Project 1, PHY981 Spring 2016

February 9, 2016

Project 1, Hartree-Fock Calculations of Neutron drops, Deadline March 4

Introduction

Neutron drops are a powerful theoretical laboratory for testing, validating and improving nuclear structure models. Indeed, all approaches to nuclear structure, from ab initio theory to shell model to density functional theory are applicable in such systems. We will, therefore, use neutron drops as a test system for setting up a Hartree-Fock code. This program can later be extended to studies of the binding energy of nuclei like ^{16}O or ^{40}Ca. The single-particle energies obtained by solving the Hartree-Fock equations can then be directly related to experimental separation energies. For those of you interested in such studies, the program you will end up developing here can be used in later projects, with simple extensions. Since Hartree-Fock theory is the starting point for several many-body techniques (density functional theory, random-phase approximation, shell-model etc), the aim here is to develop a computer program to solve the HF equations in a given single-particle basis, here the harmonic oscillator.

You are encouraged to collaborate and working groups of 2-3 people are often optimal. You can hand in a common report.

The Microscopic Neutron Drop Hamiltonian

The Hamiltonian for a system of N neutron drops confined in a harmonic potential reads

$$\hat{H} = \sum_{i=1}^{N} \frac{\hat{p}_i^2}{2m} + \sum_{i=1}^{N} \frac{1}{2} m\omega r_i^2 + \sum_{i<j} \hat{V}_{ij}, \tag{1}$$

with $\hbar^2/2m = 20.73$ fm^2, $mc^2 = 938.90590$ MeV, and $\hat{V}_{ij}$ is the two-body interaction potential whose matrix elements are precalculated and to be read in by you.

The Hartree-Fock algorithm can be broken down as follows. We recall that our Hartree-Fock matrix is

$$\hat{h}_{\alpha\beta}^{HF} = \langle\alpha|\hat{h}_0|\beta\rangle + \sum_{j=1}^{N}\sum_{\gamma\delta} C_{j\gamma}^{*}C_{j\delta}\langle\alpha\gamma|V|\beta\delta\rangle_{AS}.$$

Normally we assume that the single-particle basis $|\beta\rangle$ forms an eigenbasis for the operator $\hat{h}_0$ (this is our case), meaning that the Hartree-Fock matrix becomes

$$\hat{h}_{\alpha\beta}^{HF} = \epsilon_\alpha\delta_{\alpha,\beta} + \sum_{j=1}^{N}\sum_{\gamma\delta} C_{j\gamma}^{*}C_{j\delta}\langle\alpha\gamma|V|\beta\delta\rangle_{AS}.$$

The Hartree-Fock eigenvalue problem

$$\sum_{\beta}\hat{h}_{\alpha\beta}^{HF}C_{i\beta} = \epsilon_i^{\mathrm{HF}}C_{i\alpha},$$

can be written out in a more compact form as

$$\hat{h}^{HF}\hat{C} = \epsilon^{\mathrm{HF}}\hat{C}.$$

The equations are often rewritten in terms of a so-called density matrix, which is defined as

$$\rho_{\gamma\delta} = \sum_{i=1}^{N}\langle\gamma|i\rangle\langle i|\delta\rangle = \sum_{i=1}^{N} C_{i\gamma}C_{i\delta}^{*}. \tag{2}$$

It means that we can rewrite the Hartree-Fock Hamiltonian as

$$\hat{h}_{\alpha\beta}^{HF} = \epsilon_\alpha\delta_{\alpha,\beta} + \sum_{\gamma\delta}\rho_{\gamma\delta}\langle\alpha\gamma|V|\beta\delta\rangle_{AS}.$$

It is convenient to use the density matrix since we can precalculate in every iteration the product of two eigenvector components C.

