# ME 5329 Assignment 0

### Introduction to HPCC and Computer Setup

### September X, 2025

### Dr. Aquino Fall 2025

## Objectives:

- Set up our laptops to be successful in this class for the semester.
- Log on to the HPCC for the first time.
- Learn basic Linux commands and how to use the terminal.
- Run a test ORCA calculation using SLURM.
- Install and create a conda environment.

## Computer Setup:

1. Create an HPCC account by going to the website below and requesting it. Use 0000-0000-0000-0000 as the ORCID number.
   - HPCC Account Registration
2. Follow the instructions on the websites linked below based on if you have a Mac/Linux or Windows laptop on how to log on to the HPCC and what our suggested programs are for each operating system. If you have a Windows computer, you will need to install a terminal such as MobaXterm or PUTTY. For all operating systems, you will need to install molecular visualization programs such as VESTA, JMOL and Avogadro. You should also install a text editor such as VS Code or sublime and you should install either Anaconda or Miniconda to be able to run Python easily and have specific coding environments. We will use each of these programs mentioned this semester.
   - Mac/Linux Instructions
   - Windows Instructions
3. Install Conda on the HPCC by following the instructions on the website linked below.
   - Install Conda on HPCC

4. Create a *"dft"* conda environment by running `conda create -n dft python=3.12` in the Anaconda Prompt (on Windows) or your terminal (on Mac/Linux). Then, activate the environment using `conda activate dft`.

    - You may want to have a conda installation locally on your laptop, in order to postprocess the results easily. I recommend Miniconda/Miniforge3 which is a lightweight command line version of Anaconda and only installs what you need. When you load a jupyter notebook in VSCode, it will automatically find this local python version. Ask Jeremy if you need help installing a local version of python.

5. Read the article below on basic LINUX/UNIX commands and how to interact with the LINUX/UNIX terminal. This is the only way to interact with the HPCC or Super Computer.

    - Introduction to the Terminal Article

6. Read the article below which introduces the *.bashrc* file. This is an important file in our home directory that sets up our linux environment and gives us access to the computational programs such as ORCA and Quantum Espresso that we will use during this semester.

    - The *.bashrc* File

7. Read the tutorial article on how to use ORCA linked below.

    - ORCA Tutorial

8. Download `molecule_lib` from the instructions given by Jeremy in class. We will use this Python library this semester. Below is the link to the documentation of `molecule_lib`.

    - `molecule_lib` Documentation

9. Listed below are some other Python libraries to install in the *"dft"* conda environment that we will use this semester. You can install these libraries by first being in the active *"dft"* conda environment and then running the command `conda install pandas` for example.

    - `pandas`
    - `matplotlib`
    - `numpy`
    - `RDKit`

# First ORCA Calculation:

1. Generate the initial structure for Butane ($C_4H_{10}$). There are multiple ways to do this described below.

   - Using Avogadro, you can physically draw Butane by clicking left click and dragging on each Carbon atom in the visualization window.
   - Using the internet, you can search the web for a .xyz file of Butane. PubChem is a good resource to find SMILE strings and conformers of molecules. You can also use RDKit to read a .sdf file and save it as a .xyz file.
   - Using RDKit, you can use the code shown below to generate Butane using the SMILE string for it (*"CCCC"*). You can easily find SMILE strings for molecules by looking at Wikipedia or PubChem. SMILE strings do not explicitly include Hydrogens so make sure you use the `Chem.AddHs()` on the RDkit mol object to add Hydrogens where they belong. The `AllChem.EmbedMolecule()` function turns the rdkit mol object into a 3D molecule which is needed because the `Chem.MolFromSmiles()` creates a 2D RDKit mol object. The `AllChem.MMFFOptimizeMolecule()` does a rough force field optimization on the 3D RDkit mol object. There is also a SMILE string function you can use in Avogadro.

```
from rdkit import Chem
from rdkit.Chem import AllChem
import os

smile_string = "CCCC"
filename = "butane.xyz"

rdkit_mol = Chem.AddHs(Chem.MolFromSmiles(smile_string))
AllChem.EmbedMolecule(rdkit_mol)
AllChem.MMFFOptimizeMolecule(rdkit_mol)
filepath = os.path.join(os.getcwd(), filename)
Chem.MolToXYZFile(rdkit_mol, filepath)
```

2. On the HPCC, create the orca.inp file with the geometry we previously generated. The multiplicity of this system is 1 and the charge of the system is 0. Do an ORCA geometry optimization using KS-DFT functional PBE and basis set SVP.

   - The ORCA input line should look like this: *"! PBE SVP OPT"*
   - The order of items after the *"!"* in the input line does not matter in ORCA.

3. Obtain the SLURM file from instructions in class or from the ORCA Tutorial.

4. Run the calculation using the `sbatch` command and retreive the results.

    - Use the Bash command `squeue --me` to see the status of your calculation in the queue. If the time is **00:00:00**, then the calculation is still waiting to be ran. If the time is ticking up, then your calculation is actively running. If the job you submitted did not show up, then that means it has finished running.

5. Congratulations! You have successfully submitted a DFT calculation.

# Grading Rubric

No report is necessary to submit. This assignment is just to ensure everything we need for the semester is being set up and ORCA works for you.