Construction of isotope-averaged Synspec-compatible line lists from EXOMOL data

1 Physical background

The line opacity for the isotopologue Y$\equiv {}^k\mathrm{A}^l\mathrm{B}$ is given by

$$\kappa_{ij}(\nu) = (gf)^Y_{ij}(N_Y/U_Y)\,\phi^Y_{ij}(\nu, T). \tag{1}$$

(we consider here for simplicity a diatomic molecule, but the situations is analogous for any number of atomic constituents). Here

$$\phi_{ij}(\nu, T) = \frac{\pi e^2}{mc}\exp(-E_i/kT)\frac{H(a_{ij}, x)}{\sqrt{\pi}\Delta\nu_D}, \tag{2}$$

However, it is impractical to consider all isotopoloques as independent species. As usual, one instead expresses the line opacity as a function of the total number density for a molecule AB.

$$\kappa^Y_{ij}(\nu) = (gf)^Y_{ij}\,\frac{N_{AB}}{U^{\mathrm{ref}}_{AB}}\,f^k_A f^l_B\,\phi^Y_{ij}(\nu, T), \tag{3}$$

where the "reference" partition function is given by

$$U^{\mathrm{ref}}_{AB} = \sum_k \sum_l U^{k,l}_{AB}\,f^k_A\,f^l_B. \tag{4}$$

and f^k_A stands for the fractional abundance of the isotope ${}^k\mathrm{A}$ with respect to the total abundance of element A, etc. This expression is valid assuming that the atomic partition functions are isotope-independent, the dissociation constants for all isotopologues are equal, and neglecting differences in the reduced mass for the individual isotopologues.

The line opacity can then be written as

$$\kappa^Y_{ij}(\nu) = (gf)^{\mathrm{mod}}_{ij}\,\frac{N_{AB}}{U^{\mathrm{ref}}_{AB}}\,\phi^Y_{ij}(\nu, T), \tag{5}$$

where the "modified" gf value is given by

$$(gf)^{\mathrm{mod}}_{ij} = (gf)^Y_{ij}\left(\frac{U^Y_{AB}}{U^{\mathrm{ref}}_{AB}}\right) f^k_A f^l_B, \tag{6}$$

In the following, we describe a practical evaluation of the modified gf value, $(gf)^{\mathrm{mod}}_{ij}$

2 Practical construction

It is advantageous to create a parent directory for EXOMOL data, say `./EXOMOL` with a number of subdirectories for the individual species, e.g. `./EXOMOL/AlH`, `./EXOMOL/AlO`, etc, as well as a subdirectory containing the utility programs, `./EXOMOL/xprog`.

2.1 Utility programs

There are four basic FORTRAN77 programs:

Program isosum.f

Is a program that generates isotope-averaged partition function for a given molecular species, based on the EXOMOL data for the individual isotopologues. The relative abundance of the individual istopologues is assumed to have solar values, which are given in the table stored in the file `isotps`. The EXOMOL partition functions are given as simple tables of the partition function vers. temperature. In most cases, the values of the lowest and the highest temperature considered, T_0 and T_1, is the same for all isotopologues of the given molecular species, as is the of temperature step, but not always. In such cases, the program does not extrapolates, but only takes values from the available range of temperatures. This does not present significant inaccuracy because the limited ranges of temperatures are usually adopted for isotopologues with very small relative contributions to the total abundance of the given molecular species. If the ranges of temperatures are the same for all isotopologues, but the total numbers of temperatures are not (for instance, for CO, where there are 9000 temperatures the isotopologues containing ^{16}O, while only 3000 points for those containing ^{17}O and ^{18}O). In such cases, the program interpolates to the temperatures set composed of the largest number of temperatures.

The standard input to the program has $N + 1$ lines, with N being the total number of isotopologues. The first line is

iat1, iat2 - atomic numbers of the components

and then there is one line for each component, containing

iso1, iso2, filepf - where `iso1` and `iso2` are the nucleon numbers of the given atomic component, and `filepf` is a string containing the name of the EXOMOL partition function file.

For instance, for AlO, the input file, named `alo.5`, looks like:

```
13  8
27 16 '27Al-16O__ATP.pf'
27 17 '27Al-17O__ATP.pf'
27 18 '27Al-18O__ATP.pf'
```

Program list.f

Is a program to extract line data for a particular isotopologue of a molecule from the EXOMOL data and output them in the SYNSPEC format. It assumes that the total (isotope-averaged) partition function of the molecule is already given, either computed by a previous run of the program `isosum`, or copied (linked) form some other source. Its name, however, should comply with the convention adopted here, namely containing just the chemical label, with the extension "`.pf`", e.g., `AlO.pf`.

