

LAB MANUAL & PROCEDURE

$C_4H_3F_7O$ (1,1,2,2-Tetrafluoroethyl 2,2,2-trifluoroethyl ether, TFTFE)

Aziell Paul S. Limbo, Advisor: Dr. Karine Le Bris
California State University, Los Angeles, Department of Physics and Astronomy, Los Angeles, CA
(Dated: January 2026)

Contents

GitHub Repository link	1
I. Installation	1
II. Building Molecule & Input File	1
III. Extracting Raw Spectral Data From ORCA Output	2
IV. Calculating Population Percentage	3
V. Gibbs Free Energy Verification & Reliability Test	3
VI. Generating Theoretical Cross Section for Molecule	3
Broadening Lineshape Functions	5

GitHub Repository link

https://github.com/Limbotime2/IR_LAB_2027/

I. INSTALLATION

Install the following (this is assuming Ubuntu is already installed, if needed, install Ubuntu. There are many different ways to install it if needed at all):

- **Avogadro** This program is used for "drawing" your molecule. Can be done using this link <https://avogadro.cc/>
- **ORCA** This software is the real meat of it all. Most calculations will be done here. I installed it onto Ubuntu using this tutorial: <https://spoken-tutorial.org/watch/ORCA--Computational+Chemistry/Installation+of+ORCA+on+Linux/English/>

NOTE: I admit it's a little dated, but the instructions should be similar to the time of writing this manual. I have ORCA 6.1.0 installed on my machine and used the very same tutorial. Let's hope not much has changed as time goes on.

GOOD PRACTICE: It's also not a bad idea to read the ORCA documentation and run your first calculation at this point. Documentation can be found here: <https://www.faccs.de/docs/orca/6.0/manual/>. For my first run, I read the entire beginning up until I made my first Hello Water calculation as described. Documentation also tells you the "official" way of installing ORCA, but it's not easy to follow if you're not too sure what you're doing.

I am on Windows, and access Ubuntu through the cmd prompt. As for the text editor, I use vim, but any editor can be used.

II. BUILDING MOLECULE & INPUT FILE

TTFE has three conformers (I only know this from this source: <https://pubchem.ncbi.nlm.nih.gov/compound/164596>). This means three separate files will need to be created via Avogadro. In turn, we will run ORCA three times separately for each conformer.

Learn how to draw molecules in Avogadro using this online tutorial: <https://www.youtube.com/watch?v=zLBz8WiXbAs>. Personally, I uncheck the "Adjust Hydrogens" setting so I have complete control of what I am inputting - just don't forget your hydrogen atoms where needed!

Once a rough 3D drawing is made, the geometry of the molecule needs to be optimized. It's unlikely you've drawn it the exact mathematical way if you used just your mouse to draw your molecule. Go to:

Extensions >> Optimize Geometry

It should now look like a legit molecule with mathematically correct angles of bonds and atoms. Assuming everything looks correct, feel free to use this visual in final papers and posters by exporting it as a png (I totally did this in Fig. 1).

NOTE DEFINITELY save this file! Avogadro saves its own files using the .cml extension.

Now it's time to set everything up for ORCA. Go to:

Extensions >> Orca >> Generate Orca Input...

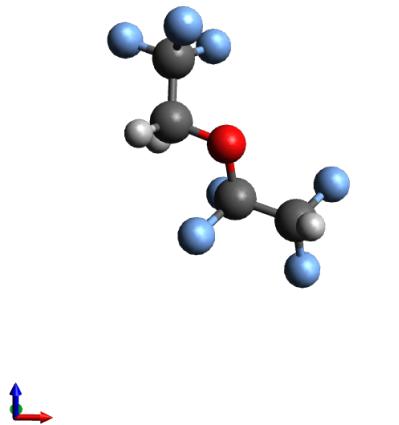


FIG. 1: An exported png file of what I am currently calling the "first conformer" of TFTFE

