MESS Input Generator Python Script Tutorial

Motivation: Generating MESS Inputs requires gathering information (energies, rotor scans, geometries, frequencies) from each pathway on a mechanism of interest. Not only is this time-consuming, but also it can be hard to keep track of everything. The goal of these scripts is to create a general structure for a MESS input, by pulling information such as geometries, frequencies, and rotors from various files to facilitate the process. I will also create and populate an *Energies Spreadsheet* with energy optimizations used to extrapolate to the infinite basis set CCSD(T)/cc-pV∞Z.

There are 5 scripts that you will be using for this:

- get_geometry.py
- geometryInfo.py
- messParsing.py
- writeMessFile.py
- get_energies.py

Step 1: Place *all* m062x, rotor and higher energy files in their corresponding *pathway directories* depending on the type of pathway they belong to. These pathways can be parent well, well, transition state and bond fission.

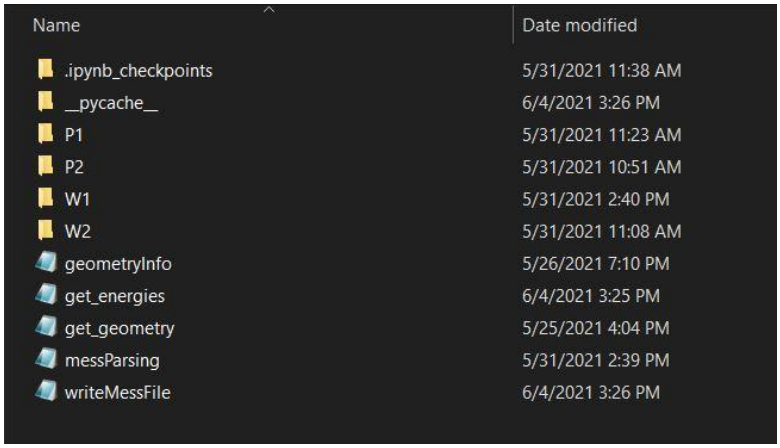

Figure 1: File structure for Mess Parsing. Place all *.py* files in the same directory as the *pathway directories (W1, W2, P1, P2, etc).*

Parent Well (W1): The parent well *pathway directory* will contain the m062x, E, E1 and rotor files for the parent molecule (the filenames must begin with 'W_' and end with '_m062x.log'/ '_E.log'/'_E1.log'/'_D#_rotor.log), it should be named W1, and there should only be one.

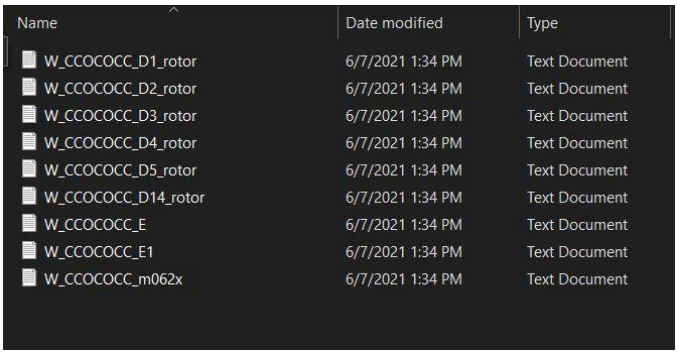

Figure 3: Files inside the W1 *pathway directory* from Figure 1.

Well (W2): The well *pathway directory* will contain the m062x, E, E1 and rotor files for the well and TS going from the parent well to the well (the well's filenames should begin with 'W_' and end with '_m062x.log'/'_E.log'/'_E1.log'/'_D#_rotor.log' – the TS filename should begin with 'TS_' and end with '_m062x.log'/'_E.log'/'_E1.log'/'_D#_rotor.log'). There can be multiple well *pathway directories*, named W3, W4, W5 and so on.

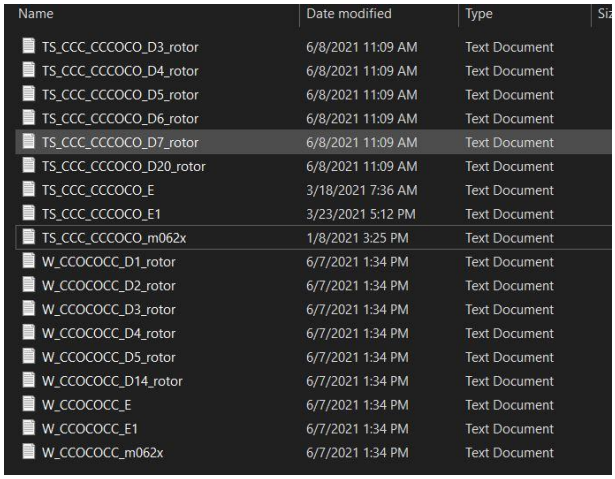

Figure 4: Files inside the W2 *pathway directory* from Figure 1.

Transition State (P1): The Transition State *pathway directory* will contain the m062x, E, E1 and rotor files for the corresponding TS (filenames must begin with 'TS_' and end with '_m062x.log'/'_E.log'/'_E1.log'/'_D#_rotor.log'). Additionally, it will also contain a *fragments* folder that will contain the individual m062x, E, E1 and rotor files for the 2 products being formed (these files must end with '_m062x.log'/'_E.log'/'_E1.log'/'_D#_rotor.log'). Note, if one of the fragments is H, you just need to include all files of the other fragment, the code will assume that the missing fragment is H and fill the MESS input accordingly. There can be multiple Transition State *pathway directories*, named P1, P2, P3 and so on.

