**dist\_allmodes\_pm.py tutorial**

[I. The basic input of ESD 2](#_Toc14339347)

[II. Step by Step workflow of a simple ESD calculation 3](#_Toc14339348)

[III. Advanced keywords on ESD 4](#_Toc14339349)

[1) The hessflag keyword 4](#_Toc14339350)

[2) Improving the difficult cases 5](#_Toc14339351)

[3) Mixing different excited state methods 6](#_Toc14339352)

[4) Shaping the spectra 7](#_Toc14339353)

[5) Other interesting keywords 8](#_Toc14339354)

[6) Final remarks 8](#_Toc14339355)

1. A diagram of a diagram

   Description automatically generated with medium confidenceStep by step workflow to do vibronic model diabatization

This is the protocol for gmcpt calculations. Steps 1 through 4 need to be completed in order to run the automated diabatization code (Step 5). After you have completed them, you can setup an input template (temp.inp).

**Necessary for composing temp.inp: step2.inp as a gmcpt calculation base file, $VEC group from step3.dat (‘Semi-canonical MO’), $DMO group from step3.dat, and $REFDET from step3.out.**

Make molecular input using wxMacMolPlt’s builder.

1: Copy the coordinates into step 1 template. Run S1, it should be no problem. If it’s CNV, have to input integer after, like CNV 2 and then a single line separating coordinates. If it’s C1 input, cannot have line separating coordinates.

2: From S1’s .log output file, search for ‘GEOMETRY LOCATED’ (case sensitive) and copy the bottom set of ‘all’ coordinates. GMS generates non-symmetry unique coordinates with negative z-values; we need all those included. Change the point group to C1 and copy the $VEC (below ‘OPTIMIZED RHF’) while changing the norb in $guess group. Appropriate combos of orbitals and electrons: 1 doubly occupied, 6 active, 7 electrons. 1 doubly occupied, 7 active, 9 electrons.

3: From S2’s .dat output, just copy and paste “Semi-canonical MOs” $VEC group and rename it as $DMO. Since we have one doubly occupied and like 7 active, delete block with orbital index 1 and rename orbitals 2-8 as 1-7. Delete all the rest. This is accomplished by *./cmo2dmo.x games.dat num\_dmo initial\_dmo*

4: Search for ADIABATIC STATES IN TRANSFORMED ORBITALS: chunks and select the last chunk. Within the last chunk select determinants and Coefp (third column) to compose the $REFDET group. Command: *./refdet\_from\_adiab.x gamess\_gmcpt.out*

5: Exact same as S4, but just change x-coordinate of an atom.

After you did these 5 steps, you can run cat dmo.dat refdet.out >> temp.inp

(and add $data \n c1?)

***python3 dist\_allmodes\_pm.py gamess\_step1.out*** to make the mctdh.op file.

Youtube video: <https://youtu.be/nLIn-N0oB-0>

IMPORTANT! Nov 27, Toby: you need to put a dash sign beside first $data group, else it GMS will only take first $data group and not the second one needed for machine gun calculations. No need to pipe in phase.out b/c refdet script already handles that.

For composing the temp.inp, it differs from base step2 as well because it seems like $VEC is from Semi-canonical MOs instead of the usual Optimized RHF.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**6**

**5**

**4**

**3**

**2**

**1**

A black text on a white background

Description automatically generated

Above is the basic input of ESD, where we compute the vibronic progression for the 1st root of the molecule test.xyz using the Adiabatic Hessian (AH) algorithm. Let’s go into details:

1: The ESD program is called by using ESD(X), where X is the type of spectra you want to compute i.e. ABS (absorption), FLUOR (fluorescence), PHOSP (phosphorescence), (RR) resonance Raman…

2: In order to compute luminescence spectra, one excited state method is necessary. Here the TD-DFT approach has been selected but any available method in ORCA can be used. On top of the number of roots, the keyword “iroot” is used to select which root will be used within the ESD module to obtain the spectra.

