Working notes on ATLAS 9 and SYNTHE

Urtzi Jauregi
Faculty of Mathematics and Physics
University of Ljubljana.
urtzi@fmf.uni-lj.si

4th July 2005

Contents

1 Introduction 2
1.1 Purpose of this document . . . 2
1.2 Structure of this document . . . 3
1.3 Where to get. 3

2 ATLAS 4
2.1 Input/output in ATLAS . . . 4
2.2 Using ATLAS . . . 6
2.2.1 Creating a new model . . . 6
2.2.2 Modifying an existing model . . . 7
2.2.3 Calculating the radiation field . . . 8
2.3 The ATLAS control cards . . . 9
2.3.1 Cards to compute a new model . . . 9
2.3.2 Cards to control various physical processes . . . 11
2.3.3 Cards to control the output . . . 12
2.3.4 Other cards . . . 13
2.4 Miscellaneous remarks . . . 13
2.4.1 Integration points for the radiation field . . . 13

3 SYNTHE 13
3.1 Introduction . . . 14
3.2 Adding command headers to model files . . . 14
3.3 Sample COM file . . . 15
3.4 Line data . . . 20
3.4.1 Atomic line data . . . 21
3.4.2 Molecular line data . . . 22

3.5 Miscellaneous remarks 23
3.5.1 ROTATE and fluxes 23

4 Non-VMS distributions of ATLAS and SYNTHE 24
4.1 The Sbordone-Bonifacio distribution 24
4.2 The Lester distribution 24

5 TO DO 24

6 Thanks to... 25

A A complete ATLAS model 25

B A complete SYNTHE COM file 28

1 Introduction

1.1 Purpose of this document

This document is a quick overview of the main features of Kurucz's model atmosphere programs ATLAS 9 ("ATLAS" for short) and SYNTHE. It is mainly based on my experiences with it, both as a user and a developer, and tries to include what I think a beginner should know. Its aims are:

- To provide you with a basic working knowledge of ATLAS and SYNTHE, so that you can immediately begin to use it;
- to update the information on existing documents about earlier versions of the programs (such as the guide about ATLAS 8 by Castelli (1988)) or, at least, to point out when that information is obsolete;
- to provide a user-oriented introduction (the only one I am aware of) to the spectrum synthesis suite SYNTHE.

These notes are by no means complete. They aim at conciseness and clarity over depth. The reader wishing to use ATLAS for serious work should at least read Castelli (1988), where all the details of the ATLAS 8 user interface are thoroughly described[1] and can be directly applied to ATLAS 9 with the exceptions pointed out here. The physics behind ATLAS and the inner workings of the first released version (ATLAS 5) are explained in depth by Kurucz (1970); althought ATLAS 5 itself is obsolete since long ago, this reference is mostly still valid and very useful when examining firsthand the source code of the current version, which is at this moment the only way to really understand how these programs work. These two documents are the most complete references available for ATLAS.

[1] indeed, most of section 2 comes from this reference

The internal working of SYNTHE is described in Kurucz & Furenlid (1979) and Kurucz & Avrett (1981); however, they do not provide anything in the way of a user's manual. This document is the only reference of its kind for SYNTHE that I am aware of.

Throughout these notes I assume that you know the basics of Fortran (if you do not, look at the references in subsection 1.3). I do not assume any previous experience in VMS; whenever something about the VMS scripts in this documents is not obvious, it will be explicitly pointed out. Of course, "not obvious" being so subjective, I might have overlooked something; any user feedback that might make these notes easier to understandand is more than welcome!

1.2 Structure of this document

Section 2 documents ATLAS 9, the model synthesis part of the Kurucz suite. It shows with examples how to carry out the different tasks ATLAS can perform (creating and modifying models, and calculating radiation fields) and lists all its input commands and its general I/O structure.

Section 3 deals with SYNTHE, the spectrum synthesis program series. It breaks down a typical SYNTHE VMS command file and explains the function of each of the programs. It also documents the line identification files (such as the ones in the Kurucz CD-ROMS 1 and 15) by listing each variable along with its meaning and its Fortran format specifier.

Section 4 describes those versions of the ATLAS code that work in platforms other than VMS. External input for this section would be especially welcome.

The Appendixes contain reference material (original scripts, etc.) that is broken down or described in the main text. This material is there so that the reader can see the intact original source.

1.3 Where to get…

- The latest version of this document is included with the Kurucz distribution for Linux by L. Sbordone and P. Bonifacio (see sec. 4.1), which can be found at `http://wwwuser.oat.ts.astro.it/bonifacio/web/`; it is also available by direct request to the author by e-mail. These notes are constantly subject to change, so be sure to check for the last version every once in a while.

- All the Kurucz software and data are publicly available either for download from the Kurucz Website: `http://kurucz.harvard.edu` or as a set of CD-ROMs upon request to him.

- ATLAS and SYHTHE are written in FORTRAN IV, an obsolete dialect of Fortran. For programmers not familiar with the older dialects of this language, many of the data structures and statements used in the source code (such as Hollerith constants, arithmetic IFs…) may be unfamiliar and confusing. All of these annoying features, and many more, are explained in the superb (and free) FORTRAN tutorial by Page (1988).

2 ATLAS

ATLAS can be used to:

1. Calculate a Kurucz stellar photosphere model (see Appendix A) from scratch or modify a preexisting one;

2. calculate the radiation field associated with an existing Kurucz model, that is, a low-resolution intensity- or flux- spectrum which is useful for photometric os spectrophotometric work.

2.1 Input/output in ATLAS

Input/output in ATLAS is managed by assigning to each source of data a predetermined FORTRAN I/O unit [2]; the program accesses these units with `READ(n,...)` or `WRITE(n,...)` statements, where `n` is the unit number. Table 2.1 lists all the I/O sources, their associated FORTRAN unit numbers and the function of each.

Description	Unit	I/O	Format
ODF file	9	I	Binary
Rosseland opacity file	1	I	Text
Command-line args.	5	I	Text
Preexisting model	3	I	Text
Molecule data	2	I	Text
Main output	7	O	Text
Auxiliary info	6	O	Text

Table 1: Fortran Input/Output units used by ATLAS. The column called I/O says whether (I)nput or (O)utput is performed. The program writes its main results (new model or radiation field) to unit 7. Output from unit 6 is mainly used for debugging purposes and it is not essential. "Format" is either "Binary" (Fortran "unformatted" output) or "Text"(plain ASCII or Fortran "formatted").

ODF file. For each metallicity and microturbulent velocity (denoted by ξ), there is a file with a set of Opacity Distribution Functions (ODFs). Each ODF file contains line opacity data for several values of temperature T and electron density N_e, in the wavelength range from 8.977 to 100000 nm. ODFs come in two different resolutions: the low-resolution ones (328 divisions of the whole interval) are called BIG and the high resolution ones (1212 divisions) are called LITTLE[3]. Radiation fields must be computed by using the LITTLE ODFs; model computation can use either of them; usually the BIG tables are used, since computation time is shorter.

[2] I/O units are defined in the batch file used to run ATLAS; see sec. 2.2 for details on how this is done.

[3] The exact number of divisions and the wavelengths they occur at are given in the subroutine `BLOCKBIG` in the ATLAS source code. Despite its name, this subroutine incudes both the BIG and the LITTLE intervals.

ODF files take names like `M25ABIG4.BDF`, where `M25` tells us that the log of metallicity $[Z/Z_\odot] = -2.5$; `A` indicates α-enhanced abundances[4]; `BIG` (`LIT`)means that the BIG (LITTLE) resolution is being used; and 4 is the value of the microturbulent velocity ξ in km/s. Note that, in order to save computing time and disk space, the ODF tables are stored in machine (i.e., human-nonreadable) format.

Since ODFs are the ground upon the Kurucz models are built, it is essential that the line data they contain are as accurate and up to date as possible. The newest set of ODFs, computed by Castelli & Kurucz (2004), is based on the solar bundances of Grevesse & Sauval (1998).

Rosseland opacities. For each possible value of chemical abundance there is a Rosseland opacity table, which includes line+continuum opacities for several values of the microturbulence parameter ξ[5]. The same Rosseland file is used for the BIG and the LITTLE intervals. The tables for solar abundances and $\xi = 2$ km/s are stored within the source code of ATLAS itself [6]; in case one wants to work with different parameters, the corresponding file must be provided.

The file names for Rosseland opacity data have the format `kapxxx.ros`, where `xxx` indicates the metallicity in the same format as for ODFs. To enable such external input, the `READ KAPPA` statement (see sec. 2.3.1) must be given to ATLAS *before* any option about the starting model.

Command-line arguments. These statements control ATLAS' behavior. They are listed in sec. 2.3.

Preexisting model. If you wish to modify a preexisting model, or to calculate a model's radiation field, you must supply it here.