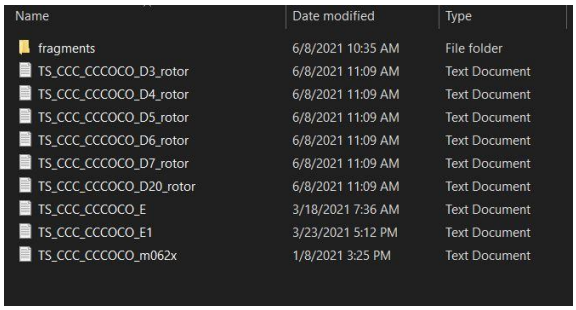

Figure 5: Files inside the P1 *pathway directory* from Figure 1.

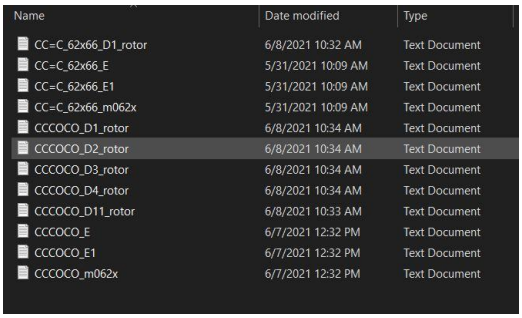

Figure 6: Files inside the *fragments* directory from Figure 5.

Bond Fission (P2): The Bond Fission *pathway directory* will contain all the m062x, E, E1 and rotor files for each product (these filenames must end with '_m062x.log'/'_E.log'/'_E1.log'/'_D#_rotor.log'). There can be multiple Bond Fission *pathway directories*, named P1, P2, P3 and so on. If files are only given for 1 fragment, the code will assume it is a H-fission.

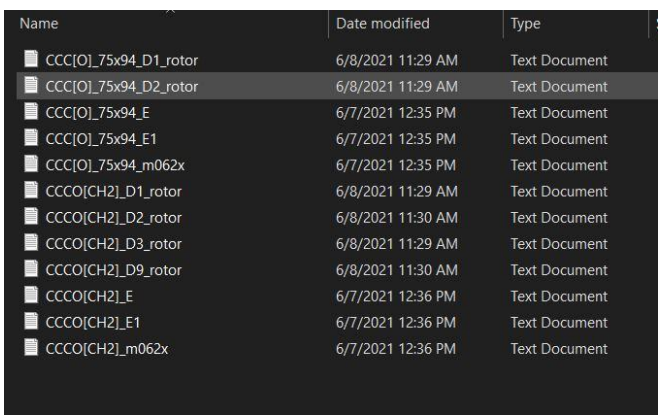

Figure 7: Files inside the P2 directory from Figure 1.

Step 2: Navigate to this directory in your terminal (make sure you have python3 installed) and run the *writeMessFile.py* python script. If you just want to generate a MESS input and followed *Step 1*, input *'n'* when prompted.

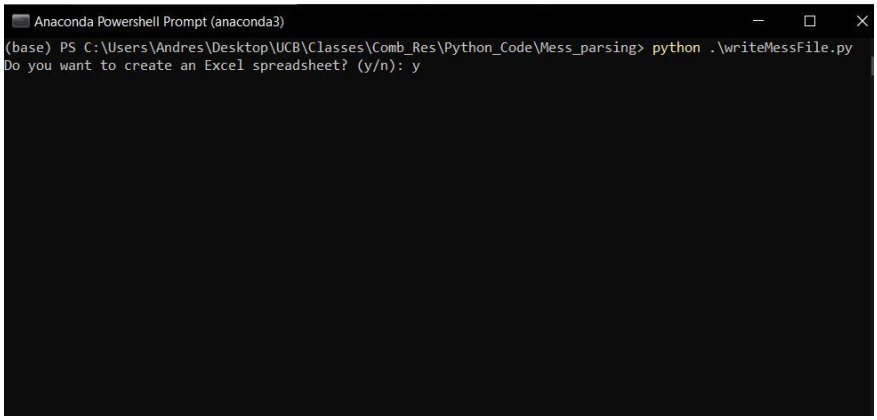

Figure 8: Calling script from the terminal.

Step 3: Output

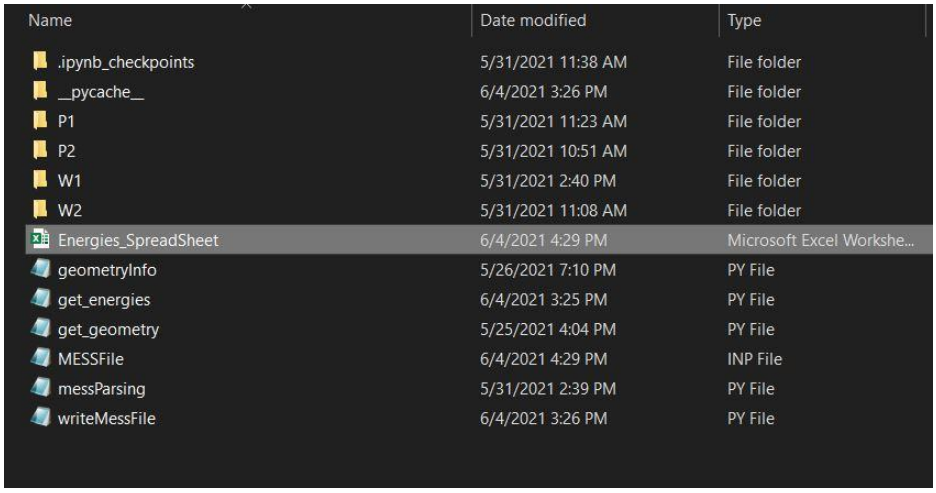

Figure 9: MESS input is generated on the root directory.

This script can be run with as many TS, BF and Wells as desired. The Mess Input file and the energies spreadsheet will be generated after calling the python scripts (*MESSFile.inp and Energies_SpreadSheet.xlsx*).