3: The keyword “GSHESSIAN” and “ESHESSIAN” correspond to the hessian file computed for the GS and the ES of interest (here root 1). We will explain later how to obtain these files.

4: The “hessflag” keyword selects which algorithm will be used within the ESD module. Here the Adiabatic Hessian (AH) algorithm is used where both GS and ES potential surface are required. Others algorithms using approximations are available and explained later.

5: Several keywords exist to tune the spectra. Here “unit” changes the unit of the spectra (cm-1, nm or eV) while “lines” changes the line shape (Lorentz, delta, Gaussian or Voigt).

6: The name of the coordinate file. For ABS, FLUOR, PHOSP and RR, the optimized **GS structure** has to be used.

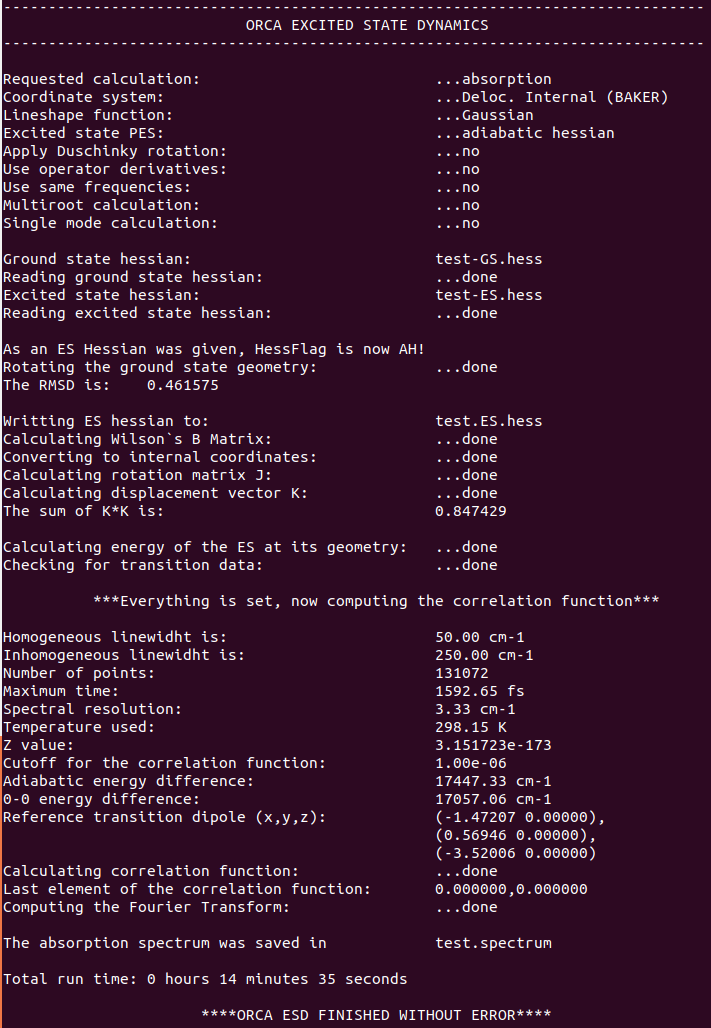
1. Step by Step workflow of a simple ESD calculation

**1-** GS optimization + frequency calculation (keyword “FREQ” for analytical frequencies)

**2-** ES optimization of the desired ESD + numerical frequency calculation (keyword “numfreq”)

**3-** ESD calculation using the input written above

**4-** Output



**1**

**7**

**6**

**5**

**4**

**3**

**2**

1: Some reminders from the ESD input, here ABS calculation, Gaussian line shape, AH algorithm, and name of the hessian files

2: The RMSD value between the GS and ES structure, if this value is too high the calculation might break.

