

Documentation SpectroscPy

Karen Oda Hjorth Minde Dundas

October 23, 2020

SpectroscPy is a Python scriptpackage designed to postprocess outputs from OpenRSP/LSDalton response property calculations and determine the desired spectroscopic properties. As of August 2019, it works for vibrational properties, specifically IR and Raman, but can be easily extended to other types of spectroscopies. It should also be a small task to link it to other program packages.

This document is a user guide, and should get you ready to use the package.

You run the code by calling `SpectroscPy_run` which is inside the file `SpectroscPy_wrapper.py`. An example is as follows

```
from SpectroscPy_wrapper import SpectroscPy_run
run_specification = ['Multiple snapshots', 'No FraME', 'Plot IR', 'Outproj']
spectroscopy_type = ['IR']
spectroscopy_specification = ['Vib modes: 1/m', 'IR: MDAC, m**2/(s*mol)']
cauchy_type = 100.0
names = ['test_w_mult_snapshots/', 0, 5, 'hf', 'optimized', 'No FraME', [2]]
harmonic_frequency_limits = 'Keep all'
spectral_boundaries = 'Keep all'
T = 298
print_level = 0
SpectroscPy_run(run_specification, spectroscopy_type, spectroscopy_specification,
cauchy_type, names, harmonic_frequency_limits, spectral_boundaries, print_level, T)
```

This is a calculation for multipole snapshots without FraME, to get both table values and a plotted spectrum of the molar decadic attenuated IR coefficients in $\text{m}^2/(\text{s} \cdot \text{mol})$ units. The broadening factor of the plot is given, and is 100.0. I will now go through the various different options one at a time and explain them.

1 Preparations

Prior to SpectroscPy calculation, you have to run an OpenRSP/LSDalton calculation to obtain the tensors needed for the desired spectroscopy. For IR this is the molecular Hessian and dipole gradient, while for Raman it is the molecular Hessian and the polarizability gradient. This gives a file of the type `.rps_tensor` which we need.

2 run_specification

Contains four different categories, how many snapshots, is it with or without PE, what should be plotted and whether the rotational and translational modes should be projected

out or not.

Snapshots

You have to specify if this is a calculation for a single structure, or if it is for a set of snapshots from an MD simulation. The first is typically the case if you have a vacuum structure, and the other if you have one with environment where you sample the configurations. The options (you must choose one of them) are 'Single snapshot' or 'Multiple snapshots'.

With or without PE

Choose whether you would like to use the PE method for the environment. The options (you must choose one of them) are 'FraME' or 'No FraME'.

What to plot

When you calculate values for IR and/or Raman spectroscopy, you can choose whether you want to plot these or not. If you choose not to, the average values will still be printed. You can also choose how many lines you want in each plot. If you choose one of the options with more than one line per plot, be aware that all the lines will be scaled to max 1.0. Choose as many as you wish from the list below, or none.

- 'Plot IR'
- 'Plot Raman separately'
- 'Plot Raman together'
- 'Plot IR and individual Raman together'
- 'Plot IR and all Raman together'
- 'Plot all Hyper-Raman separately'
- 'Plot Hyper-Raman VV and HV separately'
- 'Plot all Hyper-Raman together'

Outprojection or not

For vacuum calculations it is common to project out rotational and translational modes¹ in order to isolate the vibrational modes. The effect of this in a constraining environment is however not wellknown, and an option to not to perform this outprojection is therefore given. Specify one of the following (you must choose one) 'Outproj' or 'No outproj'.

Symbolic or written axis labels

The default label-type on the plot axes is a written one, but if you wish, you can choose a symbolic one. Do that by specifying 'Symbolic labels'. This will for instance change IR MDAC L/(cm*mol) to $\epsilon : L \cdot \text{cm}^{-1} \text{mol}^{-1}$.