Molecular data. This unit is linked to a file (here called `MOLECULES.DAT`, see sec. 2.2) that contains data about several atomic and molecular species, both neutral and ionized (including free electrons), that allow to calculate their equilibrium number densities by means of the Boltzmann-Saha equation. The file gives a label that describes the atom or molecule and several coefficients that go into a polynomial form of the Saha equation. The label has the form $Z.n$; for atoms, where Z is the atomic number and n the ionization degree (01.00 is neutral H, 8.02 is O II, etc); For molecules, Z are the atomic numbers of both components from lower to upper (608 is CO, 101.01 is H_2 II, 100 is H^-, etc).

Main output. Either a new model or a radiation field (intensity or flux). The exact output can be controlled by means of the `PUNCH` statement (see sec. 2.3.3).

[4] This means that the abundance log of the alpha-process elements (O, Ne, Mg, Si, S, Ar, Ca, and Ti) is increased by +0.4.

[5] At the time of this writing (4th July 2005) the values of ξ are 0, 1, 2, 4, and 8 km/s

[6] in the definition of the external function `ROSSTAB`, used by the subroutine `TTAUP`.

Auxiliary output. Nonessential (althought very helpful) information about the intermediate steps of the program run. It allows you to follow the whole process step by step. The exact amount of output can be controlled by means of the PRINT statement (see sec. 2.3.3).

One important piece of information you can get from this output is whether the model calculation converged or not: at the beginning of each iteration, look for the two columns labeled like this:

```
 PER CENT FLUX
ERROR    DERIV
  ...      ...
```

if the error in the flux (ERROR)is less than 1% and the error in the derivative (DERIV) is less than 10%, then the computation has successfully converged. Otherwise, you will need to use more iterations.

2.2 Using ATLAS

The preferred method of invoking ATLAS under VMS is by means of a DCL script file (they have the extension .COM, so they are usually called COM files). This file sets up all necessary environment variables and I/O units and feeds the program with the copious amount of command-line arguments it requires (again, they are all described in sec. 2.3). One should write a plain text file with .com extension (say, atlasjobfile.com) with all the required commands, submit it to a batch queue for processing and, once the job is over, examine the corresponding log file (in our example, atlasjobfile.log) that is automatically created by VMS. You will know the procedure ran to the end without errors if the log file shows that the END command was executed. This does not ensure that the calculation itself successfully converged or that it is correct; it only tells that the procedure did not crash. To find out whether the procedure converged, read the description of *auxiliary output* in section 2.1.

IMPORTANT NOTE: There are two slightly different versions of the ATLAS source code: atlas9v is used to calculate models or fluxes, whereas atlas9mem is used to generate models. The former reads BIG ODF files, and the latter LITTLE ones. The difference is in the subroutine LINOP, which reads ODF files. This will be hopefully corrected in the future, but for the moment you must make sure your scripts are invoking the correct program for each case.

In the following subsections, several examples will be given for the most important operations ATLAS can perform:

2.2.1 Creating a new model

NOTE: This procedure is explained here for completeness only. It is strongly recommended to modify an already existing model from the official Kurucz grid which is close to the desired one, rather than create a completely new one.

Here we will create a model with $T = 9000$ K, $\log g = 4.5$, $\xi = 2$ and solar metallicity and write it to a file mytest.mod.

The key command that actually creates the model is the `CALCULATE` in the third line beginning from the end; it creates an initial T vs. τ_{Ross} table for a gray atmosphere; this table must later be read via `READ DECK` or `READ DECK6` commands to create the actual model.

Note the way I/O units (`forxxx`) are linked to the files required by ATLAS and the way command-line arguments are given to it after the program is invoked. *I/O unit 5 must be linked to SYS$INPUT for these arguments to be read by* ATLAS.

Each of the commands below the invocation to ATLAS is described in section 2.3.

```
! A basic procedure to generate a Kurucz model
! with Teff=9000, log g=4.5, [Z/Zo]=0, kappa=2 km/s.
! Urtzi Jauregi, 2003-01-22
!
! Sets up working directory:
$ set def [zwitter.urtzi]
! Sets up ODF table input:
$ assign [zwitter.kurucz.atlas9]p00big2.bdf  for009
! Sets up Rosseland opacity input:
$ assign [zwitter.kurucz.atlas9]kapp00.ros  for001
! Sets up molecule data input:
$ assign [zwitter.kurucz.molecules]molecules.dat  for002
! Sets up command-line parameter input:
$ assign sys$input  for005
! Sets up final model output:
$ assign p00t9000g45k2.mod for007
! Sets up auxiliaty output:
$ assign output.txt for006
!
! Runs ATLAS with required parameters:
$ run [zwitter.kurucz.atlas9]atlas9mem
TITLE  FANCY TEST
TEFF 9000 GRAVITY 4.500
FREQUENCIES 337 1 337 BIG
ITERATIONS 15
PRINT 1 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
READ KAPPA
CALCULATE STARTING 40 -4.5 0.16666666666
BEGIN
END
```

2.2.2 Modifying an existing model

Here the model calculated in section 2.2.1 will be changed to a $T_{\mathrm{eff}} = 6000$ and a metallicity $[Z/Z_{\odot}] = -1$ or one tenth of the solar value. The key commands are, as you could have imagined, ABUNDANCE SCALE to make $[Z/Z_{\odot}] = -1$ and SCALE for all the other parameters (see sec. 2.2.1 for details). Note also that we are using an ODF file corresponding to the new abundance (M10BIG2.bdf), as well as its corresponding Rosseland opacity (file KAPM10.ROS). The model to be modified is read from unit 3.

```
! Changes a model's [Z/Zo] to -1.0
! Urtzi Jauregi, 2003-01-22
!
! Sets up working dir and I/O
$ set def [zwitter.urtzi]
$ assign [zwitter.kurucz.atlas9]kapm10.ros  for001
$ assign [zwitter.kurucz.atlas9]m10big2.bdf  for009
$ assign [zwitter.kurucz.synthe]molecules.dat  for002
$ assign [zwitter.kurucz.models]ap00t4000g45k2nover.mod  for003
$ assign sys$input  for005
$ assign am10t6000g45k2nover.mod  for007
$ assign output.txt  for006
!
! Runs ATLAS version for model computations.
$ run [zwitter.kurucz.atlas9]atlas9mem
READ KAPPA
READ PUNCH
MOLECULES ON
READ MOLECULES
FREQUENCIES 337 1 337 BIG
CONVECTION OVER 1.25 0
TITLE  [0.0] VTURB=2  L/H=1.25 NOVER
ABUNDANCE SCALE 0.1
SCALE 72 -6.875 0.125 6000. 4.5
ITERATIONS 15
PRINT 1 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
END
```

2.2.3 Calculating the radiation field

Next, we will calculate the surface intensity of our model at 5 different stellar latitudes starting from the center of the stellar disk ($\cos\theta = 1$) in the $8500 - 8750$ Å wavelength range, at intervals of 1 Å. The key command is SURFACE INTENSITY, which takes the number of $\cos\theta$ angles for which the calculation will be performed, followed by the angles themselves. Note that for radiation field computations the iteration number *must* be 1.

```
! Calculates intensities at 5 angles
! Urtzi Jauregi, 2003-01-22
!
! Sets up working dir and I/O
$ set def [zwitter.urtzi]
$ assign [zwitter.kurucz.atlas9]p00lit2.bdf   for009
$ assign [zwitter.kurucz.synthe]molecules.dat  for002
$ assign ap00t4250g45k2nover.mod  for003
$ assign sys$input  for005
$ assign [zwitter.urtzi]fp00t4250g45k2nover.mod  for007
!
! Runs ATLAS version for radiation fields.
$ run [zwitter.kurucz.atlas9]atlas9v
SURFACE INTENSITY 5  1., .8, .6, .4, .2
ITERATIONS 1
PRINT 2
PUNCH 2
READ PUNCH
MOLECULES ON
READ MOLECULES
VTURB 2.0E+5
BEGIN
END
```

2.3 The ATLAS control cards

The commands (“cards”, for historical reasons) read by ATLAS describe the operations the program will perform and the physical variables that it needs to perform them. They are read by the soubroutine `READIN`, which turns each text command given by the user to a string of numbers[7], and passes these numbers to the relevant section of the program.

Following Castelli (1988), the cards can be divided into 4 cathegories:

1. Cards to compute a new model;
2. cards to control various physical processes;
3. cards to control the output;
4. other cards.

2.3.1 Cards to compute a new model

- `GRAVITY` *n*: *n* is the log gravity in cm/s^2.

[7] This is a relic from the days when FORTRAN lacked a CHARACTER data type, so text strings could not be passed to program units.

