

Hartree-Fock Calculations of Neutron Drops

Phase 1: Development of a generic HF solver

July 21, 2014

1 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 to analyze some of the techniques that will be presented in this course. Since Hartree-Fock theory is the starting point for most of these techniques (BCS, HFB, RPA, DME, etc.), the first phase of the computational project is to develop a computer program to solve the HF equations in a given s.p. basis (e.g., HO basis).

Rather than attacking the full neutron drop problem head-on, in the first phase we will solve a simplified, somewhat contrived version of the problem in order to get a working HF implementation as quickly as possible. As we will discuss below, a well-designed HF code is split into two components

1. A **Solver class** (or Fortran module) that solves the HF equations independent of the details of the physical system (e.g., neutron drops, nuclei, quantum dots, atoms, choice of s.p. basis, calculation of two-body matrix elements, etc.).
2. A **System class** (or Fortran module) that implements/administers all the details specific to the particular system.

In the first phase we will work with a simplified picture of neutron drops in which only S -wave ($l = 0$) single particle basis states are considered. This will allow us to focus on developing the HF solver without getting bogged down with technical details (angular momentum recoupling, Talmi-Moshinsky transformation brackets, etc.) associated with generating the input two-body matrix elements for the general case.

2 Hartree Fock Equations

As shown in the lectures, the Hartree-Fock equations can be written as a matrix diagonalization problem in a given basis $|\alpha\rangle$ as

$$\sum_{\beta} h_{\alpha\beta} D_{\beta q} = \epsilon_q D_{\alpha q}. \quad (1)$$

The HF hamiltonian is defined as

$$h_{\alpha\beta} = t_{\alpha\beta} + \Gamma_{\alpha\beta}, \quad (2)$$

where the single-particle potential $\Gamma_{\alpha\beta}$ is

$$\Gamma_{\alpha\beta} \equiv \sum_{\mu\nu} v_{\alpha\nu\beta\mu} \rho_{\mu\nu}, \quad (3)$$

$v_{\alpha\beta\gamma\delta}$ are antisymmetrized two-body matrix elements (TBMEs),

$$v_{\alpha\beta\gamma\delta} = \langle \alpha\beta | V | \gamma\delta \rangle = (\alpha\beta | V | \gamma\delta) - (\alpha\beta | V | \delta\gamma), \quad (4)$$

and the density matrix is given by

$$\rho_{\mu\nu} = \sum_{i=1}^N \langle \mu | i \rangle \langle i | \nu \rangle = \sum_{i=1}^N D_{\mu i} D_{\nu i}^*. \quad (5)$$

Note that $t_{\alpha\beta}$ denotes the matrix elements of the 1-body part of the starting hamiltonian. For self-bound nuclei $t_{\alpha\beta}$ is the kinetic energy, whereas for neutron drops, $t_{\alpha\beta}$ 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 $t_{\alpha\beta}$ is diagonal.

The HF equations need to be solved iteratively, since $h_{\alpha\beta}$ depends on the eigenvectors $D_{\alpha q}$ via the density matrix. Therefore, one typically follows the following procedure

1. Start with an initial guess for $D_{\alpha q}^{(0)}$ and construct $\rho_{\nu\mu}^{(0)}$, $h_{\alpha\beta}^{(0)}$.
2. Diagonalize $h_{\alpha\beta}^{(0)}$ and use the lowest N eigenvectors $D_{\alpha i}^{(1)}$ to construct the next iteration for $\rho_{\nu\mu}^{(1)}$, $h_{\alpha\beta}^{(1)}$.
3. Continue the process until things don't change above some threshold from one iteration to the next. For instance, one could iterate until the change in the HF eigenvalues obeys

$$\frac{\sum_p |\epsilon_p^{(n)} - \epsilon_p^{(n-1)}|}{m} \leq \lambda,$$

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.

