
Numerical Algorithms Applied to Computational Quantum Chemistry
Homework 4: Build your own SCF program

Prof. Martin Head-Gordon and Steve Shen

Due Nov. 1, 2022

1 BUILD THE CNDO/2 FOCK MATRIX

In class, we have discussed semi-empirical Hartree-Fock methods in some detail. They have advantages over ab initio Hartree-Fock in speed and in simplicity of implementation. With the latter in mind, in this homework, you will implement your own version of the CNDO/2 method, which is a huge leap forwards from the extended Huckel method.

In this problem, you will build code to assemble the matrix elements of the CNDO/2 Fock operator, \mathbf{f} (in the AO basis) given the AO basis total density matrix, \mathbf{p} , where, for an open shell molecule with p α electrons and q β electrons, we define:

$$p_{\mu\nu}^{\alpha} = \sum_i^p c_{\mu i}^{\alpha} c_{\nu i}^{\alpha} \quad (1.1)$$

$$p_{\mu\nu}^{\beta} = \sum_i^q c_{\mu i}^{\beta} c_{\nu i}^{\beta} \quad (1.2)$$

$$p_{\mu\nu}^{\text{tot}} = p_{\mu\nu}^{\alpha} + p_{\mu\nu}^{\beta} \quad (1.3)$$

The diagonal Fock matrix elements are given by (here we are going to do the unrestricted equations):

$$f_{\mu\mu}^{\alpha} = -\frac{1}{2} (I_{\mu} + A_{\mu}) + \left[(p_{AA}^{\text{tot}} - Z_A) - \left(p_{\mu\mu}^{\alpha} - \frac{1}{2} \right) \right] \gamma_{AA} + \sum_{B \neq A} (p_{BB}^{\text{tot}} - Z_B) \gamma_{AB} \quad (1.4)$$

The off-diagonal elements are given by:

$$f_{\mu\nu}^{\alpha} = \frac{1}{2} (\beta_A + \beta_B) s_{\mu\nu} - p_{\mu\nu}^{\alpha} \gamma_{AB} \quad (1.5)$$

| quantity | H | C | N | O | F |
|--------------------------|-------|--------|--------|--------|--------|
| $\frac{1}{2}(I_s + A_s)$ | 7.176 | 14.051 | 19.316 | 25.390 | 32.272 |
| $\frac{1}{2}(I_p + A_p)$ | | 5.572 | 7.275 | 9.111 | 11.080 |
| $-\beta$ | 9 | 21 | 25 | 31 | 39 |

Table 1.1: Semi-empirical parameters that define the CNDO/2 model (units: eV)

Here, as usual μ, ν are atomic orbital indexes, and A, B are atom indexes such that atomic orbital $\omega_\mu(\mathbf{r})$ is centered on atom A at position \mathbf{R}_A , and atomic orbital $\omega_\nu(\mathbf{r})$ is centered on atom B at position \mathbf{R}_B .

The parameters I_μ and A_μ correspond to ionization energies and electron affinities and are fixed values tabulated below for the first row atoms in Table 1. The quantity p_{AA} is the total density on atom A :

$$p_{AA} = \sum_{\mu \in A} p_{\mu\mu}^{\text{tot}} \quad (1.6)$$

Z_A is the valence atomic number of atom A . $s_{\mu\nu}$ are elements of the AO basis overlap matrix, \mathbf{s} , which is familiar to you from building your extended Huckel program. β_A are atomic bonding parameters, and are also tabulated in Table 1.

The only thing not yet defined are the quantities γ_{AB} . *You will develop code to evaluate γ_{AB} in your compute lab this coming week, so this will be available to help you finish assembly of your Fock matrices.* We will provide the basic details as an appendix to this homework as a summary. After compute lab, you know that you have the capability to form them!

- Build your code to assemble $f_{\mu\nu}^\alpha$ (and of course $f_{\mu\nu}^\beta$ with exactly the same code).
- Be careful with units! The parameters given in Table 1 are in eV, so you will then need to make sure that your two-electron integrals are in the same units! Working in “atomic units”, the conversion factor is 27.211 eV/a.u.
- Test first on the simple H_2 example that Steve will provide reference data for (here we know \mathbf{p}^α and \mathbf{p}^β so Steve does not have to provide those!).
- Then test on the more challenging HO^\bullet example, with $N = 5$, $p = 4$ and $q = 3$ where Steve will provide inputs and outputs.