- `TEFF` n: n is the effective temperature in K.
- To start a model:
  - `CALCULATE` n_1 n_2 n_3: Creates a T vs. $\log_{10}\tau_{\rm Ross}$ distribution corresponding to a gray atmosphere with a $T_{\rm eff}$ and a $\log g$ given by the cards above. This distribution has n_1 depth points, a value n_2 in the first point and an increment of n_3 between points. It is suggested that the number of depth points n_1 be large (>40) for reasons of numerical stability. An already existing model can be evaluated at a different set of depth points by means of the `SCALE` card (see below).
  - `READ START` n: Reads n pairs $\tau_{\rm Ross}$, T (like the ones created by `CALCULATE`), which must immediatelly follow.
  - `READ DECK` n: Reads n lines with the magnitudes listed in Table 2. See `READ DECK6` for comparison.

RHOX	T	P	XNE	ABROSS	PRAD	VTURB
Mass depth variable (g/cm^2)	Temperature (K)	Gas density (dyn/cm^2)	Electron density (cm^{-3})	Rosseland absorption coeff. (cm^2/g)	Pressure due to radiation (dyn/s^2)	Turbulence velocity (Km/s)
$M(i)=\int_0^i \rho(x)\,dx$	$T(i)$	$P_{\rm gas}(i)$	$n_e(i)$	$\kappa(i)$	$P_{\rm rad}(i)$	$\xi(i)$

Table 2: Physical magnitudes given in an ATLAS 9 as a function of depth.

  - `READ DECK6` n: As `READ DECK`, but `PRAD` is substituted by `ACCRAD`, the acceleration due to radiation pressure. See `READ DECK`.

- `SCALE` n_1 n_2 n_3 n_4 n_5: Re-scales a model to a number of n_1 depth points, with a $\tau_{\rm Ross}$ of 10^{n_2} at the first point and with a spacing of 10^{n_3}. n_4 is the new effective temperature and n_5 is the new gravity. If you want a new model, Castelli (1988) recommends to modify an existing one with `SCALE` instead of calculating one from scratch with `CALCULATE`.
- `FREQUENCIES` n_1 n_2 n_3 name: Reads n_1 frequencies starting from n_2 and ending in n_3. The variable *name* can be either BIG or LITTLE. As said before, BIG is recommended for model calculations and LITTLE for radiation fields. [8]
- `ITERATIONS` n Number of iterations ATLAS will go through to calculate a model. Usually $n = 15$ for model computations; in order to compute radiation fields, n *must* be equal to 1.
- `ABUNDANCE`: Changes the chemical abundances. It accepts two arguments:

[8] For unblanketed models, `READ FREQUENCIES` n_1 n_2 n_3 or `WAVELENGTH` n_1 n_2 n_3 can be used. However, nobody computes unblanketed models anymore because ATLAS 9 does not work without lines, so this options are in practice obsolete. They are described here for those who have to work with scripts written by others.

- SCALE n: Multiplies all abundances except H and He by n. Note that n is a scale factor, not a logarithm.
- CHANGE n_1 n_2, n_3 n_4, ...: Changes the individual abundance of each element. Arguments are comma-separated pairs of numbers, where the first one is the atomic number of the element of interest, and the second one is the abundance relative to the total; a negative abundance is interpreted as a logarithm. For example, `ABUNDANCE CHANGE 1 0.8, 2 0` would change H abundance to 0.8, He to 0.2 and O to 10^{-4}.

- READ KAPPA: Reads a Rosseland opacity table from an external file KAPxxx.ROS, instead of from the array stored in its own source code. It must be used *before* any other card about the new model.

- READ PUNCH: Reads a preexisting Kurucz model from a separate external file, instead of from the same file that holds the rest of the commands.

- BEGIN: Begins the calculation.

- END: Calls VMS routine EXIT and ends the procedure.

2.3.2 Cards to control various physical processes

- OPACITY. States which opacity sources (both line and continuum) will be used in the calculation. ATLAS accepts up to 20 different sources, 15 of which are currently active. The possible arguments are:
  - ON name1, name2, ...: Turns on the stated sources. The names used as arguments are given in Castelli (1988).
  - OFF name1, name2, ...: Turns off the stated sources. The names used as arguments are given in Castelli (1988).
  - IFOP switches...: Accepts a row of 20 comma-separated switches whose value must be 1 or 0, turning on or off each opacity source.

- CORRECTION ON/OFF: Turns on/off the temperature correction mechanism used in Kurucz model synthesis (see Kurucz (1970)). Since such correction is not needed to compute a radiation field, turning it off can save computer time.

- PRESSURE ON/OFF: If on , pressure and number densities are computed.

- SURFACE. Possible arguments:
  - FLUX: Computes radiation flux only at surface (by default, it is computed at every depth; it can be accessed by using the PRINT and PUNCH cards).
  - INTENSITY n_1 n_2 n_3 n_4 ...: Computes surface radiation intensity at n_1 latitude points at angle θ_i, where $n_i = \cos\theta_i$.

  - `OFF`: Calculates radiation field at every depth.

- `SCATTERING ON` / `OFF`: Includes or excludes scattering source function as part of the opacity.

- `CONVECTION`: Includes convection in energy transport calculations (mixing-length model)[9].

  - `ON` *n*: *n* is the desired mixing-length parameter (ratio of mixing-length to pressure scale-height).
  - `OFF`: Disables computation of the convective flux.
  - `OVER` n_1 n_2: Allows overshooting to be included in the computation. n_1 is again the mxing-length parameter. n_2 is the overshooting weight parameter W defined in Castelli, Gratton & Kurucz (1997); a value of 0 disables overshooting[10] (again, as with `CONVECTION OFF`, you *must* do this in order to compute non-overshooting models; the use of just `CONVECTION ON` will apply $W = 1$.

- `MOLECULES ON/OFF`: Allows the user to input molecular partition functions to find the populations of the different ionized molecular especies.

- `TURBULENCE`: Introduces a turbulent pressure that will be included in the total pressure calculations.

  - `ON` n_1 n_2 n_3 n_4: Introduces a turbulent velocity v_t related to the turbulent pressure P_t such that

$$P_t = \frac{1}{2\rho(n_1 \times \rho^{n_2} + n_3 \times v_s/10^5 + n_4)^2} \tag{1}$$

  - `OFF`: Disables turbulent pressure.

- `LTE` / `NLTE`: Enables NLTE calculations for H and H^- continua.

2.3.3 Cards to control the output

- `PRINT` *n*: Controls output to unit 6 (auxiliary output). In model calculations, which run through *n* iterations, a set of *n* switches with the value "1" or "0" is given; the program will write all the auxiliary output for each iteration that has a "1". See examples in sec. 2.2. Normally, only the first and the last iteration are of interest.

  In radiation field calculations, *n* is a single index that goes from 0 to 4, with 0 meaning no output and 4 an extreme verboseness. Default is 2.

- `PUNCH` *n*: Controls output to unit 7 (final model or flux). Arguments are the same as for `PRINT`, the user getting a Kurucz model for each iteraton with a "1". Only the last iteration is important.

[9] An exhaustive analysis of the role of convection on ATLAS model atmospheres can be found in Castelli, Gratton & Kurucz (1997).

[10] As non-overshooting models are usually more accurate, this is the recommended choice.

2.3.4 Other cards

- `READ DEPARTURE` n_1: Reads H and H$^-$ departure coefficients for NLTE model calculation. It is not recommended, because these models are not realistic.
- `CALL`: Calls subroutine `DUMMYR`, which must be supplied by the user to enable additional operations or data sources.
- `TITLE string`: Sets the name of the model. `string` is an arbitrary text string with a maximum of 74 characters.
- `CHANGE` n_1 n_2, n_3, n_4 ...: Maps a preexisting model onto n_1 ponts at depths n_2, n_3 ...

2.4 Miscellaneous remarks

Despite its power the Kurucz suite is quite quirky; there are many non-documented features, glitches and bugs that the prospective user must be aware of. This section tries to list them. Any reader input that adds to this section is especially welcome!

2.4.1 Integration points for the radiation field

As said before, the physics behind Kurucz model- and flux- computation is explained in detail in Kurucz (1970), so the reader is submitted there for details. However, there is a potentially important remark for users and developers to be made here. In ATLAS, the radiation field is computed by using the method first detailed in Kurucz (1969) to calculate the integrals for the mean intensity and flux:

$$J_{\nu_i} = \frac{1}{2}\int_0^\infty S(t)E_1(|t-\tau|)\,dt = \sum_{j=1}^{N} \Lambda_{ij} S(\tau_j) \tag{2}$$

$$H_{\nu_i} = \frac{1}{2}\int_0^\infty S(t)E_2(|t-\tau|)\,dt = \sum_{j=1}^{N} \Phi_{ij} S(\tau_j). \tag{3}$$

As you can see, the integration is carried out by summing over the index j of the matrices Λ_{ij} and Φ_{ij}. Each value of j represents a given layer of the photosphere, with an optical depth given by τ_j.