3 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 *h0* 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 D = 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 gamma = 0; gamma < nStates; gamma++) {
                spPot = 0;
                for (int beta = 0; beta < nStates; beta++) {
                    for (int delta = 0; delta < nStates; delta++)
                    {
                        spPot += densityMatrix(beta, delta, D) *
                            matrixElement(alpha, beta, gamma, delta
                            );
                    }
                }
                h(alpha, gamma) = h(gamma, alpha) = h0(alpha, gamma) +
                    spPot;
            }
        }
        //Computing the HF one-body energies
        eig_sym(E, D, h);
        // Transposing the vectors
        D = trans(D);
    }
}
```

```

    hfIt++;
    // Convergence test
    diff = E - ePrev;
    if (abs(diff.max()) < threshold)
        break;
    ePrev = E;
}
double E0 = calcEnergy(D);
cout << "Final energy E = " << E0 << " after " << hfIt << "
    iterations, error < " << threshold << endl;
}

```

4 Project work plan

Each group should start discussing and working on the following tasks.

4.1 Statement of the model problem

To bypass complications associated with calculating the input $v_{\alpha\beta\gamma\delta}$ (angular momentum coupling, Talmi-Moshinsky transformation, etc.), we start with a simplified version of neutron drops in which our single-particle model space is comprised entirely of S -wave HO wave functions.

$$|\alpha\rangle = |n, l, m, \sigma\rangle \Rightarrow |n, 0, 0, \sigma\rangle. \quad (6)$$

In this restricted model space, we will start with the lightest “closed-shell” neutron drop, $N = 2$. (What are the other possible closed-shell drops in this model space?)

4.2 HF Solver

1. Write “pseudo-code” for your HF solver. Feel free to refer to the C++ listing above for guidance.
2. Start translating your pseudo-code into an actual implementation.

4.3 Groundwork for computing the TBMEs

1. **Two-dimensional radial integrals.** Before you tackle the Minnesota potential, with all the complications arising from its spin-dependent terms and exchange forces (i.e., terms like $V(r)P_r$, where P_r exchanges the spacial positions of particles 1 and 2.), consider a simple spin-independent local potential of gaussian form $V(r) = V_0 e^{-\mu|\mathbf{r}_1 - \mathbf{r}_2|^2}$. Evaluate the expression for the non-antisymmetrized matrix elements

$$(n_1 0 0 \sigma_1, n_2 0 0 \sigma_2 | V | n_3 0 0 \sigma_3 n_4 0 0 \sigma_4), \quad (7)$$

and reduce it to a two-dimensional radial integral. One way to see that the angular integrals are trivial is to perform a multipole expansion of the potential

$$V(|\mathbf{r}_1 - \mathbf{r}_2|) = \sum_l (2l + 1) V_l(r_1, r_2) P_l(\hat{\mathbf{r}}_1 \cdot \hat{\mathbf{r}}_2) \quad (8)$$

$$= 4\pi \sum_l \sum_{m=-l}^l V_l(r_1, r_2) Y_{lm}(\hat{\mathbf{r}}_1) Y_{lm}^*(\hat{\mathbf{r}}_2), \quad (9)$$

where

$$V_l(r_1, r_2) = \frac{1}{2} \int d(\cos \theta) V(|\mathbf{r}_1 - \mathbf{r}_2|) P_l(\cos \theta). \quad (10)$$

