ExoMolOP opacity notes

Katy Chubb

June 2023

Files on Dial in `/scratch/dp060/dc-chub1/ExoCross/files_ExoMolOP/`
Note that the cut-off has been now fixed at 25cm^{-1} using: `define['fixed_cutoff'] = 25` in the input files.

1 Non-super-lines

Using SO as an example here.

1. Set up input file and replace:

- Rename e.g. `SO_input.py` and replace these lines:
- `define['molecule_name'] = 'SO'`
- Folders in section:
  `if define['platform'] == 'cobweb':`
  (where the line-list is stored, and where all opacities should be created for that molecule)[1]
- `define['mean-mass'] = 48.064`
  (molar mass)
- `define['ptfile'] = 'SO_march23.pf'`
- `define['gamma'] = 0.1203`
  `define['gamma-n'] = 0.5`
  `define['gamma-He'] = 0.0475`
  `define['gamma-n-He'] = 0.5`
  (averaged broadening parameters for H_2 and He)
- If have broadening files uncomment and replace:
  `#define['broadeners'] = {}`
  `#define['broadeners'][0] = (0.86, 'SO2_H2.broad')`
  `#define['broadeners'][1] = (0.14, 'SO2_He.broad')`

[1] Ignore reference to cobweb, this is now Dial

- `define['min_gridspacing'] = 0.01`
  can also be set to 0.1 (in units of wavenumber) if the line list is larger
- **Define range?**

  ```
  define['ranges'] = [(100,60000)]
  define['fullrange'] = [(100,60000)]
  ```

2. Create input and batch run files

- `python generate_submit_nosplit_GRID_nosuper.py SO_input.py`[2]
- This will create the input files for all the pressure-temperature combinations in the input folder. A batch file for running (currently 594 files, should take less than an hour per file unless the line list is too large):
  `sh batch_submit_all.sh`

3. Run reinterpolate to combine xsec files

- This should be very quick to run (less than 1 hour, which is the number at the end):
  `./b_reinterpolate.sh SO_input.py 1`
  This is equivalent to the following inline command:
  `python reinterpolate_xsec.py SO_input.py`

4. Create all cross-section and k-table files for the 4 retrieval codes

- These files are all stored in:
  `/scratch/dp060/dc-chub1/ExoCross/general_replace_files_run_all`
  (RUNALL currently assumes this folder is one level back from the folder you are running the opacities in, you can change this)
- Open the file RUNALL and replace parts of these lines for the molecule:
  `sed -i -e 's/xxx/32S-16O__ExoMol_March2023/g' grep_replace_tmp`
  (the line list and iso reference)

  `sed -i -e 's/yyy/SO/g' grep_replace_tmp`
  (molecule name)

  `sed -i -e 's/zzz/48/g' grep_replace_tmp`
  (the molar mass)

  `sed -i -e 's/qqq/10.1039\\\/D2CP03051A/g' grep_replace_tmp`

[2] module load python/intel/3.6

(the DOI of the line-list publication)

`sed -i -e 's/vvv/v0_080323/g' grep_replace_tmp`
(The version number, usually just v1 and the date)
Note leave all xxx, yyy, zzz etc parts as they are.

- `sh RUNALL`
- In case of any problems, the corresponding command lines are (for an LiOH example)

```
python3 create_ktables_R1000.py --dict LiOH_input.py  --wl 0.3,50 --resolution 1000 --ngauss 20 --ncores 36
python3 sample_xsec_hdf5.py  --dict ./LiOH_input.py  --wl 0.3,50 --resolution 15000 --key_iso_ll 7Li-16O-1H
python3 create_ktables_NEMESIS_hdf5_R1000_NEWORDER_microns.py  --dict LiOH_input.py  --ncores 36 --title pe
python3 create_ktables_NEMESIS_hdf5_R1000_NEWORDER_microns.py --dict LiOH_input.py --wl 0.3,50 --resolution
```