The integration points actually used by `ATLAS 9` (i.e.the values of the τ_j) are *not* the ones that are listed in Castelli (1988) for `ATLAS 8`; in the original Kurucz code, the real values can be found in the source code of ATLAS, in the array `XTAU8` of subroutine `JOSH` (in the latest version of the code, they are the lines 16730-16734 of both `atlas9v.for` and `atlas9mem.for`).

3 SYNTHE

This section will cover only the very basics of using SYNTHE to generate spectra; subsection 3.3 explains the structure of a SYNTHE input file and subsection 3.4 lists all the somewhat obscure parameters used for line identification. Some information on the subprograms can be found in the file `aareadme.tab` in the Kurucz Web site and in the Kurucz CD-ROM no. 18.

3.1 Introduction

The SYNTHE program series[11] takes as input a model atmosphere generated with ATLAS and outputs either

1. the radiation intensity at several values of the angle θ (up to 20) or
2. the specific flux (intensity integrated over all the stellar disk)

as a function of wavelength. This means that you get either several intensity spectra or a single flux spectrum. Using the tools in SYNTHE, these spectra can be computed at a very high initial resolution and then degraded to account for rotational, macroturbulent and instrumental broadening. This is what we will do in the next subsection.

3.2 Adding command headers to model files

In order for SYNTHE to work, one can not use Kurucz models as they come out of ATLAS; in addition to writing a SYNTHE COM file as the one below, one must insert an ASCII header with additional commands at the top of each *model* file (that is, before the line beginning with `TEFF...`). These commands tell SYNTHE whether to compute intensity or flux spectra, and the positions $\cos\theta$ in the stellar disk at which intensities must be computed. *If this header is not included, the procedure will invariably crash.*

Example headers for both intensity and flux spectra are:

- For intensity spectra:

```
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
```

- For flux spectra:

```
SURFACE FLUX
ITERATIONS 1 PRINT 2 PUNCH 2
CORRECTION OFF
```

[11] SYNTHE is the name of both the whole series of programs and of one of the subprograms included in it; throughout this appendix, the term SYNTHE will be reserved for the series. Whenever it becomes necessary to refer to the subprogram, an expression like "the program SYNTHE" will be used.

```
PRESSURE OFF
READ MOLECULES
MOLECULES ON
```

Since SYNTHE uses the same input routines as ATLAS, the commands in the headers are all described in sec. 2.3. The only remark is that for SYNTHE the iteration number *must* always be 1.

In addition to this, there seems to be a bug in ROTATE that affects the way intensities and fluxes are calculated for rotating stars; see 3.5.1 for more information.

3.3 Sample COM file

This section describes step by step a fairly standard DCL script file ("COM file") for SYNTHE that was used on an Alpha workstation running VMS in the University of Ljubljana. The file is reproduced in its original form in Appendix B. It generates a spectrum in the wavelength range $[8500, 8750]$Å with a resolution $\lambda/\Delta\lambda = 500000$ which is later degraded by rotational, turbulent and instrumental broadening to an effective resolution $\lambda/\Delta\lambda = 20000$.

The input model atmosphere, which is assumed to include one of the extra headers in sec. 3.2 is read from a file called `ap00t6000g45k2nover.mod`. The model file naming scheme is described in detail in the Kurucz website; here it will only be said that

- `pxx` is the base 10 logarithm of the abundance, relative to solar (p means "plus"; metal-poor stars would obviously have have `mxx`; `p00` means solar abundance);
- `txxxxx` is the effective temperature in K;
- `gxx` is the logarithm of surface gravity in cm/s^2;
- `kx` is the microturbulent speed in Km/s;
- `nover` states that the overshooting approximation was not included in the model calculation.

Thus, `ap00t6000g45k2nover` describes a Kurucz model of a star with solar metallicity, with an effective temperature of 6000 K, $\log g = 4.5$ and a microturbulent velocity of 2 km/s which has been computed without resorting to the overshooting approximation.

Some of the I/O operations in SYNTHE (including some important ones) are not explicitly stated in this COM file; the only way of keeping track of them is by direct inspection of the source code of each subprogram. They have been also listed here.

A note on file format: SYNTHE uses two types of input/output files: The scratch files used for internal operations use a system-dependent, human-nonreadable binary form (the FORTRAN ”unformatted" file type) in order to maximize I/O speed and reduce disk usage; they have the extension .dat. The human-readable, plain ASCII files use the extension .asc (with the notable exception of `output.dat`, see below).

```
$mount/media=cdrom/undefined=(fixed:none:16) $5$dka400 cdrom24 cdrom24
```

The COM file begins by mounting the Kurucz CD-ROM 24 in case it is in the unit. This CD contains the TiO line list and will be later required if TiO is included in the spectrum computation.

```
$ SET DEF [zwitter.CALC]
```

Sets up the working directory.

```
$ ASSIGN output.dat for006
```

Assigns FORTRAN I/O unit 6 to an auxiliary ASCII output file to which many partial computations will be written, very much like the unit 6 output of ATLAS. This file will be written to (or, under VAX, a new version of the same file will be created) whenever a SYNTHE subprogram executes a WRITE(6,...) statement.

```
$ ASSIGN MODELS:ap00t6000g45k2nover.mod for005
```

Assigns I/O unit 5 to our model atmosphere file stored at the directory [MODELS]. Subsequent FORTRAN programs will access the model via READ(5,...) statements.

```
$ assign SYNTHE:molecules.dat for002
$ assign SYNTHE:pfiron.dat for004
```

Assigns FORTRAN I/O unit 4 to file `pfiron.dat`, which contains the partition functions for the elements of the iron group, from Ca to Cu.

```
$ ASSIGN SYNTHE:continua.dat FOR017
```

Assigns FORTRAN I/O unit 17 to file `continua.dat`, which contains a list of photoionization edges for the electronic states of those elements contributing to continuum opacity:

```
$ ASSIGN XNFp00t6000g45k2nover.dat for010
```

Assigns FORTRAN I/O unit 10 to file `XNFp00t6000g45k2nover.dat`, which will be used as output by the program XNFPELSYN.

```
$ RUN SYNTHE:XNFPELSYN
```

Executes XNFPELSYN, the first program in the series. XNFPELSYN pretabulates continuum opacities and number densities for different chemical elements and writes them to the file `XNFp00t6000g45k2nover.dat`. The elements entering the calculation can be found in the source code of XNFPELSYN: they are the first argument in the invocations to subroutine `POPS`, and have the form $Z.i$, where Z is the atomic number and i is the ionization stage (1.00 is the code for neutral H, 26.01 is the singly ionized iron, etc.).

```
$ ASSIGN SYS$INPUT FOR005
```

Assigns unit 5 to the system standard input. This is where the command-line arguments for the program SYNBEG will be read from.

```
$RUN SYNTHE:SYNBEG
AIR 850.0 875.0 500000. 0.0 0 300 .001 00
AIRorVAC WLBEG WLEND RESOLU TURBV IFNLTE LINOUT CUTOFF NREAD
```

Invokes the second program, SYNBEG. It reads the fundamental parameters of the process: wavelength range, resolution of the final spectrum *before degrading*, whether unclassified (predicted) lines should be included, etc., and writes them to I/O unit 93.

The command line parameters of SYNBEG are explained in the comments of its source code. The most important ones are:

1. `AIRorVAC`: Whether atmospheric lines should be included (AIR) or not (VAC).
2. `WLBEG`, `WLEND`: Lower and upper wavelength limits of the desired spectrum in nanometers.
3. `RESOLU`: The resolution $\lambda/\Delta\lambda$ for the final *non-degraded* spectrum (i.e., before rotational, macroturbulent or instrumental broadening are applied).
4. `IFNLTE`: If 0, the model was calculated with Local Thermodynamic Equilibrium (see 2.3). If 1, it is NLTE.
5. `CUTOFF`: The fraction of continuum opacity below which line wings are no longer computed.

```
$ASSIGN LINES:GF0300.100 FOR011
$RUN SYNTHE:RGFALLTEST
```

Runs the third program, RGFALLTEST. This program adds line information and line opacity data by reading the adequate opacity file from I/O unit 11 (one file per run of the program). The name of the file is `GFxxxx.yyy`, where `xxxx` is the upper wavelength limit of the file (in nm) and `yyy` is its wavelength range, also in nm. Thus, `GF0300.100` has data for lines between 200 and 300 nm. WARNING: There is an important exception. The file `GF1200.100` contains lines between 800 and 1200 nm; therefore it should be better called `GF1200.400`.

The line opacity data are sent to unit 12 (and the line identification to unit 14) if the calculation is LTE; it goes to unit 19 (line identification to unit 20) if not. Check the comments on the source code for details. This program can be run as many times as needed to include all the necessary line tables. In our example, this is repeated for other 5 intervals; see Appendix for details.

```
$assign MOLECULES:c2ax.bin FOR011
$run SYNTHE:Rmolec
```

The program RMOLEC is the homologue of RGFALLTEST for diatomic molecules. The `xxyy.bin` files referred in the line before program invocations contain line data for the transition yy of the molecule xx. RMOLEC writes line opacity to unit 12 and line identifications to unit 14.