The angular integrals are now easy; you should find that they pick out the $l = 0$ term in the multipole expansion, leaving for the spatial part of the matrix element (check the numerical pre factor!)

$$(n_1 0 n_2 0 | V | n_3 0 n_4) \sim \int r_1^2 dr_1 \int r_2^2 dr_2 R_{n_1 0}(r_1) R_{n_2 0}(r_2) V_{l=0}(r_1, r_2) R_{n_3 0}(r_1) R_{n_4 0}(r_2), \quad (11)$$

where

$$V_{l=0}(r_1, r_2) = \frac{1}{2} \int d(\cos \theta) V_0 e^{-\mu(r_1^2 + r_2^2 - 2r_1 r_2 \cos \theta)} \quad (12)$$

can be integrated analytically.

It might be possible to simplify the 2-dimensional radial integral further (e.g., reduce it to a 1-dimensional integral or even integrate it analytically), but we have not checked this. As it is, the 2-dimensional integral is easily evaluated using Gaussian quadrature. Therefore, in your **System class/module**, you should set up a function that, given some $V(r)$, computes the radial integrals of the form in Eq. 11. We will outline next how to take care of spin and antisymmetrization.

2. **Antisymmetrized matrix elements.** The basic inputs to the HF calculation are the antisymmetrized matrix elements. One straightforward approach is to calculate

the non-symmetrized matrix elements in Eq. 7 and then explicitly anti-symmetrize in the ket or bra indices. A more elegant approach is to apply the anti-symmetrizer to the potential *operator*,

$$\begin{aligned}\langle\alpha\beta|V|\gamma\delta\rangle &= (\alpha\beta|V\mathcal{A}_{12}|\gamma\delta) \\ &\equiv (\alpha\beta|\mathcal{V}|\gamma\delta)\end{aligned}\tag{13}$$

where the anti-symmetrized potential operator is defined as

$$\mathcal{V} = V\mathcal{A}_{12} = V(1 - P_{12}) = V(1 - P_\sigma P_r),\tag{14}$$

where P_σ and P_r are spin- and space-exchange operators, respectively. Recall that they are defined as

$$P_\sigma|\sigma\sigma'\rangle = |\sigma'\sigma\rangle \quad \text{and} \quad P_r|\mathbf{r}_1\mathbf{r}_2\rangle = |\mathbf{r}_2\mathbf{r}_1\rangle,\tag{15}$$

with the simple expression for P_σ

$$P_\sigma = \frac{1 + \sigma_1 \cdot \sigma_2}{2}.\tag{16}$$

There is no simple expression for P_r , though for our purposes the definition in Eq. 15 is sufficient.

Using the expression for the Minnesota potential given in the Appendix, show that the anti-symmetrized potential operator takes the form

$$\mathcal{V} = V^D + V^E P_r,\tag{17}$$

where

$$V^D = V^E = \frac{1}{2}(V_R + V_S)(1 - P_\sigma) \equiv v(1 - P_\sigma),\tag{18}$$

and $v \equiv \frac{1}{2}(V_R + V_S)$. Finally, show that we have

$$(n_1\sigma_1n_2\sigma_2|V^D|n_3\sigma_3n_4\sigma_4) = (\delta_{\sigma_1\sigma_3}\delta_{\sigma_2\sigma_4} - \delta_{\sigma_1\sigma_4}\delta_{\sigma_2\sigma_3})(n_1n_2|v|n_3n_4)\tag{19}$$

$$(n_1\sigma_1n_2\sigma_2|V^E P_r|n_3\sigma_3n_4\sigma_4) = (\delta_{\sigma_1\sigma_3}\delta_{\sigma_2\sigma_4} - \delta_{\sigma_1\sigma_4}\delta_{\sigma_2\sigma_3})(n_1n_2|v|n_4n_3),\tag{20}$$

where we've suppressed the angular momentum quantum numbers $l_i m_i$ since they are all zero in the present model. The radial integrals $(n_1n_2|v|n_3n_4)$ are the same form as in Eq. 11. The fully antisymmetrized TBMEs are then given by