Standard Input:

type - a Kurucz-type label of a molecule, e.g 813.00 for AlO

molname - a character variable with the chemical label, e.g. 'AlO'

iat1, iso1 - atomic and nucleon numbers of the first atom, e.g 13, 27 for ^{27}Al

iat2, iso2 - the same for the second atom, e.g. 8 16 for ^{16}O

core - a character variable with the core name of the EXOMOL files, e.g. `'27Al-16O__ATP'`

wbeg - lowest wavelength [nm] for a line to be included in the list

wend - the same for the longest wavelength

gflim - $\log(gf)$ limit – lines with $\log(gf)$ below gflim are excluded.

The four input lines are organized as follows:

```
type   molname
iat1  iso1  iat2  iso2
core
wbeg wend gflim
```

For instance, the input file for the line list of ^{27}Al^{16}O (the file `cc1.5` of the example in the next section) is

```
813.00   'AlO'
13  27  8  16
'27Al-16O__ATP'
100 100000 -9.0
```

Output:

std.output - a short log file

fort.10 - the resulting line list in the SYNSPEC format, but in reverse wavelength order - from the long to short. An accompanied short program `reverse.f` reverses it to the ordering needed by SYNSPEC , which is from short to long wavelengths.

Program reverse.f

This is a very simple program, with the standard input being the line list produced by `list.f`, and the standard output is the reversed list, in the SYNSPEC format.

Program merge.f

Again, a very simple program that merges two lists, the first one is being submitted as the standard input, and the second one as `fort.10`. The resulting merged file is output as `fort.11`. The program also generates a brief log file.

In the followig example, it is assumed that these program are compiled, with the names of the executables `isosum.exe`, `list.exe`, `reverse, exe`, and `merge.exe`.

2.2 Construction of line lists for individual species

Each subdirectory should contain two types of data:
1) the files directly downloaded from the EXOMOL database;
2) the input files to programs generating SYNSPEC-compatible line lists.

The files of the first type, are, for instance for the `./AlO` subdirectory, the following:

```
26Al-16O__ATP.pf
26Al-16O__ATP.states
26Al-16O__ATP.trans
```

and analogous triads `27Al-16O__ATP.*`, `27Al-17O__ATP.*`, and `27Al-18O__ATP.*`, that represent files obtained directly from the EXOMOL database. These files contain the partition functions (`*.pf`), energy level parameters (`*.states`), and the data for transitions (`*.trans`) , for the individual isotopologues.

The files of the second type have to be generated by the user. These are:
1) the input file to the program `isosum.f` to generate the isotope-averaged partition function (its example is showed in the previous section), or, alternatively, an already computed averaged partition function; and
2) the script file `R1` to generate the line list, which in the case looks like this:

```
 ln -s -f ./xprog/isotops .
../xprog/list.exe <cc1.5 > cc1.log
../xprog/reverse.exe <fort.10 >cc1.list

../xprog/list.exe <cc2.5 > cc2.log
../xprog/reverse.exe <fort.10 >cc2.list

../xprog/list.exe <cc3.5 > cc3.log
../xprog/reverse.exe <fort.10 >cc3.list
```

```
ln -s -f cc1.list fort.10
../xprog/merge.exe <cc2.list >cc12.log
mv fort.11 cc12

ln -s -f cc12 fort.10
../xprog/merge.exe <cc3.list >cc13.log

# remove the isotope specific line lists (if not need for other purposes)
rm -f cc[1-3].list

mv fort.11 AlO.list
```

Here, the file `isotops`, located in the `./EXOMOL/xprog` directory, contains a table of the solar isotopic ratios for all important chemical elements. The utility programs `list, reverse`, and `merge` and the input data for them are described above.

2.3 Practical implementation

The package is distributed as a tar gzipped file `exomol.tar.gz`. Upon extracting, it generates directory `./exomol` with two subdirectories, `./exomol/xprog` and `./exomol/AlO`. The first one contains utility programs, as described before, as well as the file `isotops`. These four program need to be compiled, e.g.
`gfortran -fno-utomatic -O3 -o isosum.exe isosum.f`
and analogously for other programs.

Directory `AlO` represent an example of an individual directory for a given molecule, as described above. It contains the script file `R0` to generate the total partition function, the input file `alo.5` which is used by the script `R0` (via program `isosum`). Then it contains script `R1` to generate the line list for AlO, and the input files `cc1.5, cc2.5` and `cc3.5` used by the script. The directory also contains file `AlO.pf-orig` for a check of a correct run of script `R0`.

Before running these scripts, the user has to download the corresponding `*.pf, *.states`, and `*.trans` files by visiting the EXOMOL website,
`https://www.exomol.com/data/molecules/`
and selecting the appropriate molecule, in this case AlO. In this particular case, one does not need to download files for 26Al-16O, albeit the data are present there, because the 27Al is the only isotope present in the Sun (see the table `isotops`). Once the files are downloaded, the user may run the script `R0` to get the total partition function, and then the script `R1` to generate the line list for AlO.