Again, this can be repeated as many times as desired to include all needed molecular lines.

```
$ ASSIGN CD24:schwenke.bin for011
$run SYNTHE:Rmolec
```

This is a special invocation of RMOLEC. Here it is run for the TiO line lists on Kurucz CD 24, which was mounted at the beginning of the COM file. For high temperatures (> 4500 K), they can be ignored; this will speed up the whole procedure *enormously*.

```
$ ASSIGN XNFp00t6000g45k2nover.dat for010
$ RUN SYNTHE:SYNTHE
```

Here the file `XNFp00t6000g45k2nover.dat` created by XNFPELSYN is set as the input for the program SYNTHE, which computes line opacity data based on the fundamental parameters of the model (the ones written by SYNBEG to unit 93) and writes them to I/O unit 93.

```
$ ASSIGN MODELS:ap00t6000g45k2nover.mod for005
$ASSIGN i4600-9000.dat FOR007
$ASSIGN SYS$INPUT FOR025
$assign synthe:molecules.dat for002
$RUN synthe:SPECTRV
0.0       0.        1.        0.        0.        0.       0.       0.
0.
RHOXJ     R1        R101      PH1       PC1       PSI1     PRDDOP   PRDPOW
```

The program SPECTRV reads from unit 9 the file written by the program SYNTHE and computes the continuum opacities and the overall synthetic spectrum (intensities at 17 angles); this spectrum is then written to file `i4600-9000.dat` at I/O unit 7.

This is the final result if no spectrum degrading is required.

```
$ASSIGN i4600-9000.dat FOR001
$assign sys$input for005
$ ASSIGN f85008750v0.dat rot1
$ ASSIGN f85008750v20.dat rot2
$ ASSIGN f85008750v100.dat rot3
$run synthe:rotate
   3
0.        20.        100.
```

The program ROTATE handles rotational broadening of spectral lines. It also integrates the intensity data given by SPECTRV to compute total flux. The arguments are the number of values of the projected rotational velocity $v\sin i$ (first row) and the velocities themselves (second row). Here we use 3 values; ROTATE accepts up to 5 values; if more are desired, this whole clock of commands has to be repeated for the extra velocities, in groups of 5 or less.

We see that the full-resolution spectrum at the file `i4600-9000.dat` mentioned above is read from I/O unit 1 and the command-line parameters from unit 5. A file is created for each rotational speed, and the program loops 5 times (the first line after program invocation), each time performing the broadening for one of the projected velocities $v\sin i$ shown in the next line. Rotational line broadening is carried out by convolving the flux data with the rotational profile of the star to obtain the broadened profile; see Gray (1995) for details on the convolution method.

ROTATE writes its output to I/O unit 9.

```
$ assign sys$input for005
$ assign f85008750v0.dat for021
$ assign f85008750v0-r20000p00t6000g45k2nover.dat for022
$ run synthe:BROADEN
GAUSSIAN 20000. RESOLUTION
```

The program BROADEN broadens the spectrum to simulate the effect of macro turbulence and/or of an instrumental degrading. It uses the output from ROTATE as input (at I/O unit 21) and its output is a binary file to unit 22. The command-line parameters for BROADEN are, in this order:

1. Type of the broadening profile (only the first 4 letters are relevant): Can be either of:

   (a) `MACR` (macroturbulent);

   (b) `GAUS` (gaussian);

   (c) `SINX` ($\sin x/x$ -type);

   (d) `RECT` (rectangular box); or

   (e) `PROF` (pretabulated custom profile, which must be included in the COM file immediately after this line).

2. Full width at half-maximum (FWHM) of the broadening profile. It gives the resolution of the output. The units of this number are given by the next parameter.
3. Units for the FWHM. Can be either of:
   (a) `KM` (FWHM given in km/s);
   (b) `RESO` ($\lambda/\Delta\lambda$, so the profiles will have FWHM $= \frac{c}{\Delta\lambda/\lambda}$;
   (c) `PM` (picometers; makes FWHM $= \frac{c}{1000\times\lambda_c} \times$ `PM`, where $\lambda_c = (\lambda_{\mathrm{min}} + \lambda_{\mathrm{max}})/2$.

```
$ assign f85008750v0-r20000p00t6000g45k2nover.dat for001
$ assign f85008750v0-r20000p00t6000g45k2nover.asc for002
$ run synthe:CONVERFSYNNMTOA
$ delete/noconf f85008750v0-r20000p00t6000g45k2nover.dat.*
```

As the binary data output from BROADEN is not readable by humans, they must be converted to formatted output before they can be used. This is done by invoking the program CONVERFSYNNMTOA, which is invoked for each rotational speed in turn.

This program can also write line identification data; this can be avoided by simply commenting out the relevant code (the last DO loop in the main program, which contains the WRITE(2,8) statement).

CONVERFSYNNMTOA reads input from unit 1; it writes the ASCII spectra and the line identification data to unit 2[12]

```
$ copy  f85008750v0-r20000p00t6000g45k2nover.asc [zwitter.calc]
$ copy  f85008750v20-r20000p00t6000g45k2nover.asc [zwitter.calc]
$ copy  f85008750v100-r20000p00t6000g45k2nover.asc [zwitter.calc]
$ DEL/NOCONF *.*.*
```

Final cleanup. Copies ASCII output files from CONVERFSYNNMTOA to a personal directory and deletes everything else. It might be necessary to delete this last line if the intermediate results must be kept for a detailed analysis or for debugging/testing a new version of the programs.

3.4 Line data

The Kurucz line lists included in the CD-ROM 1 and 15 span about 58 million lines. The whole list can be found at the Kurucz website, in the subdirectory LINELISTS/. There is also useful information in the source code of the program RGFALLEST. In this section, the fields listed for each line are shown; where possible, the variable names used by SYNTHE and their associated

[12] If you want the line data, but in a different file, just change the unit number in the statement mentioned above (but do not use the number 1!!) and add a suitable link in the COM file, then recompile the program.

FORTRAN format specifications (between brackets), have been included[13]. Whenever there was a discrepancy between the formats on the web site and in the program source code, the latter was preferred.

3.4.1 Atomic line data

The data fields for atomic lines are, in this order:

1. Wavelength of the line (nm): WL (F10.4).
2. $\log gf$: GFLOG (F7.3).
3. Element code (i.e., 01.00 for neutral hydrogen, 11.01 for singly ionized Na, etc.): CODE (F6.2).
4. First (lower) energy level (cm^{-1}): E (F12.3).
5. Total angular momentum $J = L + S$ for the first (lower) level: XJ (F5.1).
6. Configuration for the first (lower) level in spectroscopic notation: LABEL (A10).
7. Second (higher) energy level (cm^{-1}): EP (F12.3).
8. Total quantum angular momentum $J = L + S$ for the second (higher) level: XJP (F5.1).
9. Configuration for the second (higher) level in spectroscopic notation: LABELP (A10).
10. $\log g_{\mathrm{R}}$, where g_{R} is the radiative damping constant: GAMMAR (F6.2).
11. $\log g_{\mathrm{S}}$, where g_{S} is the Stark damping constant: GAMMAS (F6.2).
12. $\log g_{\mathrm{VdW}}$, where g_{VdW} is the Van der Waals damping constant per hydrogen atom for hydrogen at 10000 K: GAMMAW (F6.2).
13. Reference field where the bibliographical source for the line are written. These codes are explained in the files `gfall.ref` and `gfelem.ref` in the subdirectory `/linelists/lines` of the Kurucz website: REF (A4).
14. Non-LTE departure coefficient for the lower level: NBLO (I2).
15. Non-LTE departure coefficient for the upper level: NBUP (I2).
16. Isotope numbers for up to 2 components. NOTE: There are two complementary variables, that add up to $\log gf$to account for the effect of isotopes: the fractional isotopic abundances (17) and (19). ISO1 (I3).

[13] This information has been taken from either the introductory page to subdirectory `LINELISTS/`, cited above, or by direct inspection of the SYNTHE source code (RGFALLTEST for atomic lines and RMOLEC for molecules) and of the line list files.

17. Hyperfine component log fractional strength. See (16). X1 (F7.3).
18. Isotope number for the second element, if the line comes from a diatomic substance.In this case, there is no hyperfine term: ISO2 (I3).
19. Log of isotopic abundance fraction. See (16): X2 (F7.3).
20. Hyperfine shift for first level in mK; to be added to the energy of the first level. (I5).
21. Hyperfine shift for second level in mK; to be added to the energy of the second level. (I5).
22. Hyperfine f (oscillator strength) for the first level. (I1).
23. Note on character of hyperfine data for first level: `z` means "none", `?` means "guessed". (A1).
24. Hyperfine f for the second level. (I1).
25. Note on character of hyperfine data for second level: `z` means "none", `?` means "guessed". (A1).
26. 1-digit code, sometimes for line strength classes. (I1).
27. 3-character code such as AUT for autoionizing. (A3).
28. Landé G factor for first level multiplied by 1000: GLANDE (I5).
29. Landé G factor for second level multiplied by 1000. GLANDEP (I5).
30. Isotope shift of wavelength (mÅ): ISOSHIFT (I6).