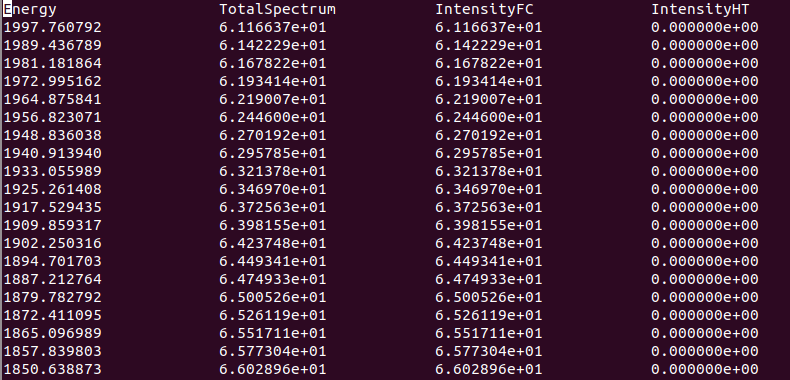
3: For each ESD calculations, the program prints a file containing the ES hessian if one needs to run again a calculation. It is not very useful in our case because we have plugged in one ES hessian but for other ESD algorithms it is important.

4: The sum of the K\*K is the most important value to check, it is basically the trust factor of the ESD calculation. It will print a message if K\*K > 8 meaning that the calculation is breaking and the results cannot be trusted. I recommend to check already at K\*K >5 if nothing wrong is happening. There are ways to decrease this value for troublesome cases.

5: Some parameters concerning the shape of the spectra, i.e. linewidth, spectra resolution and selected temperature. All these parameters can be changed.

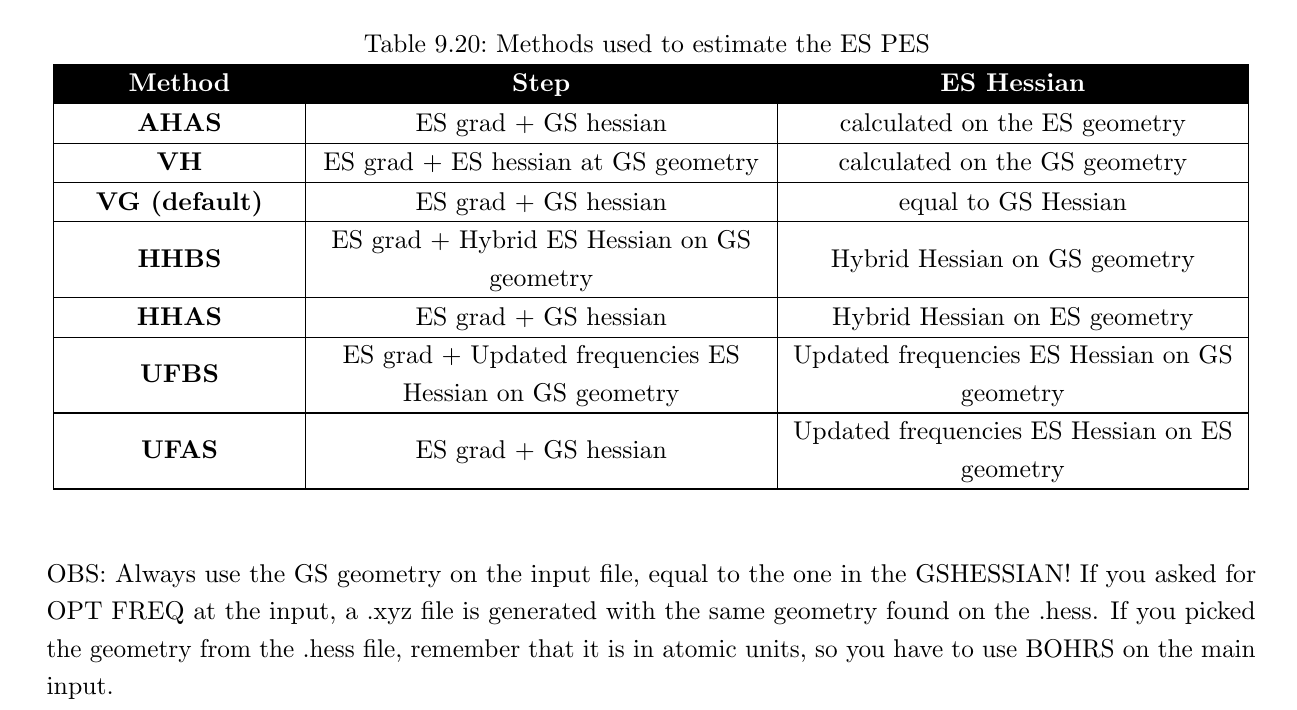
6: The adiabatic and 0-0 energies computed using the vertical excitation selected from the TDDFT module. The computed transition dipoles using the transition dipole selected from the TDDFT module. The transition dipole is defined as .

