An Iterative Minimization Algorithm to Solve Schrödinger Equation

Abby Pavlicek, *University of Chicago*

THEORY, BACKGROUND, MOTIVATION

It is well-known that the Schrödinger equation **can only be solved exactly for a one-particle system**. In practice, **this isn't very useful**. This project explores a widely-used algorithmic approach to modeling many-electron systems governed by the time-independent Schrödinger equation:

$$E = \langle \psi | H | \psi \rangle$$

where E represents the energy eigenvalue associated with the particle's wavefunction Ψ and H is the Hamiltonian operator. Hartree-Fock theory seeks to 'solve' this equation by **variationally optimizing an antisymmetrized Slater-Determinant wavefunction** by **solving for molecular orbital (MO) coefficients** self-consistently. The Roothan equation,

$$\sum_{\nu} F_{\nu\mu} C_{\nu i} = \epsilon_i \sum_{\nu} S_{\nu\mu} C_{\nu i}$$

Is used to achieve this task computationally. F is the Fock operator: $F = H + J - \frac{1}{2} K$, a sum of the kinetic, potential, exchange, and overlap integrals. C is an orbital coefficient matrix and orbital energy eigenvalues are given by ϵ_i. The indexes **sum over the AO basis functions** with a chosen basis set. Given occupied orbitals from C, we construct the density matrix P, which describes the electron density contained in each MO. Then, the **electronic energy is given by**