Hopefully by this point you've learned a thing or two about how to configure your calculations from the **Hello Water** example described in the documentation for **ORCA**. This time you edit the input file within the **Orca Input Parameters** pop-up window instead of manually writing an input file as you may have done with the **Hello Water** example using some arbitrary text editor. Around line 4 (where the exclamation point is) is where you tell **ORCA** what to calculate and more importantly *how* to calculate. Anything following a # symbol is a comment. For my first example, I used this input:

```
.
.
.
! B3LYP def2-SVP OPT FREQ
.
.
```

NOTE: More of a confession really. At the time of writing this, I don't fully understand the various DFT and basis set calculations yet. I have arbitrarily chosen the **B3LYP** method as it was referenced in my prior reading into doing any kind of calculations. **def2-SVP** was arbitrarily chosen as it is one (of the many) method(s) used in Dr. Le Bris's previously published papers. Hopefully this note gets deleted very soon so I may say otherwise.

The **OPT** and **FREQ** directions are taken as we are interested in furthering optimizing the geometry of the molecule (apparently Avogadro does not do this fully,

think of it as an added step to ensure all calculations are accurate) as well as calculating the various frequencies of the molecule.

Around line 6 (where the asterisk, *, is) shows the xyz coordinates, charge (usually zero), and multiplicity (usually 1) - in that order. I've left this setting for all conformers (for now):

```
* xyz 0 1
```

Now generate the input file, and ensure the file is saved as an .inp file. I recommend saving it as (or something similar):

```
[molecule name].inp
```

Now it's time to run **ORCA**! Now run your input file as you may have done in the **Hello Water** example. For me, I run this command in **Ubuntu**:

```
orca [molecule file name].inp > [your molecule file name].out
```

It took me 30 minutes to an hour for each conformer. So it may take a while!

III. EXTRACTING RAW SPECTRAL DATA FROM ORCA OUTPUT

Ensure that the IR spectrum data is available in the **ORCA** output file. It should have a header followed by data like this:

```
.
.
.
-----
IR SPECTRUM
-----
Mode freq eps Int ...
cm**-1 ...
-----
[DATA]
.
.
```

Now the goal here is to generate a spectra graph for your molecule. Ideally we want to extract this data into a 2 column format so we can plot it. This can be done by hand, but I've made a **MATLAB** program that

accomplishes this by scanning and extracting the data and outputs it into a .dat file. The program is entitled `IR_DATA_EXTRACT.m`. The source code is provided in the accompanying GitHub repository.

For plotting the data, I used `xmgrace`. However, any plotting software can be used. The output of this is in Fig. 2

IV. CALCULATING POPULATION PERCENTAGE

Now we wish to find the population percentage of each conformer. It's a simple calculation that needs no program to do. For a given conformer, i , the population percentage is:

$$\text{Population\%}_i = \frac{g_i \cdot \exp(-\frac{G_i - G_{min}}{kT})}{z} \quad (1)$$

where g_i is the degeneracy value for the i th conformer, and z is the partition function:

$$z = \sum_i g_i \exp(-\frac{G_i - G_{min}}{kT}) \quad (2)$$

k is the Boltzmann constant and G_{min} is the lowest Gibbs free energy from each conformer, and this value is provided by `ORCA`. T is temperature - and we assume room temperature for now at the value 298.15K. So each G_i corresponds to the Gibbs free energy for the i th conformer. For TFTFE there are three conformers, so there are three Gibbs free energies. To ensure the population percentage calculation is correct, the sum of these percentages should of course be equal to 1. This is all calculated in `pop_perc.m`. All results are displayed in Table I.

NOTE `ORCA` gives each G value in Hartrees instead of Joules! So the conversion $1He = 4.359748199E - 18J$ must be applied.

NOTE Mathematically calculating the degeneracy value, g_i , requires knowledge of group theory, a completely stand alone field. However, for relatively small molecules, it is often easier to just find the degeneracy by rotating each group and see if that gives you the same molecule by mirror image or 180-degree rotation of the whole molecule.

- If it is the same molecule, you now have two states sharing the same energy $\rightarrow g_i = 2$
- If it is a different molecule, you have a new rotamer with its own harmonic modes and bands strength.