Anharmonic

Anharmonic corrections can be calculated, both for frequencies and intensities. Various levels of theory are available, for intensities, up to DVPT2, and for frequencies up to GVPT2. The options are made with the view of being comparable to Gaussian, and are

- Nothing: harmonic
- 'Harmonic': harmonic
- 'Anharmonic': GVPT2 frequencies and DVPT2 intensities, all with 1-1 resonance checks (default).
- 'Anharmonic: Freq DVPT2, Int VPT2': DVPT2 frequencies and VPT2 intensities.
- 'Anharmonic: DVPT2': DVPT2 frequencies and intensities.
- 'Anharmonic Freq: DVPT2': DVPT2 frequencies and intensities.
- 'Anharmonic: DVPT2, w/ 1-1 checks': DVPT2 frequencies and intensities, with 1-1 resonance checks.
- 'Anharmonic: Freq GVPT2, Int DVPT2': GVPT2 frequencies and DVPT2 intensities
- 'Anharmonic: Freq GVPT2, Int DVPT2, w/ 1-1 checks': GVPT2 frequencies and DVPT2 intensities, with 1-1 resonance checks (same as 'Anharmonic').

The fermi resonance checks are based upon

$$|\omega_i - \omega_j - \omega_k| \leq 200\text{cm}^{-1} \quad (2.1)$$

$$\Delta \geq 1\text{cm}^{-1} \quad (2.2)$$

where Δ is given by the Martin parameters². In order to identify a Fermi resonance, both conditions must be satisfied. The 1-1 checks are based on

$$|\omega_i - \omega_j| \leq 100\text{cm}^{-1} \quad (2.3)$$

$$|K| \geq 1\text{cm}^{-1} \quad (2.4)$$

$$\frac{|K|}{(\omega_i - \omega_j)^2} \geq 1\text{cm}^{-1} \quad (2.5)$$

where K is the coupling of the term; k_{ijkl} for quartic terms, $k_{ijk} \cdot k_{lmn}$ for cubic terms and the coefficient sum for the Coriolis terms. For a 1-1 resonance to be identified, the first condition must be true, and at least one of the two others.

3 spectroscopy_type

Here you choose which spectroscopic values you wish to calculate. Choose at least (but more if you want) among 'IR', 'Raman' and 'Hyper-Raman'. You must choose at least one.

4 spectroscopy_specification

In the literature and in many codes there is a high degree of ambiguity when it comes to what is actually meant by IR 'intensities'. In many cases it is actually the molar attenuated coefficient, or parts of it, that is reported. To remove any uncertainty, when you have 'IR' in `spectroscopy_type`, you have to specify one of the following through `spectroscopy_specification`

- 'IR: SSDG, a.u.': Infrared SSDG (summed and squared dipole gradients) in a.u
- 'IR: SSDG, C**2/kg': Infrared SSDG (summed and squared dipole gradients) in C^2/kg
- 'IR: SSDG, D2A2/amu': Infrared SSDG (summed and squared dipole gradients) in $(D/A)^2/amu$. A is here for angstrom.
- 'IR: MDAC, m**2/(s*mol)': Infrared MDAC (Molar decadic attenuated coefficient) in m^2/mol , or w/o lineshape in $m^2/(s \cdot mol)$
- 'IR: MDAC, L/(cm*s*mol)': Infrared MDAC (Molar decadic attenuated coefficient) in $L/(cm \cdot mol)$, or w/o lineshape in $L/(cm \cdot s \cdot mol)$
- 'IR: NIMAC, m/mol': Infrared NIMAC (Naperian integrated molar absorption coefficient) in m/mol
- 'IR: NIMAC, km/mol': Infrared NIMAC (Naperian integrated molar absorption coefficient) in km/mol (Default in Dalton)

The definitions of these values will be described in more detail in my thesis or somewhere else, but if you would like to know them before it's finished, contact me.