7: The name of the file with the printed spectra with the following information. Energy is the x axis, Total spectrum is the sum of the contribution from the Franck-Condon approximation (IntensityFC) and from the Herzberg-Teller part (IntensityHT). In order to plot the spectra on gnuplot, the first line has to be removed.



1. Advanced keywords on ESD
2. The hessflag keyword

In the hessflag keyword you are defining how you want to treat the ES potential energy surface. Here is a very clear table from the ORCA manual:



In this table all the approximations used to estimate the ES PES are listed. I will develop here only the approximations I have used so far.

* Adiabatic hessian (AH): we have used this one in the model case. It corresponds to a zero approximation model where both GS and ES PES of the system at their equilibria positions are provided. This is the model to use in order to have the best precision towards experimental measurements. However, this is the more time consuming method as it requires both hessians (GS+ES). Moreover, this approach works only on semi-rigid molecules. Indeed, if the ES geometry is too different from the GS one, the K\*K sum will be large and the calculation fails.
* The default hessflag is the Vertical Gradient (VG). In this approximation, the GS hessian is set to be equal to the ES one. Then, the geometry in the ES is updated following the mode of the GS hessian. This method is the fastest one to do while giving a good representation of the spectra. It is also very useful when molecules are very flexible.
* The last hessflag I will discuss is the Adiabatic Hessian After Step (AHAS). Here one optimization step is done on the ES PES starting from the GS geometry and the ES hessian computed on the new geometry. It captures some of the information of the ES PES while staying close to the GS geometry (good for flexible molecules). However it is time consuming due to the ES hessian calculation.

1. Improving the difficult cases

It often happens that the ESD calculation breaks because of a large K\*K sum. To deal with this issue, several keywords can be used.

* First, instead of AH using a hessflag approximation like VG can improve the results but some information of the ES PES will be lost.
* The second way is to play on the frequencies of the GS/ES modes. Some negative (imaginary) frequencies can be obtained during the hessian calculations leading to some troubles. First, by increasing the printlevel (at least 3), the list of the frequencies used in ESD are printed. Then the keyword “IFREQFLAG” let you choose what to do with these frequencies. You can either put them positive (“POSITIVE”, default), leave them (“LEAVE”) or remove them (“REMOVE”). Finally you can also filter all the frequencies below a certain value. You have to have “TCUTFREQ” and an integer. I find it the best way to decrease the K\*K sum. Be careful not to remove some important frequencies. For troublesome case, a value of 400 is often good to improve the results.
* Finally, the last way to decrease K\*K sum is to change the coordinate system of the normal modes using the “COORDYS” keyword. By default, the value internal coordinates are used, but other can be chosen like Cartesian (“CART”). Unfortunately, I never managed from my experience to decrease the K\*K sum using this.

1. Mixing different excited state methods

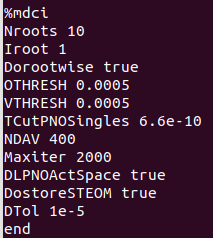
To have a better description of your system, one often uses more accurate methods than DFT/TDDFT. In my case, the DLPNO-STEOM-CCSD method is better than TDDFT to compute the vertical excitation of fluorescent dyes. However, DLPNO-STEOM-CCSD cannot be used to compute geometries optimization and/or frequencies.

