Lab 2: Scanning a Potential Energy Surface
CHE 525/PHY 567, Spring 2021, Stony Brook University

1 Introduction

A collection of many electronic structure calculations can be used to construct a potential energy surface, or for large molecules at least projections of a potential energy surface on to some reduced number of dimensions (e.g. reaction coordinates). For chemical dynamics happening in the electronic ground state (i.e. most of chemistry), the ground state potential energy surface contains the critical information for calculating equilibrium constants, reaction rates, etc. Thus calculating potential energy surfaces is one of the most important tasks of quantum chemistry.

In this electronic structure lab exercise, you will explore the potential energy surface of azomethane. Azomethane has two stable geometries, a *trans* configuration and a *cis* configuration which you will try to find in this lab. Starting from a globally optimized geometry calculated with Psi4's `optking` module, which you learned how to use in Lab1, you will calculate the potential energy for different values of the torsional coordinate around the central N-N bond (the C-N-N-C dihedral angle), mapping out the potential energy surface of the isomerization reaction. You will do this in two different ways. First, you will rotate the molecule rigidly—leaving all coordinates besides the dihedral angle at their equilibrium values. Second, you will perform a series of constrained optimizations—holding the dihedral angle constant at a series of values, and relaxing the remainder of the coordinates. This second method will give us the true minimum energy path (within the HF/6-31G approximation), while the first method is merely an approximation to the minimum energy path.

2 Procedure

The procedures and Python code you need to do this lab were mostly covered in Lab0 and Lab1. Example calculations for hydrogen peroxide that track the four steps in this exercise are provided:

1. `Lab2starter_1_GeometryOptimization.py` - Initial geometry optimization.
2. `Lab2starter_2_RigidRotation.py` - Performs a rigid PES scan and outputs the results.
3. `Lab2starter_3_RelaxedRotations.py` - Performs a relaxed PES scan and outputs the results.
4. `Lab2starter_4_MakePlots.py` - Creates a plot of the rigid and relaxed PES scans, for comparison.

Please read through these sample scripts carefully, incuding all comments, and run them yourself to become familiar with the procedure. The comments in the scripts provide valuable information about technical details that is complimentary to the information provided in this document.
You will work through these same steps for azomethane. In words, here are the steps:

1. Use Molden to generate the Z-matrix of a planar *trans* form of azomethane (CH3-N=N-CH3). It will be more convenient later if you have added all four non-hydrogen atoms first, and then the six hydrogen atoms afterwards. As you add each atom, are they in the correct geometry? You can check this by manually changing it in Molden. Pro tip: Slightly breaking the symmery of the initial guess structure, by avoiding perfect 180 degree angles, for example, can help the optimizer find the true lowest energy structure.
2. Optimize the geometry of *trans* azomethane by performing a regular (unconstrained) optimization using the `scf/6-31G` method starting from your Molden input. Use `Lab2starter_1_GeometryOptimization.py` as an example to get you started.
3. From your initial optimized geometry, generate a Z-matrix as a starting point for a 'rigid' torsional angle scan. For this you need a definition of the Z-matrix that let's you change the dihedral angle as a parameter. You can do this quite easily in Psi4 by manipulating the string argument passed to `psi4.geometry()` as shown in the example in `Lab2starter_2_RigidRotation.py`. Calculate the energy of at least 10 angles between the initial 180° and 0°, keeping all the other relative coordinates fixed. Produce molden files at each interation (as done in the example), and inspect them using molden. Does the geometry look as expected? If not, carefully inspect your Z-matrix and work to fix the problem(s).

4. Now re-initialize your molecule and perform a series of constrained optimizations at fixed dihedral angles. You can do this several ways. For example, you can try and to do this by feeding the geometries you used in the rigid scan to `frozen_dihedral` optimizations, similar to what we did in Lab1. `Lab2starter_3_RelaxedRotation.py` provides an example of this approach to get you started. If you explore the Psi4 manual, you will find other ways to perform the same calculation. Again, produce molden files at east step (as done in the example), and check to make sure that all went smoothly. Be careful that the way you define the Z matrix is consistent with your choice of constraint you define via `frozen_dihedrals`! The order in which you list the atoms in the constraint matters, and it may not be the same as it was for hydrogen peroxide!

3 Saving and plotting data

Of course, *everything* is saved in your .dat file, and you can always manually go in to that file, find the data you need, and cut/paste it into your favorite analysis/plotting program to produce the deliverables below. However, it is convenient to make use of the Psi4 Python API which already has functions like `psi4.optimize()` return to you the energies etc. as variables and Python objects. You can plot the results directly from your code that runs the calculation using `matplotlib`, however for longer calculations this is not recommended, since every time you want to change something about the plot (e.g. the marker used), you would have to run the calculation again.

Instead, what you can do is have Python save the data for loading later and plotting. There are several ways to do this at various levels of sophistication. For examples, you can look at the file `CHE525_FileIO.py` which gives examples of saving and reading arrays (and more) to a file using `numpy`, `pandas`, and the powerful hdf5 interface `h5py`. In the starter code, you will see the data saved to a spreadsheet (in CSV format) using `numpy` using just a few lines of code.

For plotting, you can use whatever plotting tool you like. Reloading your data into Python can be useful because then it is easy to do subsequent calculations on it and also the `matplotlib` library of Python can make sophisticated plots. `Lab2starter_4_MakePlots.py` gives example Python code that loads the raw results from the starter calculation using `pandas`, converts the data to `numpy` arrays, performs a few manipulations to the data (e.g. converting energy units), and plots the results. Note that standard spreadsheet software will also load CSV files like those produced by the sample scripts.

4 Deliverables

- Make a table of the energies from these relaxed optimizations as a function of angle. Make a figure showing both the rigid and relaxed potential energy surfaces on the same axes. Make another figure zoomed in on the relaxed optimizations. Convert all of the energies from Hartree (atomic units) to a more widely used unit (such as kcal/mol or eV) and shift all of the energies by a constant such that the lowest energy is zero. Also, choose the y-axis scales wisely—we should be able to see what is happening to the energy as the molecule rotates.

- Describe in words the difference between the rigid and relaxed PESs in your plots.

- At which point is the error of the rigid surface largest relative to the relaxed one? Look at the geometries where the differences are largest. Can you see why relaxing the geometry was important here? Adding pictures of the geometries will likely help you explain more clearly.

- Based on these calculations, provide an estimate for the equilibrium constant of this isomerization reaction as computed at the HF/6-31G level of theory. Justify your answer. You may assume the change in entropy for this reaction is zero.

- Based on these calculations, provide an estimate for the activation energy of this isomerization reaction as computed at the HF/6-31G level of theory. Describe a procedure for improving your estimate. Justify this answer in words (and, if you want, pictures).