Raman is by in the literature reported in arbitrary units, but it is useful to make distinctions on what these are based on. In SpectroscPy there are no arbitrary units, everything is based on absolute values. There is also a choice regarding the combination of the polarizability gradient terms a_i^2 and b_i^2 . You therefore need to specify one of the following through `spectroscopy_specification`

- 'Raman: CPG 45+4, a.u.': Combined polarizability gradients for combination $(45a_i^2 + 4b_i^2)$ in a.u.
- 'Raman: CPG 45+7, a.u.': Combined polarizability gradients for combination $(45a_i^2 + 7b_i^2)$ in a.u.
- 'Raman: PCPG 45+4, Å⁴/amu': Pseudo combined polarizability gradients for combination $(45a_i^2 + 4b_i^2)$ in $\text{Å}^4 \cdot \text{amu}^{-1}$ (Default in Dalton)

- **'Raman: PCPG 45+7, Å⁴/amu'**: Pseudo combined polarizability gradients for combination ($45a_i^2 + 7b_i^2$) in Å⁴ · amu⁻¹
- **'Raman: SCS 45+4, SI units'**: Absolute differential scattering cross section for combination ($45a_i^2 + 4b_i^2$) in SI units
- **'Raman: SCS 45+7, SI units'**: Absolute differential scattering cross section for combination ($45a_i^2 + 7b_i^2$) in SI units

For hyper-Raman the literature is less diverse, and the only options (one must be chosen) are therefore

- **'Hyper-Raman: SI complete'**: Straight forward calculation in SI units
- **'Hyper-Raman: Relative'**: Since hyper-Raman intensities often are of very small magnitudes, it is sometimes convenient to report them relative to each other, $I_a^{\text{rel}} = I_a/(I_a + I_b + \dots)$. However, take care with this option if outprojection is not chosen.

For questions, the same holds as for IR, contact me or check my thesis.

In many quantum chemistry code, many different quantities are reported as "intensities". Specify one of the following through **spectroscopy_specification**

- **'Vib modes: 1/m'**: Wavenumbers in m⁻¹.
- **'Vib modes: 1/cm'**: Wavenumbers in cm⁻¹.
- **'Vib modes: 1/s'**: Frequencies in s⁻¹.
- **'Vib modes: ang. 1/s'**: Angular frequencies in s⁻¹.
- **'Vib modes: Eh'**: Vibrational energies in hartree.

5 cauchy_type

When plotting a spectrum, we need to somehow determine a broadening factor, unless we wish only a stick spectrum. The latter can be fine for a single snapshot, but definitely is not a good representation for multiple snapshots. SpectroscPy makes a lineshape based on the Cauchy distribution, more on which can be found in my thesis. For this distribution, a parameter γ needs to be determined. We can do this either by specifying or collecting a default fixed γ value for the whole spectrum, or determining a γ value for each different vibrational mode based on the distribution of the snapshots for that specific peak. In this version you have to choose one of the following options for **cauchy_type**

- Any float or integer, which will be used as the γ parameter. All snapshots will be used in the plotting
- **''** or **'GS'** will both give a default γ parameter, which is based on the maximum wavenumber value. GS stands for given gamma, all snapshots. All snapshots will be used in the plotting
- **'WS'** creates a γ parameter based on the distribution of the snapshots for each given peak. All snapshots will be used in the plotting
- **'Discrete'** creates a stick spectrum with no broadening.

6 names

Names of files to be used as input. Here are two different setups, one for a single snapshot and one for multiple. It is important that the ordering is as specified as below, which is a comment from `SpectroscPy_run`

```
# names contains the names and locations etc for the files.
# Two different setups, either for single or multiple snapshots
# A) Single snapshot
#   0) file location: folder where files are stored. Must have backslash after
#   1) rsp_tensor file
#   2) mol file
# B) Multiple snapshots
#   0) file location: folder where files are stored
#   1) start_snapshot: start number
#   2) end_snapshot: end number
#   3) dal name base
#   4) mol name base
#   5) json name base. If FraME not used, write 'No FraME'
#   6) list of any snapshots you wish to ship, f.eks [4, 48]
```

7 Other inputs

`harmonic_frequency_limits` specify cut-off of harmonic frequencies. When running calculations with solvent, some lower-frequency modes might be seen as contaminated by translation and rotation. These should therefore be removed so as not to confuse your results and to not enter into seemingly higher-frequency overtones and combination bands. If you wish to keep all modes, choose 'Keep all', choose minimum and maximum values as in `harmonic_frequency_limits = [min_x, max_x]`. In the latter case, make sure that you choose a `max_x` value higher than you highest frequency when performing anharmonic calculations. Otherwise you might up severely changing the anharmonic corrected results, also for modes that you have not chosen to remove.