V. GIBBS FREE ENERGY VERIFICATION & RELIABILITY TEST

`Gaussian`, another similar quantum chemistry calculation software, apparently outputs inconsistent and unreliable values for the Gibbs free energy value, G . It is absolutely worth checking if `ORCA` does the same. Otherwise, the results of population percentages will not be accurate. The best way to see if `ORCA` provides reliable values for G or not is to check the G values for different basis set calculations. It should absolutely be similar or the same. If not, then enthalpy values will be used in place of G for equations (1) and (2). To test `ORCA`, two new runs will be made, but this time I'll use the cc-pVDZ to cc-pVTZ basis sets and compare the G values of these runs. B3LYP will also be used for both runs. Results are shown in Table II

VI. GENERATING THEORETICAL CROSS SECTION FOR MOLECULE

At this time, it is assumed that the Gibbs free energy and corresponding population percentage values are good enough. Based off of Table II, I may have to switch to using the enthalpy values and their corresponding population percentages instead.

So by this point, we have the theoretical relevant wavenumbers of interest, the integrated absorption coefficients, and conformer population weights. It's time to put it all together. The goal is to calculate the cross section. The roadblock is that we need to convert the absorption coefficients (y-axis) into the cross-section. A MATLAB code has been provided (authored by Dr. Le Bris), so the following are the mathematical notes based off that. That file is entitled `generate_spec_conformers.m` and has been uploaded to the GitHub repo located in the folder `KLB_m_source_codes`. However, actual implementation/calculations about to be described in this section are used in the program entitled `THEORETICAL_CROSS_SECTION.m`. The results are visualized in Fig. 3.

At present, the coefficients calculated by `ORCA` suggests that we only have discrete values, and we need continuous values as when it comes time to gather actual experimental data, the values will be given as a continuous spectrum (cross section). In other words, real spectra have broadened peaks instead of infinitesimally small peaks shown in Fig. 2. Mathematically broadening these infinitesimally small peaks is done via three different methods, Gaussian (not to be confused with `Gaussian`, the other computational quantum chemistry software mentioned earlier), Lorentzian, and Voigt. In the language of math and in general,