2 EVALUATE THE OPEN SHELL CNDO/2 SELF-CONSISTENT FIELD EQUATIONS

With your Fock matrix assembly code for \mathbf{f}^α and \mathbf{f}^β from the previous question working, you are ready to implement your own SCF code for CNDO/2. In the AO basis, in matrix form, the SCF equations are:

$$\mathbf{f}^\alpha \mathbf{c}^\alpha = \mathbf{c}^\alpha \epsilon^\alpha \quad (2.1)$$

$$\mathbf{f}^\beta \mathbf{c}^\beta = \mathbf{c}^\beta \epsilon^\beta \quad (2.2)$$

Or if you prefer them written out explicitly, they are:

$$\sum_{\nu} f_{\mu\nu}^{\alpha} c_{\nu i}^{\alpha} = c_{\mu i}^{\alpha} \epsilon_i^{\alpha} \quad (2.3)$$

$$\sum_{\nu} f_{\mu\nu}^{\beta} c_{\nu i}^{\beta} = c_{\mu i}^{\beta} \epsilon_i^{\beta} \quad (2.4)$$

Your code should execute the classic SCF algorithm:

1. Guess $\mathbf{p}^{\alpha} = \mathbf{p}^{\beta} = \mathbf{0}$
2. Build \mathbf{f}^{α} and \mathbf{f}^{β} using your code from above.
3. Solve the eigenvalue problems, Eqs. 2.3 and 2.4, to obtain new MO coefficients, \mathbf{c}^{α} and \mathbf{c}^{β} , and the corresponding eigenvalues, ϵ^{α} and ϵ^{β}
4. Copy the old density matrices to $\mathbf{p}_{\text{old}}^{\alpha}$ and $\mathbf{p}_{\text{old}}^{\beta}$
5. Assemble new density matrices by occupying the p lowest energy α MOs and the q lowest energy β MOs and using Eqs. 1.1, 1.2 and 1.3
6. If the maximum magnitude of the change in the α and β density matrices is less than a tolerance you specify (e.g. 10^{-6}), then you have converged. Otherwise return to step 2.

Make sure you extract the final prize from converging your SCF calculations, namely the total energy:

$$E_{\text{CNDO}/2} = \frac{1}{2} \sum_{\mu\nu} p_{\mu\nu}^{\alpha} (h_{\mu\nu} + f_{\mu\nu}^{\alpha}) + \frac{1}{2} \sum_{\mu\nu} p_{\mu\nu}^{\beta} (h_{\mu\nu} + f_{\mu\nu}^{\beta}) + \sum_A \sum_{B < A} \frac{Z_A Z_B}{R_{AB}} \quad (2.5)$$

The last term is the nuclear repulsion energy (be careful about its units also!). In the energy expression, $h_{\mu\nu}$ is the core hamiltonian matrix, which consists of the one-electron parts of the Fock matrices (i.e. those that are independent of the electron density). Specifically this means:

$$h_{\mu\mu} = -\frac{1}{2} (I_{\mu} + A_{\mu}) - \left(Z_A - \frac{1}{2} \right) \gamma_{AA} - \sum_{B \neq A} Z_B \gamma_{AB} \quad (2.6)$$

$$h_{\mu\nu} = \frac{1}{2} (\beta_A + \beta_B) s_{\mu\nu} \quad (2.7)$$

You can readily make code for this core hamiltonian as a stripped down version of your Fock matrix builder.

So, altogether, we want you to:

1. Write and debug the unrestricted CNDO/2 SCF algorithm above, using armadillo to handle the matrix operations. Use H_2 as your first debugging case (where there should only be 2 iterations). Then HF as a second case. Steve will provide reference data for you for these cases.

2. Explore a little bit of chemistry with your CNDO/2 code. For example, you could look at the molecular orbitals and the bond energy of a simple diatomic molecule, such as N_2 ($p = 5, q = 5$; singlet multiplicity; $R_{NN} = 1.098\text{\AA}$), getting the bond energy by also evaluating the N atom ($p = 4, q = 1$; quartet multiplicity). It would also be maybe nice to go evaluate the equilibrium bondlength, by minimizing the total energy, and compare with the experimental value of $R_{NN} = 1.098\text{\AA}$. And/or you could do an open shell diatomic such as O_2 ($p = 7, q = 5$; triplet multiplicity; $R_{OO} = 1.208\text{\AA}$) vs O atom ($p = 4, q = 2$), and optimize its bondlength to compare against the experimental value of $R_{OO} = 1.208\text{\AA}$. Or you could choose something yourself and surprise us! Your code can in principle do quite large molecules, if you just provide the structure.

3 APPENDIX: NOTES FOR COMPUTE LAB ON EVALUATING γ_{AB}

γ_{AB} are two-center 2-electron repulsion integrals evaluated over the square of the valence s orbital centered on atoms A and B:

$$\gamma_{AB} = \int_{\mathbf{r}_1} \int_{\mathbf{r}_2} s_A^2(\mathbf{r}_1) \frac{1}{r_{12}} s_B^2(\mathbf{r}_2) \quad (3.1)$$

While 6-dimensional integrals look intimidating, this one can be analytically reduced to a 1-dimensional integral. Remembering that

$$s_A(r) \equiv \sum_k^3 (d_{ks_A} N_k^s) \omega_k^s(\mathbf{r} - \mathbf{R}_A) = \sum_k^3 d'_{ks_A} \omega_k^s(\mathbf{r} - \mathbf{R}_A) \quad (3.2)$$

The integral then becomes a sum over two-electron integrals over primitive Gaussians:

$$\gamma_{AB} = \sum_k^3 \sum_{k'}^3 \sum_l^3 \sum_{l'}^3 d'_{ks_A} d'_{k's_A} d'_{ls_B} d'_{l's_B} [0]^{(0)} \quad (3.3)$$