The solution is to ask ESD to use the DLPNO-STEOM-CCSD values associated with the DFT/TDDFT frequencies/geometries.

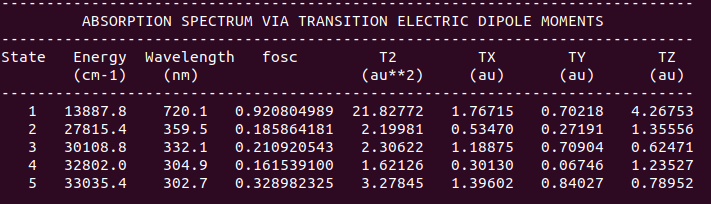
The workflow of this approach can be summarized as the following if the AH algorithm is used:

1/ DFT opt +freq, TDDFT opt + numfreq

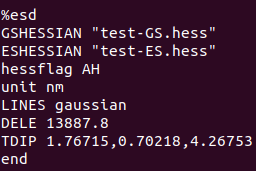
2/DLPNO-STEOM-CCSD vertical absorption energy



Here is the standard input I am now using for DLPNO-STEOM-CCSD (still in benchmark). At the end of the calculation, you have to identified the root of interest (here 5) as well as the following information.



The energy and the transition dipole (TX,TY,TZ) can be plugged directly inside the ESD module, the later becoming:

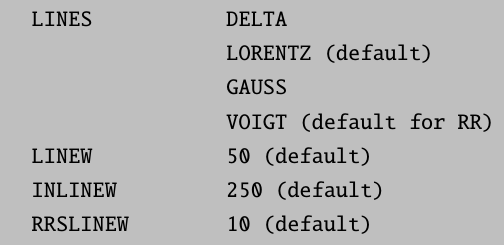


The “DELE” keyword corresponds to the adiabatic energy computed in ESD (see 6 in Workflow) and TDIP to the transition dipole of the selected transition (TX,TY,TZ or TX.re,TX.im,TY.re,TY.im,TZ.re,TZ.im). These two data will be used in the ESD calculation without taking into account the computed transition specified in the tddft module. This can also be used for any hessflag.

**Warning:** This is not the optimal workflow when used together with the AH algorithm. Indeed, the vertical excitation and transition dipole are not the same as the adiabatic one. Thus for the AH algorithm, I replaced the tddft block by the mdci one written above. The calculation will take more time but in the end the correct adiabatic energy and dipole are printed and can then be used later.