$$E_{elec}^{RHF} = 0.5 (F_{\nu\mu} + H_{\nu\mu}) P_{\nu\mu}$$

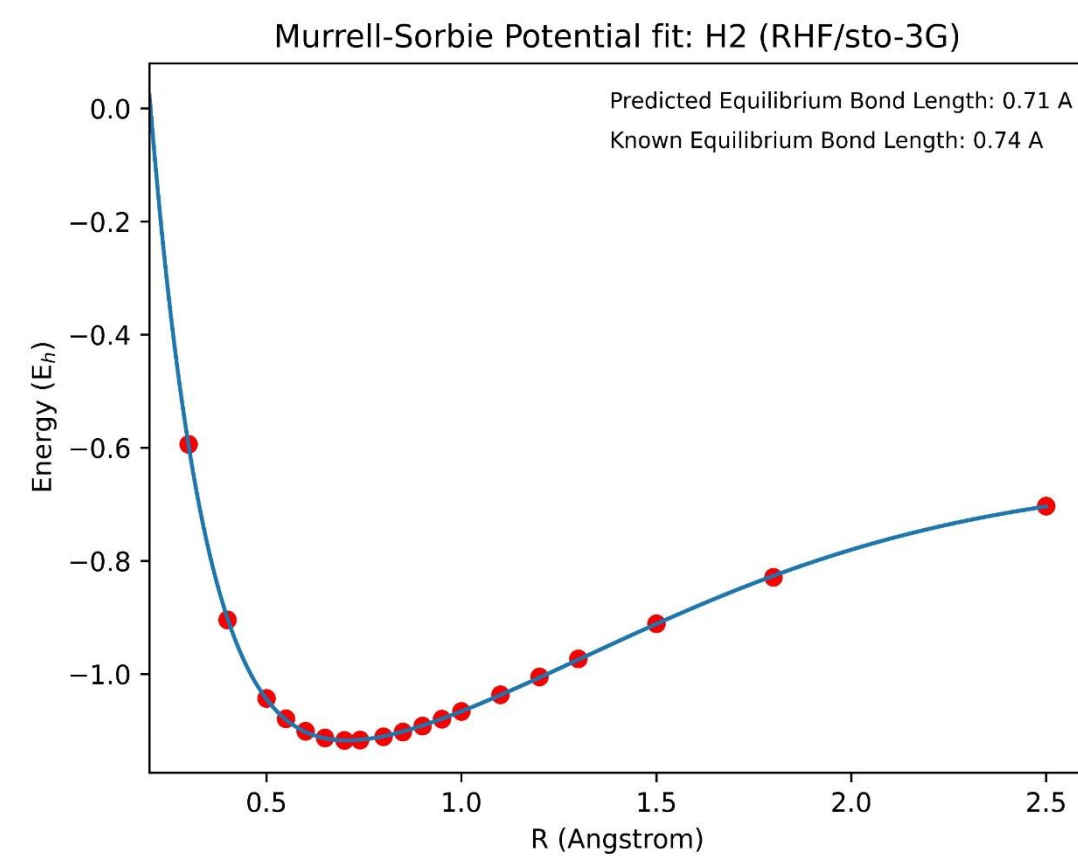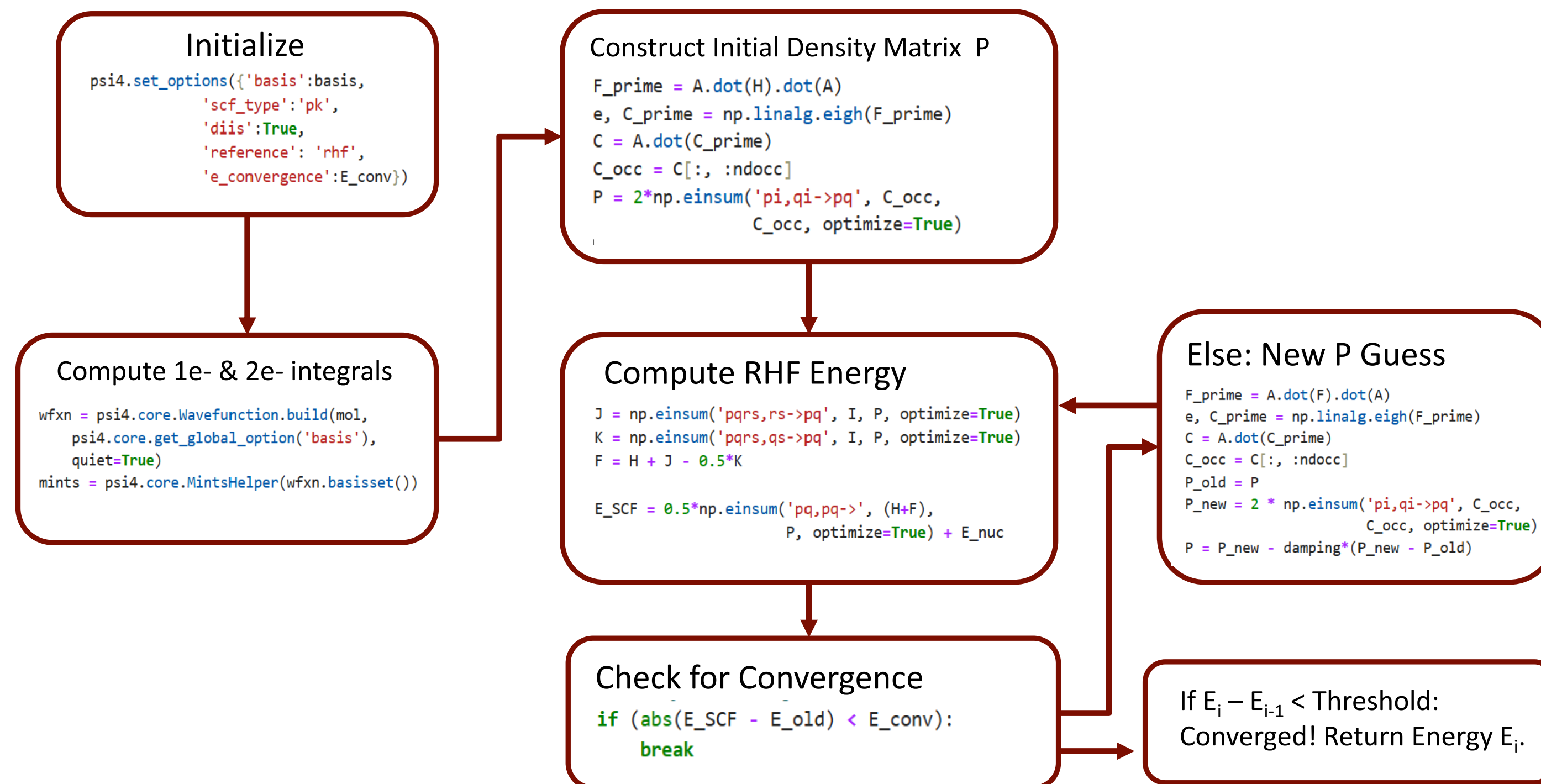Then, **the total energy of the system is given by**