$$\langle n_1\sigma_1n_2\sigma_2|V|n_3\sigma_3n_4\sigma_4\rangle = (n_1\sigma_1n_2\sigma_2|V^D|n_3\sigma_3n_4\sigma_4) + (n_1\sigma_1n_2\sigma_2|V^E P_r|n_3\sigma_3n_4\sigma_4).\tag{21}$$

5 Structure of the HF equations for the simplified model space

Once the antisymmetrized TBMEs are in hand, the HF calculation can be done using the algorithm in the snippet of code provided above. Note, however, that this implementation doesn't exploit any symmetries of the problem. One should therefore think of refining the algorithm to take advantage of all possible symmetries.

1. Starting from the expression for the density matrix elements

$$\langle n_1\sigma_1|\rho|n_3\sigma_3\rangle = \sum_{i=1}^N \langle n_1\sigma_1|\phi_i\rangle\langle\phi_i|n_3\sigma_3\rangle, \quad (22)$$

where $|\phi_i\rangle$ are the HF s.p. orbitals, convince yourself that the density matrix is diagonal in spin (and independent of it).

2. Now consider the HF single-particle field matrix elements,

$$\langle n_2\sigma_2|\Gamma|n_4\sigma_4\rangle = \sum_{\sigma_1\sigma_3} \sum_{n_1n_3} \langle n_1\sigma_1n_2\sigma_2|V|n_3\sigma_3n_4\sigma_4\rangle\langle n_1\sigma_1|\rho|n_3\sigma_3\rangle. \quad (23)$$

Using the result that the density matrix is diagonal in spin, convince yourself that this implies Γ (and hence the HF hamiltonian h) is diagonal in spin (and independent of it).

3. Because of the previous two simplifications, the HF hamiltonian is block-diagonal in spin projection. Moreover, the spin up and spin down blocks are the same. Therefore, one only needs to set up the HF matrix for one spin projection (say, σ)

$$\langle n_2\sigma|h|n_4\sigma\rangle = (2n_2 + 3/2)\hbar\omega\delta_{n_2n_4} + \sum_{\sigma'} \sum_{n_1n_3} \langle n_1\sigma'n_2\sigma|V|n_3\sigma'n_4\sigma\rangle\rho_{n_3n_1}. \quad (24)$$

A The Microscopic Neutron Drop Hamiltonian

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

$$\hat{H} = \sum_{i=1}^N \frac{\hat{\mathbf{p}}_i^2}{2m} + \sum_{i=1}^N \frac{1}{2} m \omega \mathbf{r}_i^2 + \sum_{i < j} \hat{V}_{ij}, \quad (25)$$

with $\hbar^2/2m = 20.73 \text{ fm}^2$, $mc^2 = 938.90590 \text{ MeV}$, and \hat{V}_{ij} is the two-body, local, finite-range Minnesota interaction potential

$$\hat{V}(\mathbf{r}_1, \mathbf{r}_2) = \left[\hat{V}_R(\mathbf{r}_1, \mathbf{r}_2) + \frac{1}{2} (1 + \hat{P}_\sigma) \hat{V}_t(\mathbf{r}_1, \mathbf{r}_2) + \frac{1}{2} (1 - \hat{P}_\sigma) \hat{V}_s(\mathbf{r}_1, \mathbf{r}_2) \right] \times \frac{1}{2} (1 + \hat{P}_r), \quad (26)$$

with \hat{P}_σ the spin-exchange operator, and \hat{P}_r the space-exchange operator. The spatial form-factors are

$$\hat{V}_R(\mathbf{r}_1, \mathbf{r}_2) = +V_{0,R} e^{-\kappa_R(\mathbf{r}_1 - \mathbf{r}_2)^2}, \quad (27)$$

$$\hat{V}_t(\mathbf{r}_1, \mathbf{r}_2) = -V_{0,t} e^{-\kappa_t(\mathbf{r}_1 - \mathbf{r}_2)^2}, \quad (28)$$

$$\hat{V}_s(\mathbf{r}_1, \mathbf{r}_2) = -V_{0,s} e^{-\kappa_s(\mathbf{r}_1 - \mathbf{r}_2)^2}. \quad (29)$$

The numerical parameters for the range of the Gaussians and the energy scales are listed in the table below.

Table 1: Parameters defining the Minnesota potential

V	Value	κ	Value
$V_{0,R}$	200.00 MeV	κ_R	1.487 fm^{-2}
$V_{0,t}$	178.00 MeV	κ_t	0.639 fm^{-2}
$V_{0,s}$	91.85 MeV	κ_s	0.465 fm^{-2}