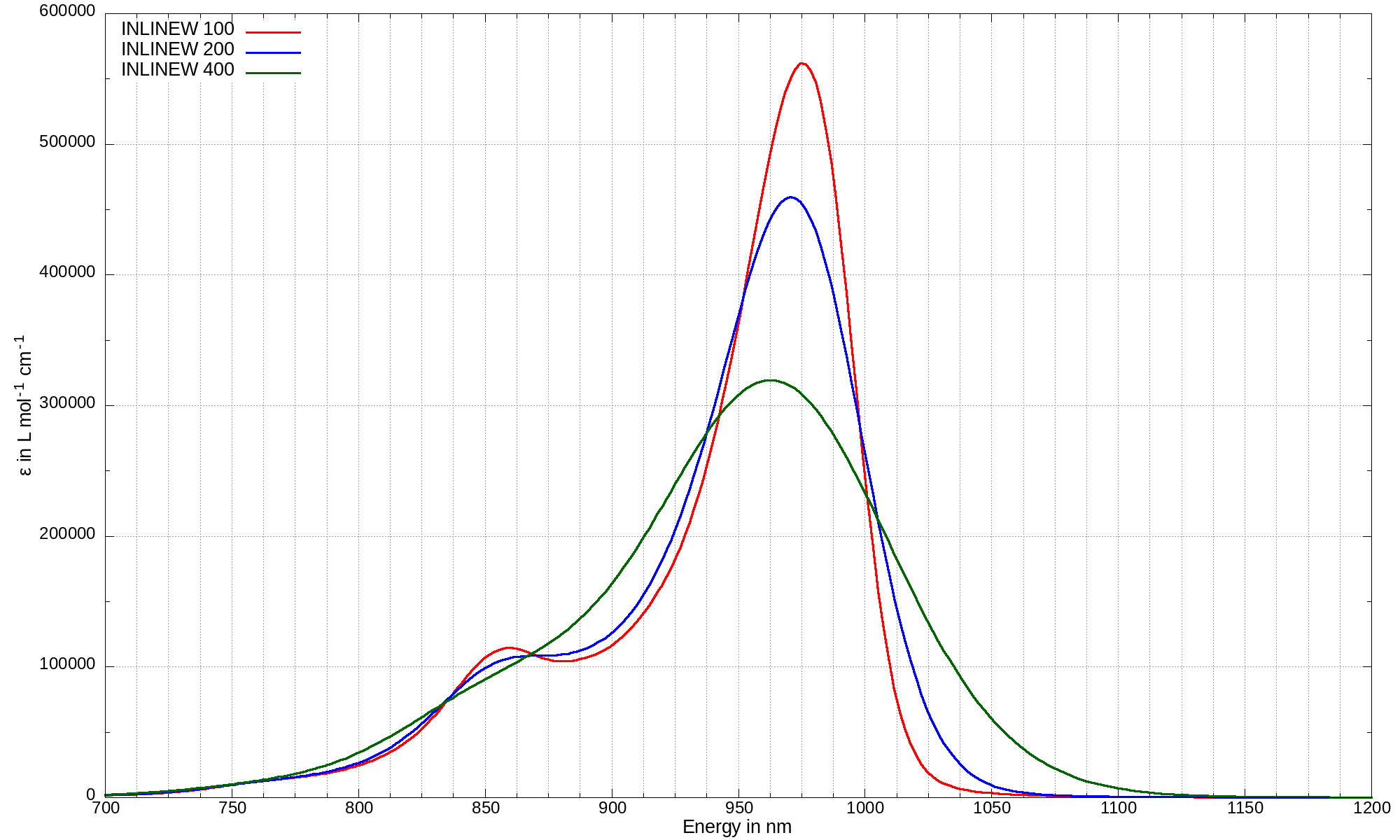
1. Shaping the spectra

Now that the spectrum is created we can play on the line shape and line width to reproduce the experimental one. There are several keywords to play with this. We talk earlier about the different type of lines. Here is a recap:



The LINEW and INLINEW keyword are used to change the full width half maximum (FWHM). Depending of the line shape selected both keywords may not have an impact. For the LORENTZ shape, the LINEW keyword has to be changed while for the GAUSS one this is the INLINEW.

Here is an example for one absorption spectrum using the GAUSS line shape and several values of INLINEW.



1. Other interesting keywords

I want to mention other keywords that are useful within the ESD module.

* “MULTISTATE”: the multistate keyword allows the computation of a spectrum while selecting multiple roots. To do it, remove the iroot from the %tddft block” and the “STATES” keyword in the ESD block together with the roots you want to select i.e.

STATES 1,2,3,4,5 for computing the 5 first roots of a molecules

* Studying dark transitions: To study forbidden transition within ESD, the keyword “DOHT”, i.e. compute the Herzberg-Teller part corresponding to vibronic coupling. Warning, it will take much more time

DOHT true

1. Final remarks

The ESD module is very well described in the ORCA manual in both “Running typical calculations” (Excited state dynamics) and “Detailed documentations” (More on the Excited State Dynamics)

**Warning:** For other type of spectra (PHOSP/RR) others mandatory keywords/files are necessary, please check the manual.