

chem279_hw04

https://github.com/jaycee-pang/chem279_hw04

Object files:

- ``molecule.o``: Contains implementations of the ``Atom`` struct and ``Molecule`` class with functions to read in data from .txt files located in `sample_input/` and various printing functions, this is where the main EHT calculations occur, build basis functions from STO-3G basis sets, call functions of AO objects (basis functions to represent an atomic orbital). Instantiate atoms, molecules, and build basis functions based on atoms in the molecule. Every molecule has a struct of Atoms that store the atomic number, coords, and element names. From the atoms listed, the molecule will build the basis set based on the atom types (hard-coded for shell-types eg. "H1s", "C2s", "C2p", etc.) as a vector of AO objects that store angular momentum quantum numbers, the contraction coefficients, alpha exponents, and atomic centers. Here is where the overlap matrix function will be called to build S.

- ``AO.o``: functions to calculate integrals, get normalization constants, and use contraction coefficients to get overlap integrals analytically. 1D, 3D, and also overlap matrix

From HW03, I changed around my classes to make it easier for a molecule to calculate overlap integrals by making it store only the AO object (instead of cGTOs made up of primitives) and (following Zhe's solutions), to immediately calculate the normalization constants to absorb into the contraction coefficients (1 less constant to remember in later calculations). I also added a few functions for getting certain types of parameters of the molecule such as searching an atom by index or an AO by shell type or a particular atom's set of basis functions. To improve this, I would make the AO class store an atom or some way to map specific atoms to their basis functions (maybe add basis functions as an attribute of the Atom struct). I actually spent a lot of time trying to achieve this, but I got errors upon errors, maybe something to do with const correctness, copy constructors, move operators, or other issues (there were a lot of error messages in all of the ways I tried to connect the Atom struct and the AO basis functions belonging to an atom). For example, if I want to get a specific atom's basis functions, I want to only get the basis functions associated with that atom, not another atom (in the case of a diatomic or other molecules with multiple of the same atom type). The way I did it (constraint the number of basis functions returned based on the known number of basis functions for an atom) is okay, but not that elegant, I think either the atom or molecule class should keep an atom index.

- ``cndo.o``: building the Fock matrices, core Hamiltonian, solving 2-electron integral in gamma, and running the SCF routine to converge the energy (circular because we need to know P to know F). The CNDO class takes in a Molecule object, number of iterations to run the algorithm, and the tolerance criteria. After the CNDO converges according to the tolerance and maximum iterations criteria, the total energy is calculated.

Although this might be my biggest failure in a homework assignment yet, this was the most exciting. I had problems computing the gamma matrices. There are a lot of equations, and I spent a lot of time planning and writing down on paper my plans, but I still could not achieve the proper results.

Generate a library of object files ``mylib.a``

To generate a library of object files and the executables type `make all`.

Executables:

- `main`: takes in command-line argument to build a molecule, its basis functions, and runs SCF for solving the CNDO/2 SCF energy. To run, type `./main H2` or any molecule name such as `./main HF` or `./main HO`.
- `explore`: exploration of a diatomic molecule, N2. To run, type `./explore`.

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6727215/>