`spectral_boundaries` specify the boundaries of the x-axis. If you wish to plot the whole spectrum, choose 'Keep all', choose minimum and maximum values as in `spectral_boundaries = [min_x, max_x]`.

`T` is the temperature in Kelvin, and is a mandatory argument in `SpectroscPy_run` even though it is only used for Raman. If you are running IR, just put `T` to some random value.

`print_level` is a number speicifying the level of printing. 0 is the lowest possible value.

8 Citations/Aknowledgements

`SpectroscPy` is a program package developed by and containing contributions from

- Karen Dundas
- Magnus Ringholm

- Yann Cornaton
- Benedicte Ofstad

However, we would also like to thank Orian Louant and Per-Olof Åstrand who have been present at each their stage. John Kendrick should be mentioned in relation to the IR units, and the mass dictionary used in `vib_analysis.py` is taken from Gohlke’s `elements.py`³. Also Jógvan Magnus Haugaard Olsen for discussion, technical help and other guidance.

It should also be noted that the functionality in `transform_nc_to_nm.py` and `vib_analysis.py` to various extents replicates or was written on the basis of (or inspired by) corresponding functionality in the Dalton quantum chemistry program and unreleased code associated with Dalton/OpenRSP. In the released Dalton program, the source code files containing this functionality are DALTON/abacus/abavib.F and DALTON/abacus/hergp.F. An overview of the routines in these two scripts for which this applies and the routines in Dalton from which the functionality was replicated, based upon or inspired by, can be found in Table 1.

Table 1: Origin/inspiration of routines

Routine in <code>transform_nc_to_nm.py</code>	Corresponding routine(s)
<code>transform_cartesian_to_normal_one_rank</code>	VIBV3 in DALTON/abacus/abavib.F (and unreleased OpenRSP/Dalton functionality)
<code>transform_cartesian_to_normal</code>	VIBV3 in DALTON/abacus/abavib.F (and unreleased OpenRSP/Dalton functionality)
Routine in <code>vib_analysis.py</code>	Corresponding routine(s)
<code>project_out_transl_and_rot</code>	VIBHES in DALTON/abacus/abavib.F
<code>get_vib_harm_freqs_and_eigvecs</code>	VIBNOR and VIBANA, possibly also VIBCTL, ISOMOL (all former four in DALTON/abacus/abavib.F), VIBMAS in DALTON/abacus/hergp.F

Dalton is released under a GNU Lesser General Public License v2.1, which can be found at <https://gitlab.com/dalton/dalton/blob/master/LICENSE>. The full sourcecode of Dalton can be found at <https://gitlab.com/dalton/dalton>.

9 Brief history of the code

SpectroscPy grew out of a need to collect and systematize many unpublished scripts that had been used by the authors for various projects over the years. Ofstad’s master thesis from 2014¹ is one example, as is Cornaton, Ringholm, Louant, Ruud and others’ work involving corrections to vibrational energy levels and anharmonic infrared and Raman spectra before and around 2016⁴. In the work with my (Dundas’) PhD work it became necessary to use and automatize and these scripts, and I therefore decided to sew it all together to one package and at the same time author more necessary functionality. As these scripts are a natural companion to enable applications of data produced with other software in our community (FraME/OpenRSP/LSDalton), it was also important that they were made readily available and more user friendly.

References

- [1] Benedicte Ofstad. Vibrational motion in molecules. Master’s thesis, Institutt for kjemi, 2014.
- [2] J. M. L. Martin, T. J. Lee, P. R. Taylor, and J. François. The anharmonic force field of ethylene, C₂H₄, by means of accurate ab initio calculations. *J. Chem. Phys.*, 103(7):2589–2602, 1995.
- [3] Christoph Gohlke. elements.py, 2005-2019.
- [4] Y. Cornaton, M. Ringholm, O. Louant, and K. Ruud. Analytic calculations of anharmonic infrared and raman vibrational spectra. *Phys. Chem. Chem. Phys.*, 18(5):4201–4215, 2016.