For example, one of the three lines of the Ca II triplet (see Appendix B), at $\lambda = 8542.091$ Å, will have the following profile (the information has been split after field no. (9) for clarity; in the actual line files, this whole entry is a single line):

```
854.2091 -0.362 20.01 13710.880 2.5 3d 2D 25414.400 1.5 4p 2P
8.20 -5.55 -7.80BWL 2 3 0 0.000 0 0.000 0 0 5 1200 1334
```

3.4.2 Molecular line data

For molecular lines, the line data format is:

1. Wavelength of the line (nm): WL (F10.4).
2. $\log gf$: GFLOG (F7.3).

3. Total quantum angular momentum $J = L+S$ of the lower level: XJ (F5.1).
4. Energy of the lower level (cm^{-1}): E (F12.3).
5. Total quantum angular momentum $J = L+S$ for the higher level: XJP (F5.1).
6. Higher energy level (cm^{-1}): EP (F12.3).
7. Code of the molecule, that is, the list of atomic numbers for each component (i.e., 814 for SiO) and the label for the exact transition involved: CODE (F9.2).
8. Label for the lower level: LABEL(1) (A8).
9. Label for the higher level: LABELP(1) (A8).
10. Isotope number: ISO (I2).

A sample line for SiO would be :

```
350.0288 -0.121 69.0-32654.280 70.0 -61215.190 814X28 E10 29
```

3.5 Miscellaneous remarks

This subsection lists various quirks and bugs of SYNTHE. As its ATLAS homologue, any reader input for this section is especialy welcome.

3.5.1 ROTATE and fluxes

The card `SURFACE FLUX` makes SYNTHE crash when ROTATE is in the SYNTHE COM file. According to a private communication by F. Castelli, this is probably due to a bug in ROTATE. To get a flux spectrum for a rotating star, one must use `SURFACE INTENSITY` in the model file. In other words:

1. `SURFACE FLUX` without ROTATE = Flux spectrum of a non-rotating star.
2. `SURFACE FLUX` with ROTATE = program crashes
3. `SURFACE INTENSITY` without ROTATE = Intensity spectrum at several angles for a non-rotating star.
4. `SURFACE INTENSITY` with ROTATE = Flux spectrum of a rotating star.

Note that you can also get flux spectra of non-rotating stars with `SURFACE INTENSITY` and $v\sin i = 0$. Therefore, it is recommended to always use `SURFACE INTENSITY` instead of `SURFACE FLUX`.

4 Non-VMS distributions of ATLAS and SYNTHE

There are two non-VMS distributions of the Kurucz programs that I am aware of. Both are meant for UNIX:

4.1 The Sbordone-Bonifacio distribution

This distribution, detailed in Sbordone *et al.* (2004), includes both ATLAS 9 [14] and SYNTHE. The authors tweaked the original source code so that it successfully compiles with the Intel Fortran Compiler for Linux[15], which accepts pre-Fortran 66 syntax. They also substituted the VMS scripts for csh ones.

As it does only minimal and easy-to-keep track-of changes to the original code, the this distribution gives very similar results under Linux to those of the original code under VMS; therefore, it as suitable for serious research work as the original code itself.

4.2 The Lester distribution

This is a complete rewrite of both ATLAS 9 and SYNTHE in Fortran 90 by John B. Lester of the University of Toronto. There is no recent paper that details its usage, but some (very outdated) references exist by Jeffrey *et al.* (1994) and Jeffrey (1994).

Besides a completely revamped and abundantly commented source code, this distribution includes important changes in the physics of spectrum synthesis: its version of SYNTHE includes stellar pulsation as well as rotation.

Lester wrote his distribution in strict Standard Fortran 90, which means that it should compile and run without problems on any machine with a suitable compiler. However, I have not been able to reproduce Kurucz's spectra. Any report of successful work with the Lester distribution would be extremely welcome!

5 TO DO

This document still has a long way to go before being a comprehensive user's guide to the Kurucz program suite. The items on the following list have priority and will be added in later versions.

User feedback. These notes are meant to be a practical resource for the community of users of the Kurucz code. I want them to be as useful as possible, and for that it is important to have as much user feedback as possible. If you think something in these notes should be included, corrected, clarified and/or omitted, don't hesitate to contact me!

- Document the non-VMS distributions better.
- Document ATLAS 12 with examples, etc.

[14] The next version of the distribution, which is expected very soon, will also include a port of ATLAS 12.

[15] Freely available for non-commercial use at
`http://www.intel.com/cd/software/products/asmo-na/eng/219758.htm`

- Explain the data format of the ODF, the Rosseland opacity files, etc. Do the same for all the SYNTHE data files.
- Make a table for SYNTHE like Table 2.1 for ATLAS, listing all the I/O units and their functions.
- Include a pointer to a tutorial of VMS for beginners. Will go in Section "Where to get...". Also, include an "VMS-To-Linux" document.

6 Thanks to...

- Fiorella Castelli, from the Osservatorio Astronomico di Trieste, for sharing her remarkable knowledge of the intrincacies of the Kurucz programs, and for the patience with which she answered my endless stream of questions.
- Rosanna Sordo, from the Università di Padova, for pointing out some important mistakes in the early version of this document.

A A complete ATLAS model

Here you can see a typical Kurucz model of a star with $T_{\text{eff}} = 6000$ K, $\log g = 1.0$ and solar metallicity. The model was calculated with convective transport with a mixing-length ratio of 1.25 and without overshooting. Turbulent effects were neglected.

Without going into details (see Castelli (1988) for a complete description), you can see that the model is divided in several sections:

- A 4-line header with all the essential parameters (T_{eff}, $\log g$, opacity sources used ...).
- A block with lines beginning with `ABUNDANCE...`, where chemical abundances are given both globally (via the `ABUNDANCE SCALE` card), and for each invidual element (with `ABUNDANCE CHANGE`).
- A list of several physical magnitudes at several depths in the atmosphere See the descriptions of `READ DECK` or `READ DECK6` in section 2.3.1.
- A last line of the form `PRADK` x, where x is the surface radiation pressure in dyn/cm^2.