Note that $\langle\alpha|\hat{h}_0|\beta\rangle$ denotes the matrix elements of the one-body part of the starting hamiltonian. For self-bound nuclei $\langle\alpha|\hat{h}_0|\beta\rangle$ is the kinetic energy, whereas for neutron drops, $\langle\alpha|\hat{h}_0|\beta\rangle$ represents the harmonic oscillator hamiltonian since the system is confined in a harmonic trap. If we are working in a harmonic oscillator basis with the same ω as the trapping potential, then $\langle\alpha|\hat{h}_0|\beta\rangle$ is diagonal.

The Hartree-Fock equations are, in their simplest form, solved in an iterative way, starting with a guess for the coefficients $C_{i\alpha}$. We label the coefficients as $C_{i\alpha}^{(n)}$, where the superscript n stands for iteration n. To set up the algorithm we can proceed as follows.

1. We start with a guess $C_{i\alpha}^{(0)} = \delta_{i,\alpha}$. Alternatively, we could have used random starting values as long as the vectors are normalized. Another possibility is to give states below the Fermi level a larger weight. We construct then the density matrix and the Hartree-Fock Hamiltonian.

2. The Hartree-Fock matrix simplifies then to

$$\hat{h}_{\alpha\beta}^{HF}(0) = \epsilon_\alpha \delta_{\alpha,\beta} + \sum_{\gamma\delta} \rho_{\gamma\delta}^{(0)} \langle \alpha\gamma|V|\beta\delta\rangle_{AS}.$$

   Solving the Hartree-Fock eigenvalue problem yields then new eigenvectors $C_{i\alpha}^{(1)}$ and eigenvalues $\epsilon_i^{\mathrm{HF}}(1)$.

3. With the new eigenvalues we can set up a new Hartree-Fock potential

$$\sum_{\gamma\delta} \rho_{\gamma\delta}^{(1)} \langle \alpha\gamma|V|\beta\delta\rangle_{AS}.$$

   The diagonalization with the new Hartree-Fock potential yields new eigenvectors and eigenvalues. This process is continued till for example

$$\frac{\sum_p |\epsilon_i^{\mathrm{HF}}(n) - \epsilon_i^{\mathrm{HF}}(n-1)|}{m} \le \lambda,$$

   where λ is a user prefixed quantity ($\lambda \sim 10^{-8}$ or smaller) and p runs over all calculated single-particle energies and m is the number of single-particle states.

Outline of a HF Solver

When setting up the Hartree-Fock code, we normally recommend to

1. Write first a "pseudo-code" for your HF solver. Feel free to refer to the C++ listing below for guidance.

2. Start translating your pseudo-code into an actual implementation.

Code Example

An example of a function in C++ which performs the Hartree-Fock calculation is shown here. In setting up your code you will need to write a function which sets up the single-particle basis, the matrix elements $t_{\alpha\gamma}$ of the one-body operator (called *h*0 in the function below) and the antisymmetrized TBMEs (called *matrixElement* below) and the density matrix elements $\rho_{\beta\delta}$ (called *densityMatrix* below).

```
void hartreeFock::run() {
    double spPot;
    // ---------------- Setting up the HF-hamiltonian using D
        = 1 as guess, Armadillo is used for vectors
    mat h;
    vec E = zeros(nStates, 1);
    vec ePrev = zeros(nStates, 1);
    mat C = eye(nStates, nStates);
    vec diff;

    // Hartree-Fock loop
    int hfIt = 0;
    while (hfIt < HFIterations) {
        cout << "iteration = " << hfIt << endl;

        h = zeros(nStates, nStates);
        for (int alpha = 0; alpha < nStates; alpha++) {
            for (int beta = 0; beta < nStates; beta++) {
                spPot = 0;
                    for (int gamma = 0; gamma < nStates;
                       gamma++) {
                        for (int delta = 0; delta < nStates;
                            delta++) {
                            spPot += densityMatrix(gamma,
                               delta,D) * matrixElement(
                               alpha, gamma, beta, delta);
                        }
                    }

                h(alpha, beta) = h(beta, alpha) = h0(alpha,
                   beta) + spPot;
            }
        }
        //Computing the HF one-body energies
        eig_sym(E, C, h);
        // Transposing the vectors
        C = trans(C);
        hfIt++;
        // Convergence test
        diff = E - ePrev;
        if (abs(diff.max()) < threshold)
            break;
        ePrev = E;
    }
    double E0 = calcEnergy(C);
    cout << "Final energy E = " << E0 << " after " << hfIt
       << " iterations, error < " << threshold << endl;
}
```

