 49 -10.38 50 -10.04 ABUNDANCE CHANGE 51 -11.04 52 -9.80 53 -10.53 54 -9.87 55 -10.91 56 -9.91 ABUNDANCE CHANGE 57 -10.87 58 -10.46 59 -11.33 60 -10.54 61 -20.00 62 -11.03 ABUNDANCE CHANGE 63 -11.53 64 -10.92 65 -11.69 66 -10.90 67 -11.78 68 -11.11 ABUNDANCE CHANGE 69 -12.04 70 -10.96 71 -11.98 72 -11.16 73 -12.17 74 -10.93 ABUNDANCE CHANGE 75 -11.76 76 -10.59 77 -10.69 78 -10.24 79 -11.03 80 -10.91 ABUNDANCE CHANGE 81 -11.14 82 -10.09 83 -11.33 84 -20.00 85 -20.00 86 -20.00 ABUNDANCE CHANGE 87 -20.00 88 -20.00 89 -20.00 90 -11.95 91 -20.00 92 -12.54 ABUNDANCE CHANGE 93 -20.00 94 -20.00 95 -20.00 96 -20.00 97 -20.00 98 -20.00 ABUNDANCE CHANGE 99 -20.00