$$E^{RHF} = E_{elec}^{RHF} + E_{nuc}^{RHF}$$

COMPUTATIONAL DETAILS

The code for this project is **written in Python via Jupyter Notebook**. The primary libraries used for the RHF function are **NumPy and Psi4**; NumPy allows for manipulation of eigenvalue equations, while Psi4 is a quantum chemistry toolkit which allows the user to build wavefunctions, compute common integrals, etc. in the Jupyter Notebook interface. All plots are created using the **MatPlotLib.PyPlot** library. 'Basis sets' are a finite set of orthogonal functions that approximate AO probability distributions. This project uses 'Slater-type' orbitals (STO), which are characterized by $n, m,$ and l quantum numbers, and a linear combination of x Gaussian functions (e.g., **STO-xG basis**). Molecular geometries obtained from experimental CCCBDB database.

THE SELF-CONSISTENT FIELD (SCF) PROCEDURE

The flow chart below outlines the algorithmic approach to an RHF calculation, with some sample code (the full script is available on GitHub at github.com/pavliceka/phys250_finalproject).
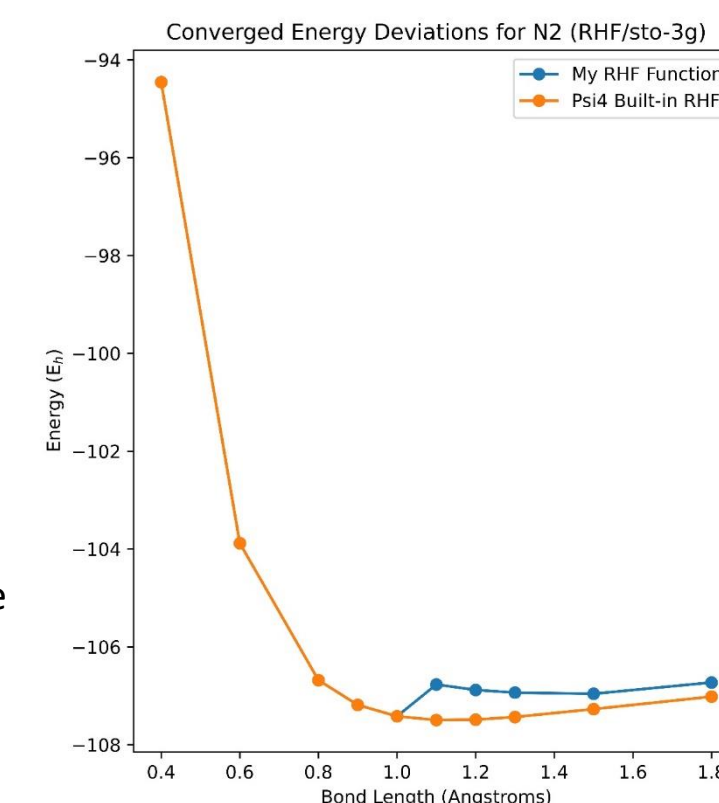
Initialize
```
psi4.set_options({'basis':basis,
        'scf_type':'pk',
        'diis':True,
        'reference': 'rhf',
        'e_convergence':E_conv})
```

Compute 1e- & 2e- integrals
```
wfxn = psi4.core.Wavefunction.build(mol,
    psi4.core.get_global_option('basis'),
    quiet=True)
mints = psi4.core.MintsHelper(wfxn.basisset())
```

Construct Initial Density Matrix P
```
F_prime = A.dot(H).dot(A)
e, C_prime = np.linalg.eigh(F_prime)
C = A.dot(C_prime)
C_occ = C[:, :ndocc]
P = 2*np.einsum('pi,qi->pq', C_occ,
                C_occ, optimize=True)
```