Single-particle data and two-body matrix elements

We will perform Hartree-Fock calculations for eight, $N = 8$, neutrons in an oscillator potentials with an oscillator frequency $\hbar\omega = 10$ MeV. This means that we are filling the $0s$ and the $0p$ shells and that these single-particle states define the reference state, or our ansatz for the ground state. The total set of single-particle states will comprise four major shells only, that is the $0s$, $0p$, $1s0d$ and $1p0f$ shells.

The input file *spdata.dat* contains the information of all single-particle quantum numbers needed to define this space. In total we have 40 single-particle states labeled by n, j, l and m, where m is the projection of the total single-particle angular momentum j. To every set of single-particle quantum numbers there is a unique number p identifiying them, meaning that the two-body matrix elements in the file *twobody.dat* are identified as $\langle pq|\hat{v}|rs\rangle$.

You will need to read these two files and set up arrays which store the matrix elements while running the program.

Now let us consider the general case. Our HO single particle basis states are labeled as $|nljm\rangle$. **The HF states are denoted $|\bar{n}ljm\rangle$, where we will put aways put a bar over the principle quantum numbers to distinguish them from HO states.** Expanding the HF states in the oscillator basis gives

$$|\bar{n}ljm\rangle = \sum_{n'} |n'ljm\rangle\langle n'ljm|\bar{n}ljm\rangle \equiv \sum_{n'} |n'jlm\rangle C^{lj}_{n'\bar{n}}\,. \tag{3}$$

We have made use of the fact that only HO states with the same ljm values as the HF state contribute, and the overlap C-matrix is independent of m-value.

It is a good exercise (try it!) to show that the HF s.p. hamiltonian is diagonal in ljm (and independent of m), and that the HF equations can be written as

$$\sum_{n_3} h^{lj}_{n_1 n_3} C^{lj}_{n_3 \bar{n}} = \epsilon_{\bar{n}lj} C^{lj}_{n_3 \bar{n}} \,. \tag{4}$$

where the s.p. HF hamiltonian matrix elements are

$$h^{lj}_{n_1 n_3} = \delta_{n_3 n_1}(2n_1 + l + 3/2)\hbar\omega + \sum_{n_2 n_4} \sum_{l'j'}^{occ} \langle n_1 l j n_2 l' j' | V | n_3 l j n_4 l' j' \rangle \rho^{l'j'}_{n_4 n_2} \,. \tag{5}$$

The *occ* on the second summation is to remind you that the sum is over $l'j'$ values of occupied HF states only– this follows from the fact that the density matrix is diagonal in this quantum numbers. Note well that we need to do the additional summation over l', j'.

The m, m'-averaged matrix element (note the lack of m,m' in the above TBME is not a typo) is given by

$$\langle n_1 l j n_2 l' j' | V | n_3 l j n_4 l' j' \rangle \equiv \frac{1}{(2j+1)(2j'+1)} \sum_{mm'} \langle n_1 l j m n_2 l' j' m' | V | n_3 l j m n_4 l' j' m' \rangle \,. \tag{6}$$

Finally,we have defined

$$\rho^{lj}_{n_4 n_2} \equiv \sum_{\bar{n'}} O^{\bar{n}l'j'} C^{l'j'}_{n_4,\bar{n'}} C^{*l'j'}_{n_2 \bar{n'}} \,, \tag{7}$$

where $O^{\bar{n'}l'j'}$ is the number of occupied HF states with quantum numbers $\bar{n'}l'j'$, which is $2j'+1$ for closed-shell systems.

A first useful test of your code is to run the HF calculations without the two-body force. You should then just get the harmonic oscillator single-particle energies.