5. Folders with a file for all 4 codes should now be created (within a few hours)

- PetitRADTRANS folder:
  `xsec_ktable_petitRADTRANS_R1000_final_v1_16g/`
- PetitRADTRANS file in that folder:
  `32S-16O__ExoMol_March2023.R1000_0.3-50mu.ktable.petitRADTRANS.h5`
- TauREx3 folder:
  `xsec_hdf5_sampled_R15000_0.3-50_final_v1/`
- TauREx3 file in that folder:
  `32S-16O__ExoMol_March2023.R15000_0.3-50mu.xsec.TauREx.h5`
- NEMESIS folder:
  `xsec_ktable_NEMESIS_final_v1_20g/`
- NEMESIS file in that folder:
  `SO_R1000_0.3-50mu.ktable.TauREx.h5`
  (needs extra conversion, see below)
- ARCiS folder:
  `xsec_ktable_R1000_0.3-50mu_20g_final_v1/`
- ARCiS file in that folder:
  `32S-16O__ExoMol_March2023_R1000_0.3-50mu.ktable.TauREx.pickle`
  (needs extra conversion, see below)

6. Convert ARCiS

NEMESIS and ARCiS require an extra step for converting.

- For ARCiS:
  In `NEW_TauRex2ARCiS.py` replace:
  `32S-16O__ExoMol_March2023_R1000_0.3-50mu.ktable.TauREx.pickle`
- and then run:
  `python NEW_TauRex2ARCiS.py`
  `gzip 32S-16O__ExoMol_March2023.R1000_0.3-50mu.ktable.ARCiS.fits`
- For NEMESIS:
  In `Convertbinary_R1000_neworder.py` replace:
  filename and `output_filename`
- `x=60 #IDGAS1 FROM HITRAN`
  (The ID is from `NEMESIS_IDS.txt`, if there is a new molecule without an ID there I usually check with Jo Barstow which new ID to use)
- and then run:
  `python Convertbinary_R1000_neworder.py`

2 Super-lines

Very similar to above but with some different files and an extra stage to run (step 1 of superlines). Using SO_2 as an example here.

1. Set up input file and replace:

Same as for non-superlines setup above, but likely with more .trans files to add in.

2. Create input and batch run files

Same as above but use a different file, called `generate_submit_super_trans.py`,
Again `define['min_gridspacing'] = 0.01` can be changed to 0.1.

- `python generate_submit_super_trans.py SO2_input.py`
- This time there are two batch files, the first needs to run and complete first before the second:
  `sh batch_submit_super1.sh`
  the super-lines step 1 files need to be re-named to have .super suffix, as that is what the step 2 files will read from.

  `sh batch_submit_super2.sh`

After that everything else should be the same as for the non-super-lines case.

3 Combining opacities

If you look on DIAL at the folder `/scratch/dp060/dc-chub1/ExoCross/CO_comb` then you'll see some files:
`b_combine_isos.sh`
`run_combine_isos.csh`
`combine_isos.py`
`batch_run_combine.sh`
`batch_run_combine.sh` contains the 594 standard pressure/temperature combinations;
`combine_isos.py`: here you need to specify the folders with the xsecs for the isotopologues you would like to combine using `iso1_inp`, `iso2_inp`, etc. I think the `.xsec` files you want to combine need to have the same number of rows (i.e. computed at the same wavenumbers, using the same resolution or number of points) for each given T/P combination for each isotopologue you want to combine. If they don't you would need to interpolate them to the same grid of wavenumbers. You then need to change this section to have the relative abundances of each isotopologue, summed to 1:

```
#iso abundanaces
#1216
a1 = 0.987
#1217
a2 = 0.0000037
#1218
a3 = 0.00002
#1316
a4 = 0.011
#1317
a5 = 0.000000041
#1318
a6 = 0.00000022
```

where

$$y_c = a1 * y[i] + a2 * y2[i] + a3 * y3[i] + a4 * y4[i] + a5 * y5[i] + a6 * y6[i]$$

Again, cut down if you have less.

Then name the folder and file for your output xsecs:
`./xsec_br/CO_NA_T%s_P%s.xsec` and run `batch_run_combine.sh`.

Create directory 'xsec'.

After that run the other steps as normal but with a new name for the isotopologue and linelist (e.g `C-O-NatAbund__Li2015\verb`). For this I create a new input file, e.g. `CO_comb_input.py`, and call the molecule e.g. `CO_NA`.

HITRAN lists some of the main abundances but not all at `https://hitran.org/docs/iso-meta/`.