Compute RHF Energy
```
J = np.einsum('pqrs,rs->pq', I, P, optimize=True)
K = np.einsum('pqrs,qs->pq', I, P, optimize=True)
F = H + J - 0.5*K

E_SCF = 0.5*np.einsum('pq,pq->', (H+F),
                P, optimize=True) + E_nuc
```

Else: New P Guess
```
F_prime = A.dot(F).dot(A)
e, C_prime = np.linalg.eigh(F_prime)
C = A.dot(C_prime)
C_occ = C[:, :ndocc]
P_old = P
P_new = 2 * np.einsum('pi,qi->pq', C_occ,
                C_occ, optimize=True)
P = P_new - damping*(P_new - P_old)
```

Check for Convergence
```
if (abs(E_SCF - E_old) < E_conv):
    break
```

If $E_i − E_{i-1}$ < Threshold: Converged! Return Energy E_i.

Murrell-Sorbie Potential fit: H2 (RHF/sto-3G)
Predicted Equilibrium Bond Length: 0.71 A
Known Equilibrium Bond Length: 0.74 A

RESULTS & DISCUSSION

Potential energy curve for the bond dissociation of H_2

- Using my function, computed ground state energy eigenvalues for different bond lengths of the hydrogen diatomic
- Fit to a known potential energy function, the Murrell-Sorbie potential, to **determine predicted equilibrium bond length**
- Psi4 and my function are in very close agreement, and equilibrium bond length is **close to known value**!

Some Non-Trivial Systems

Molecule	Calculated E_h	No. of Iterations	Difference with Psi4
H2O	-74.96	8	1e-7
CH4	-39.73	7	6e-9
CH3COOH	-224.8	26*	9e-7
HCN	-91.68	25*	6e-7

*Damping Factor: 0.5

Challenges in Implementation

- Installing Psi4: conda-forge
- Checking for disagreement with built-in Psi4 energy calculation
- Methods to improve convergence, avoid oscillation; damping
- Approximation breaks down; some need higher-level energy corrections: N_2, O_2

Converged Energy Deviations for N2 (RHF/sto-3g)
My RHF Function
Psi4 Built-in RHF

SCF IN MODERN APPLICATIONS

Many novel computational algorithms for quantum chemical systems **use a form of SCF as a starting guess** for additional treatments of correlation and calculation of other electronic properties of the system. These include:

- Configuration Interaction (CI) Methods
- Multiconfigurational Methods, with MC-SCF
- Coupled-Cluster (CC) Methods:

CuH EE-EOM-CCSD(T) (aug-cc-pVDZ)
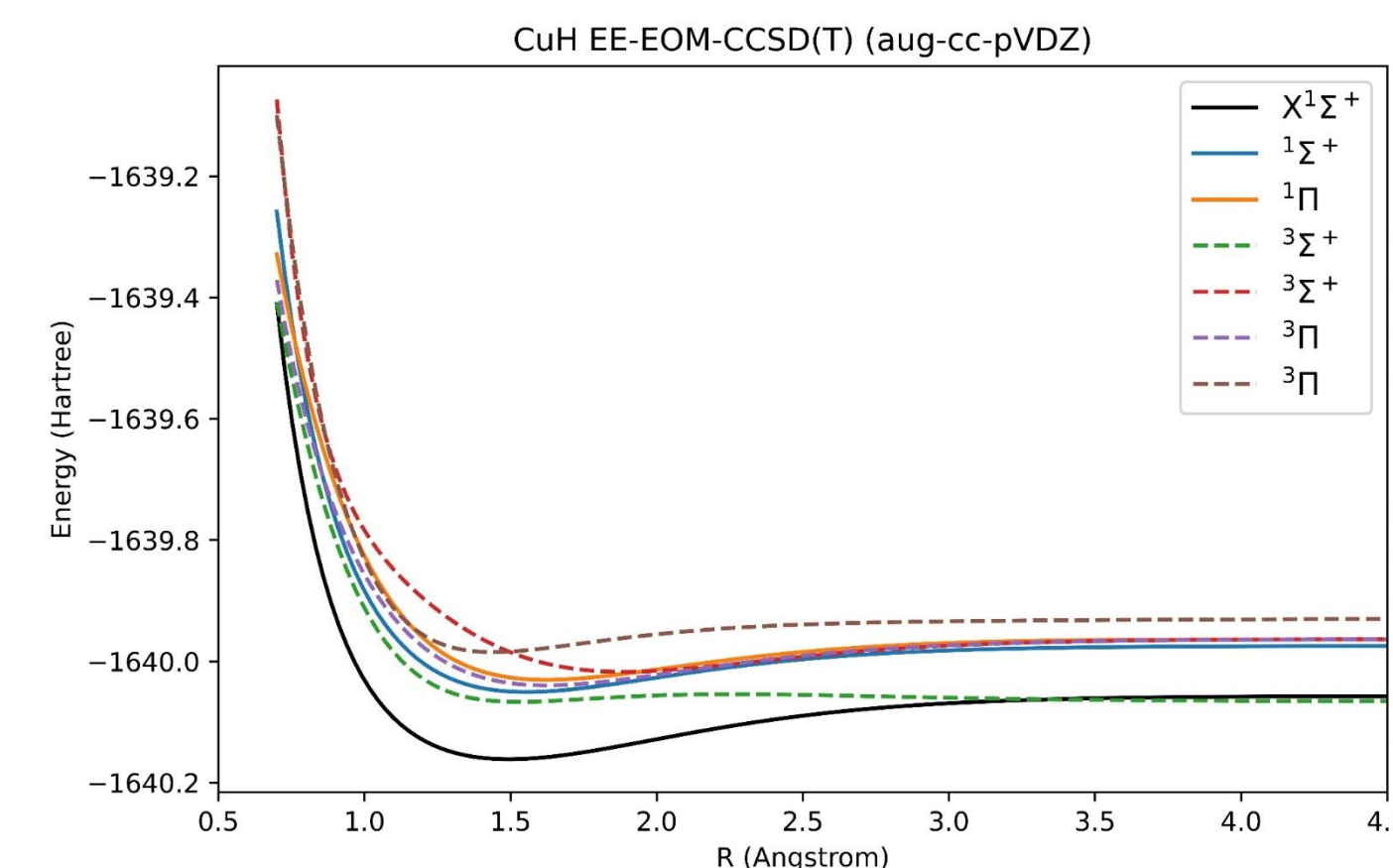
- Hybrid Classical-Quantum Algorithms that utilize quantum computers:

qUCCSD
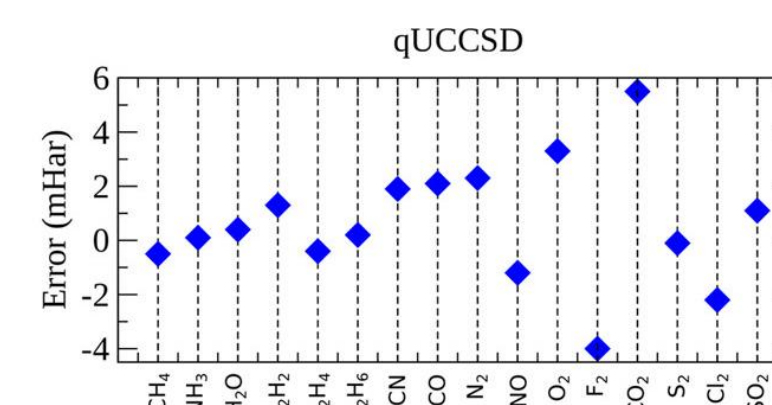
Results from an implementation of a hybrid quantum classical algorithm, using unitary coupled-cluster theory. Current research focuses on 'low-depth' circuits, which reduce demands on the physical quantum systems being measured in a quantum computer.

REFERENCES

1. Szabo. A., and N. S. Ostlund, *Modern Quantum Chemistry*, Introduction to Advanced Electronic Structure Theory. Courier Corporation, 1996.
2. "Psi4 1.4: Open-Source Software for High-Throughput Quantum Chemistry", *J. Chem. Phys.* (2020). doi: 10.1063/5.0006002
3. Chen, J., Cheng, H., and J. K. Freericks. "Low-Depth Unitary Coupled Cluster Theory for Quantum Computation", *J. Chem. Theory Comput.* (2022). doi: 10.1021/acs.jctc.1c01026

ACKNOWLEDGEMENTS