All the control statements used here are described in sec. 2.3.

```
TEFF   6000.  GRAVITY 1.00000 LTE
TITLE  [0.0] VTURB=2  L/H=1.25 NOVER
 OPACITY IFOP 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0
 CONVECTION ON   1.25 TURBULENCE OFF  0.00  0.00  0.00  0.00
ABUNDANCE SCALE   1.00000 ABUNDANCE CHANGE 1 0.91100 2 0.08900
```

```
 ABUNDANCE CHANGE  3 -10.88  4 -10.89  5  -9.44  6  -3.48  7  -3.99  8  -3.11
 ABUNDANCE CHANGE  9  -7.48 10  -3.95 11  -5.71 12  -4.46 13  -5.57 14  -4.49
 ABUNDANCE CHANGE 15  -6.59 16  -4.83 17  -6.54 18  -5.48 19  -6.82 20  -5.68
 ABUNDANCE CHANGE 21  -8.94 22  -7.05 23  -8.04 24  -6.37 25  -6.65 26  -4.37
 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  -8.69 35  -9.41 36  -8.81 37  -9.44 38  -9.14
 ABUNDANCE CHANGE 39  -9.80 40  -9.54 41 -10.62 42 -10.12 43 -20.00 44 -10.20
 ABUNDANCE CHANGE 45 -10.92 46 -10.35 47 -11.10 48 -10.18 49 -10.58 50 -10.04
 ABUNDANCE CHANGE 51 -11.04 52  -9.80 53 -10.53 54  -9.81 55 -10.92 56  -9.91
 ABUNDANCE CHANGE 57 -10.82 58 -10.49 59 -11.33 60 -10.54 61 -20.00 62 -11.04
 ABUNDANCE CHANGE 63 -11.53 64 -10.92 65 -11.94 66 -10.94 67 -11.78 68 -11.11
 ABUNDANCE CHANGE 69 -12.04 70 -10.96 71 -11.28 72 -11.16 73 -11.91 74 -10.93
 ABUNDANCE CHANGE 75 -11.77 76 -10.59 77 -10.69 78 -10.24 79 -11.03 80 -10.95
 ABUNDANCE CHANGE 81 -11.14 82 -10.19 83 -11.33 84 -20.00 85 -20.00 86 -20.00
 ABUNDANCE CHANGE 87 -20.00 88 -20.00 89 -20.00 90 -11.92 91 -20.00 92 -12.51
 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
READ DECK6 72 RHOX,T,P,XNE,ABROSS,ACCRAD,VTURB
 1.32487281E-03   3682.6 1.324E-02 4.419E+06 1.007E-04 6.975E-03 2.000E+05 0.000E+00 0
 1.77096267E-03   3693.7 1.770E-02 5.681E+06 9.875E-05 7.122E-03 2.000E+05 0.000E+00 0
 2.37588384E-03   3706.4 2.374E-02 7.361E+06 9.734E-05 7.180E-03 2.000E+05 0.000E+00 0
 3.19198623E-03   3721.2 3.190E-02 9.605E+06 9.648E-05 7.203E-03 2.000E+05 0.000E+00 0
 4.28776199E-03   3738.1 4.285E-02 1.259E+07 9.611E-05 7.285E-03 2.000E+05 0.000E+00 0
 5.75040617E-03   3757.3 5.746E-02 1.655E+07 9.625E-05 7.297E-03 2.000E+05 0.000E+00 0
 7.69327359E-03   3778.7 7.688E-02 2.179E+07 9.683E-05 7.227E-03 2.000E+05 0.000E+00 0
 1.02625864E-02   3802.0 1.026E-01 2.872E+07 9.782E-05 7.078E-03 2.000E+05 0.000E+00 0
 1.36497086E-02   3825.7 1.364E-01 3.775E+07 9.911E-05 7.122E-03 2.000E+05 0.000E+00 0
 1.80968497E-02   3851.6 1.809E-01 4.964E+07 1.009E-04 7.107E-03 2.000E+05 0.000E+00 0
 2.39098197E-02   3879.1 2.389E-01 6.520E+07 1.031E-04 7.024E-03 2.000E+05 0.000E+00 0
 3.14799737E-02   3907.9 3.146E-01 8.551E+07 1.058E-04 6.877E-03 2.000E+05 0.000E+00 0
 4.13026907E-02   3937.3 4.128E-01 1.119E+08 1.089E-04 6.792E-03 2.000E+05 0.000E+00 0
 5.39978597E-02   3967.9 5.396E-01 1.461E+08 1.126E-04 6.651E-03 2.000E+05 0.000E+00 0
 7.03314576E-02   3999.0 7.029E-01 1.903E+08 1.170E-04 6.532E-03 2.000E+05 0.000E+00 0
 9.12400269E-02   4030.7 9.118E-01 2.472E+08 1.222E-04 6.433E-03 2.000E+05 0.000E+00 0
 1.17861552E-01   4063.2 1.178E+00 3.202E+08 1.283E-04 6.376E-03 2.000E+05 0.000E+00 0
 1.51561784E-01   4096.6 1.515E+00 4.136E+08 1.356E-04 6.322E-03 2.000E+05 0.000E+00 0
 1.93971276E-01   4130.6 1.939E+00 5.327E+08 1.440E-04 6.230E-03 2.000E+05 0.000E+00 0
 2.47028432E-01   4165.2 2.469E+00 6.840E+08 1.540E-04 6.102E-03 2.000E+05 0.000E+00 0
 3.12913028E-01   4200.4 3.127E+00 8.751E+08 1.661E-04 6.042E-03 2.000E+05 0.000E+00 0
 3.94044580E-01   4236.5 3.938E+00 1.116E+09 1.805E-04 6.045E-03 2.000E+05 0.000E+00 0
 4.93218423E-01   4273.3 4.929E+00 1.418E+09 1.976E-04 6.012E-03 2.000E+05 0.000E+00 0
 6.13712298E-01   4310.2 6.134E+00 1.795E+09 2.174E-04 5.950E-03 2.000E+05 0.000E+00 0
 7.59354390E-01   4346.9 7.589E+00 2.260E+09 2.405E-04 5.868E-03 2.000E+05 0.000E+00 0
 9.34453147E-01   4383.3 9.339E+00 2.833E+09 2.675E-04 5.819E-03 2.000E+05 0.000E+00 0
 1.14361800E+00   4419.6 1.143E+01 3.536E+09 2.996E-04 5.859E-03 2.000E+05 0.000E+00 0
 1.39202516E+00   4456.1 1.391E+01 4.397E+09 3.371E-04 5.914E-03 2.000E+05 0.000E+00 0
 1.68572992E+00   4492.8 1.685E+01 5.449E+09 3.810E-04 5.960E-03 2.000E+05 0.000E+00 0
 2.03171101E+00   4529.5 2.031E+01 6.729E+09 4.320E-04 6.003E-03 2.000E+05 0.000E+00 0
 2.43798082E+00   4566.3 2.437E+01 8.284E+09 4.913E-04 6.041E-03 2.000E+05 0.000E+00 0
 2.91319671E+00   4602.8 2.912E+01 1.016E+10 5.614E-04 6.158E-03 2.000E+05 0.000E+00 0
 3.46669842E+00   4639.7 3.465E+01 1.243E+10 6.439E-04 6.336E-03 2.000E+05 0.000E+00 0
 4.10910720E+00   4677.1 4.107E+01 1.519E+10 7.410E-04 6.552E-03 2.000E+05 0.000E+00 0
 4.85250345E+00   4715.2 4.850E+01 1.853E+10 8.549E-04 6.775E-03 2.000E+05 0.000E+00 0
 5.71094812E+00   4753.6 5.707E+01 2.256E+10 9.880E-04 7.010E-03 2.000E+05 0.000E+00 0
 6.70076119E+00   4792.2 6.697E+01 2.742E+10 1.143E-03 7.273E-03 2.000E+05 0.000E+00 0
 7.83995052E+00   4830.6 7.835E+01 3.324E+10 1.326E-03 7.633E-03 2.000E+05 0.000E+00 0
 9.14883183E+00   4869.4 9.143E+01 4.024E+10 1.540E-03 8.049E-03 2.000E+05 0.000E+00 0
 1.06509428E+01   4908.6 1.064E+02 4.865E+10 1.791E-03 8.552E-03 2.000E+05 0.000E+00 0
 1.23721022E+01   4948.5 1.236E+02 5.882E+10 2.086E-03 9.167E-03 2.000E+05 0.000E+00 0
 1.43406662E+01   4989.5 1.433E+02 7.116E+10 2.435E-03 9.885E-03 2.000E+05 0.000E+00 0
 1.65870064E+01   5031.9 1.657E+02 8.620E+10 2.850E-03 1.078E-02 2.000E+05 0.000E+00 0
 1.91397794E+01   5076.7 1.912E+02 1.048E+11 3.353E-03 1.191E-02 2.000E+05 0.000E+00 0
 2.20253364E+01   5124.7 2.201E+02 1.281E+11 3.966E-03 1.330E-02 2.000E+05 0.000E+00 0
 2.52657253E+01   5177.2 2.524E+02 1.579E+11 4.729E-03 1.505E-02 2.000E+05 0.000E+00 0
 2.88729684E+01   5235.5 2.884E+02 1.968E+11 5.693E-03 1.728E-02 2.000E+05 0.000E+00 0
 3.28431001E+01   5301.4 3.281E+02 2.489E+11 6.945E-03 2.020E-02 2.000E+05 0.000E+00 0
 3.71474450E+01   5377.2 3.710E+02 3.208E+11 8.619E-03 2.415E-02 2.000E+05 0.000E+00 0
 4.17167059E+01   5465.7 4.166E+02 4.237E+11 1.097E-02 2.981E-02 2.000E+05 0.000E+00 0
 4.64444042E+01   5569.8 4.637E+02 5.759E+11 1.432E-02 3.789E-02 2.000E+05 0.000E+00 0
 5.12011280E+01   5692.3 5.111E+02 8.084E+11 1.928E-02 4.980E-02 2.000E+05 0.000E+00 0
 5.58271391E+01   5837.3 5.570E+02 1.177E+12 2.696E-02 6.840E-02 2.000E+05 1.343E-16 8
 6.01434833E+01   6009.1 5.999E+02 1.782E+12 3.940E-02 9.877E-02 2.000E+05 1.504E-13 8
 6.39878238E+01   6212.5 6.378E+02 2.809E+12 6.050E-02 1.508E-01 2.000E+05 8.848E-12 3
 6.72200726E+01   6455.4 6.695E+02 4.630E+12 9.916E-02 2.479E-01 2.000E+05 4.742E-10 1
 6.97558712E+01   6743.3 6.941E+02 7.939E+12 1.748E-01 4.423E-01 2.000E+05 3.654E-08 4
 7.15726141E+01   7093.0 7.111E+02 1.428E+13 3.417E-01 8.808E-01 2.000E+05 5.335E-06 2
 7.27480231E+01   7502.8 7.213E+02 2.612E+13 7.336E-01 1.937E+00 2.000E+05 4.243E-04 9
 7.34229176E+01   7998.6 7.259E+02 4.888E+13 1.773E+00 4.504E+00 2.000E+05 9.518E-03 2
 7.38088802E+01   8466.1 7.276E+02 7.958E+13 3.760E+00 6.740E+00 2.000E+05 2.613E-01 7
 7.41143108E+01   8729.0 7.287E+02 1.003E+14 5.470E+00 6.233E+00 2.000E+05 5.321E-01 9
 7.44043400E+01   8977.1 7.299E+02 1.211E+14 7.472E+00 5.480E+00 2.000E+05 6.891E-01 1
 7.47076853E+01   9146.5 7.314E+02 1.355E+14 9.023E+00 4.653E+00 2.000E+05 7.722E-01 1
 7.50410445E+01   9347.7 7.332E+02 1.523E+14 1.099E+01 4.638E+00 2.000E+05 8.359E-01 1
 7.54152703E+01   9533.0 7.353E+02 1.668E+14 1.273E+01 4.113E+00 2.000E+05 8.571E-01 1
 7.58518157E+01   9727.5 7.377E+02 1.805E+14 1.444E+01 4.710E+00 2.000E+05 8.786E-01 1
 7.63633342E+01  10021.4 7.404E+02 1.972E+14 1.648E+01 4.896E+00 2.000E+05 8.609E-01 1
 7.69811194E+01  10258.2 7.420E+02 2.068E+14 1.730E+01 1.017E+01 2.000E+05 8.167E-01 1
 7.79128042E+01  11707.2 7.365E+02 2.113E+14 1.289E+01 2.174E+01 2.000E+05 1.379E-01 9
 8.00878551E+01  13836.5 7.219E+02 1.803E+14 4.912E+00 1.161E+01 2.000E+05 1.117E-04 1
 8.65459537E+01  15703.8 7.263E+02 1.642E+14 2.824E+00 7.005E+00 2.000E+05 1.920E-05 6
PRADK 1.7318E+00
```