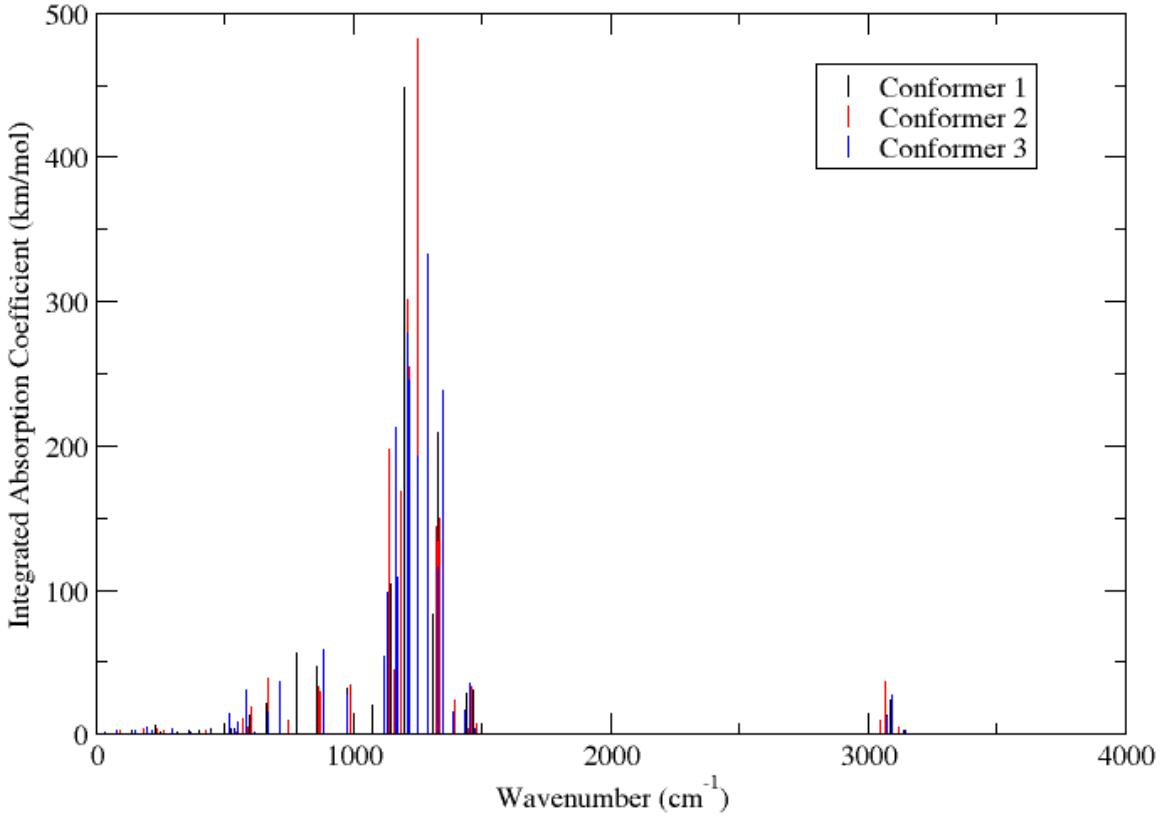


FIG. 2: Raw spectral graph of TFTFE produced by `xmgrace`. Data is directly from ORCA. I've plotted all coefficients as infinitesimally small peaks as these are only discrete values, obtaining continuous values are described later.

Conformer #	g_i	Gibbs Free Energy (J)	Population Percentage	Enthalpy (J)	Population Percentage
1	1	$-4.04248581 \times 10^{-15}$	83.2358451254%	$-4.0422778686 \times 10^{-15}$	97.17891360%
2	1	$-4.04245942 \times 10^{-15}$	0.136797200712%	$-4.0422433201 \times 10^{-15}$	0.02200837%
3	1	$-4.04247918 \times 10^{-15}$	16.6273576739%	$-4.0422632667 \times 10^{-15}$	2.79907803%

TABLE I: Table showing the Gibbs Free Energy in Joules for each conformer. Method used: B3LYP def2-SVP. $G_{min} = G_1$. Enthalpy values and their corresponding population percentages are also included for this run in case the Gibbs free energy values calculated are not reliable. See section **Gibbs Free Verification & Reliability Test** for more info.

$$\sigma_{\text{total}}(\nu) = \sum_{i=1}^N w_i \cdot \sigma_i(\nu) \quad (3)$$

where σ_{total} is the cross section as a function of the wavenumber (e.g. what we want!!!), N is the number of conformers, w_i is the population percentage for the i th conformer, and $\sigma_i(\nu)$ is the spectrum of conformer i alone:

$$\sigma_i(\nu) = \sum_{j=1}^M S_{i,j} \cdot L(\nu - \nu_{i,j}) \quad (4)$$

where M is the number of discrete present for that particular i th conformer, $S_{i,j}$ is the integrated absorption coefficients, and $L(\nu - \nu_{i,j})$ is the broadening lineshape function (Gaussian, Lorentzian, or Voigt).

So in the special case where there is only one conformer:

Conformer	cc-pVDZ			cc-pVTZ		
	G.F.E. (J)	Pop. Perc.	Enthalpy (J)	Pop. Perc.	G.F.E. (J)	Pop. Perc.
1	-4.0457459125 $\times 10^{-15}$	91.0146294371%	-4.0455266677 $\times 10^{-15}$	83.86733969%	-4.047156367 $\times 10^{-15}$	97.2880045024%
2	-4.0457169541 $\times 10^{-15}$	0.08014994706%	-4.0454995632 $\times 10^{-15}$	0.11587021%	-4.0471263421 $\times 10^{-15}$	0.0666877814306%
3	-4.0457363444 $\times 10^{-15}$	8.90522061589%	-4.0455198526 $\times 10^{-15}$	16.01679010%	-4.0471415104 $\times 10^{-15}$	2.64530771622%

TABLE II: Table comparing the G values with it's corresponding calculated population percentage between the cc-pVDZ and cc-pVTZ basis sets. Enthalpy values and their corresponding population percentages are included. B3LYP is still used for both basis sets. $G_{min} = G_1$ for both basis sets.

