Manual for the linelist converter v.2022.11.05

Marco Pezzella, [a] Jonathan Tennyson, [a,1] Sergei Yurchenko, [a]

[a] *Department of Physics and Astronomy, University College London, London, WC1E 6BT, UK*

The program LINELIST converts the outputs from DVR3D procedure (fort.14 from DVR3D and ROTLEV, fort.13 from DIPOLE) in the EXOMOL format. The files fort.14 have information on the energy levels for each rotational state and symmetry (and it will be referred as *energy file*); the file fort.13 stores the data from the transitions (initial state, final state, Einstein A coefficient, and (optionally) transition frequency *intensity file*).
The program reads an energy file for each rotational state and an intensity file for each combination of bras and kets, generating an unique states and transition file. Those two files collect all the data required to produce a proper cross section file using EXOCROSS.

Compilation and running

The program is written to be compiled using the Intel compiler, using the makefile attached. It is run using the following command:

```
./linelist.x <input > output
```

The code is tested on UBUNTU 20.04 on 32 GB, 8 core laptop machine, and on a CentOS server.

Input structure

The input is based on the Fortran90 input parsing module written by Anthony J. Stone in 2005, and distributed under the terms of the GNU General Public

[1] Author to whom correspondence should be addressed. email: j.tennyson@ucl.ac.uk, fax +(44) 20 7679 7145, telephone +(44) 20 7679 7155

License.

```
UVVIS
NSTATES n_s
do i=1,n_s
 filename(i) [label(i)]
end do
NTRANS N_t
do j=1,n_t
 filename(j)
end do
OUT_STATE outstatefile
OUT_TRANS outtransfile
ENERMAX E_max
ENCUTOFF E_cutoff
SYMMETRY ABC/AB2
GNS GNS(1) GNS(2) [GNS(3) GNS(4)]
SEGMENTED
NOTRANS
```

UVVIS : optional parameter. If it is present in the input, inidcates that different electronic states are present.

NSTATES : followed by an integer (n_s) indicating the number of states files that have to read, max 400 files. It is followed by n_s lines containing the filenames. When multiple electronic states are considered, a second argument, the label of the electronic state, is required.

NTRANS : followed by an integer (n_t) indicating the number of transition files to read, max 400 files. It is followed by n_t lines containing the filenames.

OUT_STATE : followed by the name of the output states file.

OUT_TRANS : followed by the initial part of the name of the output transition file. As transition files can be hugely massive, the current implementation cut them every 1000 cm^{-1}, creating a series of files defined as *outtransfile.idx.trans*. Here *idx* is a number starting from 5000 and increasing by 1 unit every 1000 cm^{-1}.

ENERMAX : followed by E_{max}, the maximum energy (in cm^{-1}) of a state to be considered when generating a transition file.

ENCUTOFF : followed by E_{cutoff} the cutoff energy (in cm^{-1}) of the transitions to be included in the transition files.

SYMMETRY : followed by the molecular symmetry. Only the C_s (ABC), and the C_{2V} (AB2) symmetries are allowed.

GNS : two integer numbers for ABC symmetries and four for AB2, indicating the nuclear statistical weights. For ABC symmetries they are respectively A' and A", while for AB2 they are A_1,A_2,B_1,B_2.

SEGMENTED informs the program if the transition files are generated using a segmented file.

NOTRANS : informs the program to not produce transition files .

Output structure

As mentioned earlier, running LINELIST produces two outputs: the states file, that have the information about the energy levels, and the transition file, that contains the transition frequencies and Einstein A coefficients of each transition.

States file

The states file has all the information required to generate the energy levels and assign the parity and the Kronig symmetries. Each line has the following data:

$\boldsymbol{i}$	: the level index.
$\boldsymbol{E}$	: the energy of the corresponding level (in cm$^-$1).
$\boldsymbol{gns(2J+1)}$	: The nuclear statistical weight, times the rotational level degeneracy.
$\boldsymbol{J}$	: the rotational quantum number.
pm	: the parity of the state.
ef	: the Kronig quantum number.
symm	: the symmetry of the state.
$\boldsymbol{v}$	: the vibrational quantum number.
$\boldsymbol{l}$	: the angular momentum quantum number.
$\boldsymbol{S}$	: the spin quantum number.
Ω_s	: the spin projection quantum number.

The last four quantum numbers are set by default to "Nan", as we need to implement these functionality. The formatting of each line is given in the following lines:

```
  i ,E, gns(2J+1),J ,pm, ef ,symm, v , l , S , Omega_s
i12 , f12.6 , i6 , i7 , a1 , a1 , a4 , i2 , i2 , i2 , i2
```

Transitions file

At the moment the transition file is created as an empty file. Transitions are stored in a series of files, starting from fort.5001 onwards, each of them

storing transitions within 1000 cm^{-1} interval. These files contain the following information:

$\boldsymbol{f}$	: index of the final state.
$\boldsymbol{i}$	: index of ithe initial state.
A	: Einstein A coefficient (s^{-1}).
E_{trans}	: energy of the transition (in $cm^{-}1$).

The formatting of each line is given in the following lines:

```
f,i,A,E_trans
i12,i12,es16.8,f16.6
```

The final version of the transition can be created combining these files and using various functions in bash (*e.g.* sort function), R (*e.g.* order function), and Python.