B A complete SYNTHE COM file

This is the complete and unaltered VMS COM file examined in section 3.3.

```
$ SET DEF [zwitter.CALC]
$ ASSIGN output.dat for006
$ ASSIGN MODELS:ap00t6000g45k2nover.mod for005
$ assign SYNTHE:molecules.dat for002
$ assign SYNTHE:pfiron.dat for004
$ ASSIGN SYNTHE:CONTINUA.DAT FOR017
$ ASSIGN XNFp00t6000g45k2nover.dat for010
$ RUN SYNTHE:XNFPELSYN
$ ASSIGN SYS$INPUT FOR005
$RUN SYNTHE:SYNBEG
AIR       850.0     875.0     500000.   0.0    0         300 .001             00
AIRorVAC  WLBEG     WLEND     RESOLU    TURBV  IFNLTE LINOUT CUTOFF        NREAD
$ASSIGN LINES:GF0300.100 FOR011
$RUN SYNTHE:Rgfalltest
$ASSIGN LINES:GF0400.100 FOR011
$RUN SYNTHE:Rgfalltest
$ASSIGN LINES:GF0500.100 FOR011
$RUN SYNTHE:Rgfalltest
$ASSIGN LINES:GF0600.100 FOR011
$RUN SYNTHE:Rgfalltest
$ASSIGN LINES:GF0800.100 FOR011
$RUN SYNTHE:Rgfalltest
$ASSIGN LINES:GF1200.100 FOR011
$RUN SYNTHE:Rgfalltest
$assign MOLECULES:c2ax.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:c2ba.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:c2da.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:c2ea.bin FOR011
$run SYNTHE:Rmolec
```

```
$assign MOLECULES:chax.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:chbx.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:chcx.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:cnax.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:cnbx.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:coax.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:coxx.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:h2bx.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:h2cx.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:mghax.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:mghbx.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:nhax.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:nhca.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:ohax.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:ohxx.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:sihax.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:sioax.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:sioex.bin FOR011
$run SYNTHE:Rmolec
$assign MOLECULES:sioxx.bin FOR011
$run SYNTHE:Rmolec
$ ASSIGN XNFp00t6000g45k2nover.dat for010
$ RUN SYNTHE:SYNTHEkur
$ ASSIGN MODELS:ap00t6000g45k2nover.mod for005
$ASSIGN i4600-9000.dat FOR007
$ASSIGN SYS$INPUT FOR025
```

```
$assign synthe:molecules.dat for002
$RUN synthe:SPECTRV
0.0       0.        1.        0.        0.        0.        0.        0.
0.
RHOXJ     R1        R101      PH1       PC1       PSI1      PRDDOP    PRDPOW
$ASSIGN i4600-9000.dat FOR001
$assign sys$input for005
$ ASSIGN f85008750v0.dat rot1
$ ASSIGN f85008750v20.dat rot2
$ ASSIGN f85008750v100.dat rot3
$run synthe:rotate
   3
0.        20.       100.
$ assign sys$input for005
$ assign broadenout.txt for006
$ assign f85008750v0.dat for021
$ assign f85008750v0-r20000p00t6000g45k2nover.dat for022
$ run synthe:BROADEN
GAUSSIAN   20000.   RESOLUTION
$ assign f85008750v0-r20000p00t6000g45k2nover.dat for001
$ assign f85008750v0-r20000p00t6000g45k2nover.asc for002
$ run synthe:CONVERFSYNNMTOA
$ delete/noconf f85008750v0-r20000p00t6000g45k2nover.dat.*
$ assign f85008750v20.dat for021
$ assign f85008750v20-r20000p00t6000g45k2nover.dat for022
$ run synthe:BROADEN
GAUSSIAN   20000.   RESOLUTION
$ assign f85008750v20-r20000p00t6000g45k2nover.dat for001
$ assign f85008750v20-r20000p00t6000g45k2nover.asc for002
$ run synthe:CONVERFSYNNMTOA
$ delete/noconf f85008750v20-r20000p00t6000g45k2nover.dat.*
$ assign f85008750v100.dat for021
$ assign f85008750v100-r20000p00t6000g45k2nover.dat for022
$ run synthe:BROADEN
GAUSSIAN   20000.   RESOLUTION
$ assign f85008750v100-r20000p00t6000g45k2nover.dat for001
$ assign f85008750v100-r20000p00t6000g45k2nover.asc for002
$ run synthe:CONVERFSYNNMTOA
$ delete/noconf f85008750v100-r20000p00t6000g45k2nover.dat.*
$ copy  f85008750v0-r20000p00t6000g45k2nover.asc [zwitter.calc]
$ copy  f85008750v20-r20000p00t6000g45k2nover.asc [zwitter.calc]
$ copy  f85008750v100-r20000p00t6000g45k2nover.asc [zwitter.calc]
$ DEL/NOCONF *.*.*
```

References

CASTELLI, F. (1988): *Kurucz's models, Kurucz's fluxes and the ATLAS code. Models and fluxes available at OAT*. Osservatorio Astronomico di Trieste. Publicazione OAT n. 1164.

CASTELLI, F.; GRATTON, R. G.; KURUCZ, R.(1997): Notes on the convection in the ATLAS 9 model atmospheres. *Astron. Astrophys.* **318**, 841-869.

CASTELLI, F.;KURUCZ, R.(2004): Modelling of Stellar Atmospheres, IAU Symposium.

GRAY, D. F.(1995): *The observation and analysis of stellar photospheres (Second Edition)*. Cambridge University Press.

GREVESSE, N.; SAUVAL, A.J.(1998): *SpaceSciRev* **85**, 161.

Jeffrey, C. S.(1994):`atlas.lester`*: Utility programs for manipulating ATLAS9 i/o*. CCP7 User Note 12.1. Armagh Observatory.

Jeffrey, C. S.; Short, C. I.; Lester, J. B.(1994):`atlas.lester`*: a UNIX implementation of ATLAS and SYNTHE*. CCP7 User Note 15.0. Armagh Observatory.

KURUCZ, R. (1969): A matrix method for calculating the source function, mean intensity and flux in a model photosphere. *ApJ* **156**, 235-240.

KURUCZ, R. (1970): ATLAS: A computer program for calculating model stellar atmospheres. SAO Special Report 309.Smithsonian Institution.

KURUCZ, R.; FURENLID, I. (1979): *Sample spectral atlas for Sirius*. SAO Special Report 387. Smithsonian Institution.

KURUCZ, R.; AVRETT, H. (1981): *Solar spectrum synthesis I. A sample atlas from 224 to 300 nm*. SAO Special Report 391. Smithsonian Institution.

PAGE, C. R. (1988): *Professional Programmer's Guide to FORTRAN 77*. Freely available at `http://www.lahey.com/other.htm`.

Sbordone, L; Bonifacio, P; Castelli, F; Kurucz, R.(2004): *Mem. Soc. Ast. It.Suppl.* 5, 93.