Handout 5: Implementing the Hartree–Fock Method

While clear and detailed expositions to molecular integrals are hard to find in the literature (hopefully until now) there are many excellent resources that go into some detail about the Hartree–Fock–Roothan–Hall (HF) method. (Perhaps the most complete and excellent description of HF theory at a level of rigor consistent with the present paper is offered in the classic textbook by Szabo and Ostlund.)

Equipped with the values computed by the methods presented in Handout 4, one is ready to finish computing the solution to the Schrödinger equation. This is done by the *self-consistent field* (a.k.a., Hartree–Fock) method – it is a looping "recipe" whereby molecular orbitals are constructed from the atomic orbitals we have been working with this entire paper. The looping is necessary because the first guess at the nature of the molecular orbitals (which must be made to begin iteration) tends to be crude, but over a series of iterations these orbitals are tweaked, step-by-step, until there is little change in the orbitals between steps. The HF procedure is much easier to code than than the molecular integrals it depends on.

Implementing the HF method involves just a few additional NumPy techniques, namely computing the inverse of a matrix, taking dot products, and computing eigenvalues and eigenvectors of matrices. These will be introduced in individual steps of the algorithm we now present.

The HF algorithm is as follows:

1. Specify a molecule (as an `.xyz` file containing atom coordinates and atomic number) and a basis set (in the present study, STO-3G). This step was done when computing molecular integrals, as described in Handout 4.

2. Calculate all required molecular integrals: **S**, **T**, **V**, and **G**. (Completed in Handout 4.)

3. Sum the kinetic and electron-nuclear attraction integral matrices to form the core Hamiltonian:

$$\mathbf{H}^{\text{core}} = \mathbf{T} + \mathbf{V}\,. \tag{1}$$

   With NumPy, matrix addition is as simple as the following line:

```
Hcore = T + V
```

4. Generate the orthonormal transformation matrix **X** from **S** with the following equation:

$$\mathbf{X} = \mathbf{S}^{-1/2}\,. \tag{2}$$

```
X = linalg.sqrtm(linalg.inv(S))
```

5. Guess the initial density matrix **D**. The simplest guess is the null (zero) matrix. Hence, let $\mathbf{D} = \mathbf{0}$, where **D** is $K \times K$ in dimension, and K is the number of atomic orbitals in the basis.

6. Calculate the two-electron contribution **P** to the Fock matrix with

$$P_{\mu\nu} = \sum_{\lambda} \sum_{\sigma} D_{\lambda\sigma} \left(G_{\mu\nu\sigma\lambda} - \frac{1}{2} G_{\mu\lambda\sigma\nu} \right) \tag{3}$$

```
# K is the basis size (e.g., K = 7 for H2O in the STO-3G basis)
for m in range(K):
  for n in range(K):
    P[m,n] = 0.0
    for l in range(K):
      for s in range(K):
        P[m,n] += D[l,s] * (G[m,n,s,l] - 0.5*G[m,l,s,n])
```

7. Add **P** to the core Hamiltonian to obtain the Fock matrix:

$$\mathbf{F} = \mathbf{H}^{\text{core}} + \mathbf{P}\,. \tag{4}$$

8. Transform the Fock matrix to the orthonormal basis with dot products involving the original Fock matrix **F**, and the transformation matrix **X**:

$$\mathbf{F}' = \mathbf{XFX}\,. \tag{5}$$

It is easy to calculate dot products of matrices (with Python 3.5 or above) with the "at" symbol:

```
Fp = X @ F @ X
```

9. Diagonalize **F’** to obtain its eigenvalues ϵ and eigenvectors **C**’.

```
e, Cp = linalg.eigh(Fp)
```

10. Calculate the molecular orbitals **C**:

$$\mathbf{C} = \mathbf{XC}'\,. \tag{6}$$

11. Form a new and improved density matrix **D** from **C** using

$$D_{\mu\nu} = 2\sum_{a}^{N/2} C_{\mu a} C^{*}_{\nu a}\,. \tag{7}$$

```
# K is the basis size (e.g., K = 7 for H2O in the STO-3G basis)
# N is the number of electrons (e.g., N = 10 for neutral water)
for m in range(K):
  for n in range(K):
    D[m,n] = 0.0
    for a in range(int(N/2)):
      D[m,n] += 2 * (C[m,a] * C[n,a])
```

12. Determine whether self-consistency has been achieved (i.e., whether the Hartree–Fock electronic energy has converged). Do so by determining whether the Hartree–Fock energy expectation value E_e

$$E_e = \frac{1}{2}\sum_{\mu}\sum_{\nu} D_{\nu\mu}(H^{\text{core}}_{\mu\nu} + F_{\mu\nu}) \tag{8}$$

is the same as the energy expectation value of the previous iteration within a specified criterion (e.g., $10^{-7}E_h$). If the procedure has not converged, return to step 6, passing it the newest density matrix **D** from step 11.

13. If the procedure has converged, append the nuclear-nuclear repulsion to the converged electronic energy E_e (from Step 12).

$$E_{\text{tot}} = E_e + \sum_{A} \sum_{B>A} \frac{Z_A Z_B}{R_{AB}} \tag{9}$$

You have solved Schrodinger's equation! Specifically, you've found the total energy of the molecule (an expectation value), the eigenvector set of molecular orbitals, and the molecular orbital energies.