The 6-dimensional integral is hidden in $[0]^{(0)}$ (this is conventional notation for the integral over the product of 4 primitive Gaussian s functions):

$$[0]^{(0)} = \int_{\mathbf{r}_1} \int_{\mathbf{r}_2} \omega_k^s(\mathbf{r}_1 - \mathbf{R}_A) \omega_{k'}^s(\mathbf{r}_1 - \mathbf{R}_A) \frac{1}{r_{12}} \omega_l^s(\mathbf{r}_2 - \mathbf{R}_B) \omega_{l'}^s(\mathbf{r}_2 - \mathbf{R}_B) \quad (3.4)$$

$$\equiv \int_{\mathbf{r}_1} \int_{\mathbf{r}_2} \exp[-(\alpha_k + \alpha_{k'}) (\mathbf{r}_1 - \mathbf{R}_A)^2] \frac{1}{r_{12}} \exp[-(\beta_l + \beta_{l'}) (\mathbf{r}_2 - \mathbf{R}_B)^2] \quad (3.5)$$

It can be shown that (for original proof, see article by S.F. Boys, Proc. R. Soc. Lond. A200,542 (1950)):

$$[0]^{(0)} = U_A U_B \sqrt{2V^2} \sqrt{\frac{2}{\pi}} \int_0^1 \exp(-Tu^2) du \quad (3.6)$$

The quantities T, U, V are given by:

$$T = V^2 (\mathbf{R}_A - \mathbf{R}_B)^2 \quad (3.7)$$

$$U = U_A U_B \quad (3.8)$$

$$V^2 = (\sigma_A + \sigma_B)^{-1} \quad (3.9)$$

And the remaining quantities U_A and σ_A are:

$$\sigma_A = (\alpha_k + \alpha_{k'})^{-1} \quad (3.10)$$

$$U_A = (\pi\sigma_A)^{3/2} \quad (3.11)$$

with analogous expressions for U_B and σ_B .

The 1-dimensional integral that remains is a specific case of the function known as the “Boys function”, and for the special case we are interested in (s functions), $F_0(T)$ is given by:

$$F_0(T) = \int_0^1 \exp(-Tu^2) du = \frac{1}{2} \sqrt{\frac{\pi}{T}} \operatorname{erf}(\sqrt{T}) \quad (3.12)$$

Efficient evaluation of the error function is discussed in Numerical Recipes, in Sec. 6.2.2., and erf is defined as:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-v^2) dv \quad (3.13)$$

The Boys function can also be related to other standard functions, such as the incomplete gamma function, discussed in Sec. 6.2 of Numerical Recipes.

A useful computable form of the fundamental 2-center integral over 4 primitive s-type Gaussians is given by rewriting Eqs. 3.6 in terms of erf(\sqrt{T}):

$$[0]^{(0)} = U_A U_B \sqrt{\frac{1}{(\mathbf{R}_A - \mathbf{R}_B)^2}} \operatorname{erf}(\sqrt{T}) \quad (3.14)$$

This result, Eq. 3.14, is then used in Eq. 3.3 to evaluate the not-that-tricky-after-all γ_{AB} !

Well, actually there is one tricky thing: note that if $\mathbf{R}_A = \mathbf{R}_B$ (i.e. a 1-center 2-electron integral like γ_{AA}) then the denominator of Eq. 3.14 goes to zero, and since erf(0) = 0, your code will blow up! The correct result can be evaluated from Eq. 3.6:

$$[0]^{(0)}(T=0) = U_A U_B \sqrt{2V^2} \sqrt{\frac{2}{\pi}} \quad (3.15)$$

4 APPENDIX: VALENCE STO-3G PARAMETERS FOR H, C, N, O, F

Given in Figure 4.1 are the exponents and contraction coefficients for the first row elements associated with most organic molecules (you should already have C and H from your adventures with extended Huckel theory). More are available in the paper posted on the class web-site, and, especially at the basis set exchange web site: <https://www.basissetexchange.org>

| Atom | exponent | s contraction | p contraction |
|------|------------|---------------|---------------|
| H | 3.42525091 | 0.15432897 | |
| | 0.62391373 | 0.53532814 | |
| | 0.16885540 | 0.44463454 | |
| C | 2.94124940 | -0.09996723 | 0.15591627 |
| | 0.68348310 | 0.39951283 | 0.60768372 |
| | 0.22228990 | 0.70011547 | 0.39195739 |
| N | 3.78045590 | -0.09996723 | 0.15591627 |
| | 0.87849660 | 0.39951283 | 0.60768372 |
| | 0.28571440 | 0.70011547 | 0.39195739 |
| O | 5.03315130 | -0.09996723 | 0.15591627 |
| | 1.16959610 | 0.39951283 | 0.60768372 |
| | 0.38038900 | 0.70011547 | 0.39195739 |
| F | 6.46480320 | -0.09996723 | 0.15591627 |
| | 1.50228120 | 0.39951283 | 0.60768372 |
| | 0.48858850 | 0.70011547 | 0.39195739 |

Figure 4.1: Exponents and contraction coefficients for STO-3G for the standard light elements (in atomic units).