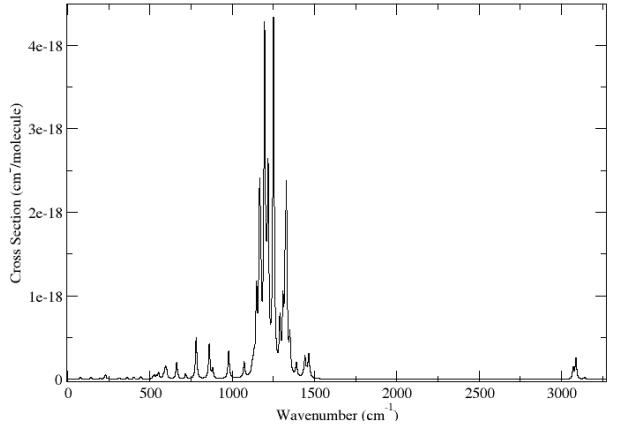


FIG. 3: Raw cross section output from `THEORETICAL_CROSS_SECTIOn.m` for B3LYP def2-SVP method. This here would serve as the theoretical data and will be compared to the upcoming experimental measured spectra.

$$\sigma_{\text{total}}(\nu) = \sigma_1(\nu) = \sum_{j=1}^M S_j \cdot L(\nu - \nu_{i,j}) \quad (5)$$

Broadening Lineshape Functions

We'll start with the Gaussian lineshape. The normalized Gaussian function is from the provided `generate_spec_conformers.m` MATLAB function.

$$L_G(\nu - \nu_0) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(\nu - \nu_0)^2}{2\sigma^2}\right) \quad (6)$$

where σ is the standard deviation (defined as `sigma=(1/2.355)*FWHM` in code), ν is the wavenumber to be evaluated, and ν_0 are the known wavenumbers obtained through quantum calculation (from ORCA). The FWHM is the Full Width at Half Maximum of the band in cm^{-1} . This code recommends to begin at a value of 10 to start, then it can be adjusted. However, I've found that the HITRAN documentation (<https://hitran.org/docs/definitions-and-units/>) has a lot more to say about that. I will definitely be exploring this value at a later date, but for now I will use 10 as the code suggests.

The Lorentzian function is:

$$L_L(\nu - \nu_0) = \frac{1}{\pi} \frac{\gamma}{(\nu - \nu_0)^2 + \gamma^2} \quad (7)$$

where γ is the Half Width at Half Maximum (HWHM) which is just FWHM divided by 2. Once again, the same HITRAN documentation has a bit more to say about this value.

As for the Voigt lineshape, HITRAN doesn't say much about it, and doesn't even give its own equation for it. It does, however, note that when using the HITRAN database, the Voigt line shape is usually assumed. The Gaussian lineshape function is a little different by HITRAN, yet the Lorentzian lineshape function is the same by HITRAN. From the provided code,

$$L_V(\nu - \nu_0) = \text{ratio} \cdot L_L(\nu - \nu_0) + (1 - \text{ratio}) \cdot L_G(\nu - \nu_0) \quad (8)$$

where the ratio (according to the code) is the fraction Lorentzian versus Gaussian in the Voigt function. A value of 1 would correspond to a pure Lorentzian function. If a ratio is not specified, the default ratio value is 0.5.

Now it brings the question, which function does one use? HITRAN suggests that in the lower atmosphere, pressure broadening of spectral lines dominates and a Lorentz profile can be used. Conversely, in the low-pressure environment of the upper atmosphere, Doppler-broadening dominates the line shape and a Gaussian profile can be used. I'm going to be completely honest, I don't know where my molecule, TFTFE, has been detected other than it has been mentioned that it is used in batteries. So I'll assume a Lorentz profile because I'm assuming there aren't many batteries in the upper atmosphere that is causing concern.